Naval Submarine Medical Research Laboratory

NSMRL Report 1148 31 October 1989

ABDX

A Decision Support System for the Management of Acute Abdominal Pain

Version 3.0

PROGRAMMER'S MANUAL

by

LCDR David Southerland, MC, USN
and
Karen Fisherkeller

Released by:

R. G. Walter, CAPT, DC, USN
Commanding Officer
Naval Submarine Medical Research Laboratory

ABDX

A DECISION SUPPORT SYSTEM FOR THE MANAGEMENT OF ACUTE ABDOMINAL PAIN

VERSION 3.0
PROGRAMMER'S MANUAL

by

LCDR David SOUTHERLAND, MC, USN
and
Karen FISHERKELLER

NAVAL SUBMARINE MEDICAL RESEARCH LABORATORY

NSMRL REPORT NO. 1148

Naval Medical Research and Development Command
Research Work Unit 63706N-M0095.005-5010

Approved and Released by

R. G. Walter

R. G. WALTER, CAPT, DC, USN
Commanding Officer
NavSubMedRschLab

Summary Page

THE PROBLEM:

To provide a programmer's manual for the Acute Abdominal Pain Diagnostic System (ABDX version 3.0) to allow ease of modification to ABDX.

THE FINDINGS:

The manual describes the distributed programs and the non-distributed utility programs. The BASIC source listings for all of the programs are presented. In addition, the formats for the help files, data files, and treatment protocol files are described and the ASCII texts of these files are listed where appropriate.

APPLICATION:

The information presented in this manual will allow programmers to modify ABDX as necessary to enhance its capabilities or to correct program malfunctions. This report replaces NSMRL Report No. 1113.

ADMINISTRATIVE INFORMATION

This investigation was conducted under Naval Medical Research and Development Command Research Work Unit M0095.005-5010. It was submitted for review on 28 August 1989 and approved for publication on 31 October 1989. It has been designated as Naval Submarine Medical Research Laboratory Report No. 1148.

Abstract

ABDX is a medical decision support system for the diagnosis and management of acute abdominal pain. This report is written to function as the programmer's manual for ABDX version 3.0. The report describes the functions of the distributed programs and the non-distributed utility programs and contains the BASIC source listings for the programs. In addition, the formats of the data files, help files, and treatment protocol files are described and the ASCII texts of the files are listed where appropriate.

Familiarity with Microsoft QuickBASIC is required to modify ABDX or to use this manual effectively to identify program malfunctions.

This report replaces NSMRL Report No. 1113.

Table of Contents

1. Intro to ABDX Programmer's Manual 1
   1.1 Purpose of the Programmer's Manual................1
   1.2 Background of ABDX.................................1
2. Description of the Distributed Program Files...........2
3. Description of the Programmer's Utility Files..........2
   3.1 ABDXSTAT.BAS.......................................2
   3.2 CONVTEXT.BAS.......................................2
   3.3 CRYPTDAT.BAS.......................................3
   3.4 INSTALL.BAS..3
   3.5 OLDCONV.BAS..3
   3.6 PACKDATA.BAS.......................................3
   3.7 TRAINCAS.BAS.......................................3
   3.8 TXTMAKE.BAS..3
4. Description of Data Files..............................4
   4.1 REAL.DAT...4
   4.2 SETUP.DAT..4
   4.3 SHIP.DAT...5
   4.4 SIMUL.DAT..5
   4.5 ABDGRAPH.DAT.......................................5
5. Description of Database Files..........................5
   5.1 FEMABD.DAT...5
   5.2 REGABD.DAT...6
6. Description of Definition Files........................6
   6.1 H14.TXT..6
   6.2 H24.TXT..6
   6.3 H34.TXT..7
   6.4 H44.TXT..7
   6.5 H54.TXT..7
   6.6 H15.TXT..7
   6.7 H25.TXT..7
   6.8 H35.TXT..7
   6.9 H45.TXT..7
   6.10 H55.TXT...7
7. Description of Help Files..............................8
   7.1 HABGY.DAT..8
   7.2 HELP.DAT...9
   7.3 HP00.DAT...9
   7.4 HP10.DAT...9
   7.5 HP11.DAT...9
   7.6 HP12.DAT...9
   7.7 HP13.DAT...9
   7.8 HPT10.DAT..9
   7.9 HPT12.DAT..9
   7.10 HSF00.DAT...9

8. Description of Treatment Protocol Files................9
8.1 TX1.DAT..10
8.2 TX2.DAT..10
8.3 TX3.DAT..10
8.4 TX4.DAT..10
8.5 TX5.DAT..10
8.6 TX6.DAT..10
8.7 TX7.DAT..10
8.8 TX8.DAT..10
8.9 TX9.DAT..11
8.10 TX10.DAT..11
9. Description of Miscellaneous Files.....................11
9.1 ABDSX.DAT..11
9.2 ABDTRN.DAT.......................................11
9.3 BESTQUES.DAT.....................................11
9.4 PHRASE.DAT.......................................12
9.5 SUBPIC.BIN.......................................12
10. Compilation/Linking Routine...........................12
11. Installation Routines.................................13

APPENDIX A - DISTRIBUTED PROGRAM LISTINGS

ABDX.BAS..A-1
ABDXNARA.BAS..A-41
ABDXONLY.BAS..A-46
ABDXSHAR.BAS..A-51
ABDXSUB1.BAS..A-72
ABDXSUB2.BAS..A-81
ABDXSUB3.BAS..A-96
ABDXSUB4.BAS..A-106
ABDXSUB5.BAS..A-113
ABDXSUB6.BAS..A-114
DATES.BAS...A-116
FSF600.BAS..A-119
TEMPLATE.BAS..A-165
FPRINT.ASM..A-179
INTRPRT.ASM...A-183
CIPHER.C..A-184

APPENDIX B - UTILITY PROGRAM LISTINGS

ABDXSTAT.BAS..B-1
CRYPTDAT.BAS..B-48
CONVTEXT.BAS..B-51
INSTALL.BAS...B-58
OLDCONV.BAS...B-62
PACKDATA.BAS..B-70
TRAINCAS.BAS..B-74
TXTMAKE.BAS...B-84

APPENDIX C - DEFINITION FILE LISTINGS
H14.TXT..C-1
H24.TXT..C-2
H34.TXT..C-4
H44.TXT..C-6
H54.TXT..C-8
H15.TXT..C-10
H25.TXT..C-11
H35.TXT..C-12
H45.TXT..C-14
H55.TXT..C-15
APPENDIX D - HELP FILE SOURCE LISTINGS
HABGY.ASC......................................D-1
HELP.ASC.......................................D-2
HP00.ASC.......................................D-4
HP10.ASC.......................................D-5
HP11.ASC.......................................D-7
HP12.ASC.......................................D-8
HP13.ASC.......................................D-10
HPT10.ASC......................................D-11
HPT12.ASC......................................D-13
HSF00.ASC......................................D-15
APPENDIX E - TREATMENT PROTOCOL LISTINGS
TX1.ASC..E-1
TX2.ASC..E-7
TX3.ASC..E-11
TX4.ASC..E-14
TX5.ASC..E-19
TX6.ASC..E-22
TX7.ASC..E-26
TX8.ASC..E-30
TX9.ASC..E-36
TX10.ASC.......................................E-38
TX11.ASC.......................................E-41
APPENDIX F - MISCELLANEOUS SOURCE LISTINGS
ABDSX.DAT......................................F-1
BESTQUES.DAT...................................F-5
PHRASE.DAT.....................................F-9
APPENDIX G - MAKE SCRIPT FILE FOR CREATING ABDX.EXE
ABDX...G-1

1. Intro to ABDX Programmer's Manual

ABDX is the Abdominal Pain Diagnostic program developed for the diagnosis and medical management of acute abdominal pain by Independent Duty corpsmen aboard submarines.

Since the publication of the ABDX Programmer's manual, Naval Submarine Medical Research Laboratory (NSMRL) Report # 1113, a gynecological database, updated treatment protocols, EGA support, and color have been added. A new manual is necessary to document the new program.

1.1 Purpose of the Programmer's Manual

The purpose of this manual is to document the actual program listings to aid any future modifications to ABDX version 3.0. This report is a programmer's manual. It contains a brief description of each program and its listing. This manual should be used by a programmer familiar with Microsoft BASICA or QuickBASIC. The manual will not be useful to other readers.

The original programmer's manual for the TEKTRONIX based program (Report #974, "Computer-Assisted Diagnosis Program for Acute Abdominal Pain Program Elements") was published in 1982 and contains much useful information on the original TEKTRONIX version of the program. NSMRL Report No.1113 documented ABDX version 2.0, the first version to run under MS-DOS.

1.2 Background of ABDX

The original program was implemented on the TEKTRONIX 4051 which was available at sea for partial use by the corpsman. In practice, the corpsman did not have adequate access to the machine and NSMRL felt that the Medical Department needed its own small microcomputer. At the time this decision was made, the MS-DOS based laptop microcomputers were the only promising compact microcomputers available. The diagnostic module was then rewritten and enhancements added to allow it to run under MS-DOS.

The only MS-DOS language compilers available at NSMRL were Microsoft BASICA and Microsoft Pascal. BASICA was chosen as the programming language since it had built-in graphics functions and Pascal did not. Also, the original program was written in TEKTRONIX BASIC and we felt that it would be easier to port the program to another dialect of BASIC than to change to an entirely different language. In retrospect, TEKTRONIX BASIC differs so much from BASICA that neither time nor effort was saved by rewriting the program in BASICA.

During conversion of the program to MS-DOS, we found that we wanted to improve the user interface. Modifications were made to the program throughout the conversion based on opinions expressed by individuals observing different interfaces we were testing. The procedure of modifying the interface as different parts were developed resulted in an acceptable user interface, but required much patch work to allow the program to function as we desired. Therefore the source code of the program is somewhat difficult to follow, but this was a necessary trade-off to have a working program in a reasonable amount of time. Future enhancements to the program will result in a more modular format to the program source code.

The initial IBM PC version of the program ran under the BASICA interpreter. This allowed the programmers to run changes immediately without having to re-compile the program. Afterwards, the program was compiled using the BASICA compiler.

The latest version of the program is compiled with Microsoft QuickBASIC 4.0. Microsoft QuickBASIC has superseded BASICA. When the QuickBasic compiler was obtained, new modifications were coded using the QuickBasic syntax (i.e., line numbers not essential, new statements, subprograms, etc).

The program will run on a machine with 512 kilobytes of RAM, and may run on a machine with less memory, but we have no microcomputers with less than 512K at NSMRL. The ABDX system takes approximately 410 kilobytes of disk storage, so it will not run on a 360 kilobyte floppy. It will run on a 3 1/2 inch floppy or a 5 1/4 high density floppy.

2. Description of the Distributed Program Files.

There is one executable program (ABDX.EXE) distributed with the ABDX system. It is the main program. The separately compiled modules ABDX.BAS, ABDXNARA.BAS, TEMPLATE.BAS, DATES.BAS, ABDXONLY.BAS, ABDXSHAR.BAS, ABDXSUB1.BAS, ABDXSUB2.BAS, ABDXSUB3.BAS, ABDXSUB4.BAS, ABDXSUB5.BAS, ABDXSUB6.BAS, FSF600.BAS, FPRINT.ASM, INTRPT.ASM, and CIPHER.C are linked together to form the single executable file, ABDX.EXE.

3. Description of Programmer's Utility Files.

The following programs are not included in the distributed ABDX system, but are useful to the programmer.

3.1 ABDXSTAT.BAS

This program is similar to ABDX and allows the user the ability to modify or delete any real case stored in the database. The simulated case and SF-600 generation routines are unavailable to make room on the Main Option Page for the new selections.

3.2 CONVTEXT.BAS

This program converts ASCII files to and from encrypted data files using the current method of encryption. Also, if the ASCII file was run thru WORD using the TTYFF.PRD printer driver and the DOC.STY stylesheet, this program will automatically add the page formatting to the encrypted data file.

3.3 CRYPTDAT.BAS

This program converts ASCII files to and from encrypted data files using the current method of encryption. This program is used for ASCII files with text only. If dealing with DATA statements, the word "DATA " and double quotes must be stripped out before using this program. Note that this program does just a straight encryption/decryption. No page formatting is done. For encrypted help files, use CONVTEXT.BAS program.

3.4 INSTALL.BAS

This program installs the ABDX system on a hard drive from a distribution floppy. The program prompts the user for the desired hard drive and sub-directory where the ABDX files will reside. It creates the subdirectory if necessary and un-compresses the files from a single compressed file ABDXPAK.EXE. It also creates batch files in the root directory of drive C: to start the program and to make backups of the patient data to a floppy disk.

3.5 OLDCONV.BAS

This program converts ASCII files to and from encrypted data files using the old method of encryption ("David was here once ..."). This program will be used only to get ASCII text from data files in old versions of the diagnostic programs.

3.6 PACKDATA.BAS

This program packs the database (including a priori data) into a compressed file for use by ABDX. The database is supplied as an ASCII file containing the probabilities as numbers between 0.1 and 100 in BASIC DATA statements. Any probability under 128 is placed in a single byte using CHR$(). If the probability is < 1, then that number is multiplied by 10 (to get a whole number) and then added to 128.

3.7 TRAINCAS.BAS

This program creates a binary file ABDTRN.DAT, containing the fifty training cases used in the program.

3.8 TXTMAKE.BAS

This program will take an input ASCII file containing the all definitions of questions used in ABDX and create the appropriate .TXT files used by ABDX.

4. Description of Data Files.

The following files contain information modified by the user.

4.1 REAL.DAT

This is the data file containing the real cases entered. Every time a real case is stored, the case information is appended to the end of this file. If the file does not exist, then it is created.

The data file is a random access file of length 128. Each record contains ten variables which are listed below along with their position in the 128 byte string and a brief description of the variable. Remember that in Microsoft BASIC, all data must be converted to strings before being saved in a random access file.

Variable	Starting Position	Length	Description
SSN$	1	11	SSN including hyphens.
AGE$	1	22	Age as a string.
A$	14	26	Responses in packed format.
OTHER$	40	40	"Other" diagnosis entered by HM.
STARTIME$	80	5	Time of exam as a string.
STARTDATE$	85	10	Date of exam as a string.
HMDX	95	2	HM's DX; use CVI().
SIMULATE	97	2	0 = Simulate; 1 = Real; use CVI().
MAXNUM	99	2	# of computer's DX (1-6); use CVI().
MAXPROB DX;	10	12	Maximum probability of computer's use CVI().

4.2 SETUP.DAT

This line sequential ASCII file is called by the SF600.BAS module. It contains on separate lines seven printer characteristics used in printing the SF600. These characteristics are:

a. LEFTMAR1 = left margin of the front page of the SF600.
b. LEFTMAR2 = left margin of the back page of the SF600.
c. TOP1 = Top margin of the front page of the SF600.
d. TOP2 = Top margin of the back page of the SF600.
e. BOP1 = Bottom margin of the front page of the SF600.
f. BOP2 = Bottom margin of the back page of the SF600.
g. LINWIDTH = Width of each line.

If the file does not exist, it is created the first time SF600.BAS checks for it. The contents of the default file appear at the top of the next page.

```
0
0
0
0
44
56
66
```

4.3 SHIP.DAT

A four line sequential ASCII file containing on separate lines the ship name, ship hull number, the corpsman's name as signed on the SF-600, and the corpsman's SSN (or blank line if the corpsman desires no SSN to be printed on the SF600). An example file looks like this:

```
USS NSMRL
SSN 999
ARTHUR DENT
123-45-6789
```

4.4 SIMUL.DAT

This data file contains the simulated cases entered by the user. The format of the file is exactly the same as that of REAL.DAT (See REAL.DAT for format information).

4.5 ABDGRAPH.DAT

This ASCII data file contains information on the type of monitor. The format of the file is a single character terminated with a CR-LF combination. The single character is "C" if a color monitor is present or "M" if a monochrome monitor is used.

5. Description of Database Files.

5.1 FEMABD.DAT

This file contains the Bayesian database information for the gynecological database. It also contains the a priori information used with the database. PACKDATA.BAS takes FEMABD.BAY as input and creates this file. The format of the file is a random access file with a record length of seven (i.e., the number of diseases considered) bytes and total number of record equals the total number of responses in ABDX.

5.2 REGABD.DAT

This file contains the Bayesian database information for the "regular" non-gynecological database. It also contains the a priori information used with the database. PACKDATA.BAS takes REGABD.BAY as input and creates this file. The format of the file is a random access file with a record length of X bytes and total number of record equals the total number of responses in ABDX. X is seven, the number of diseases considered by the database; however, dyspepsia is combined with NONSAP to give a total of six disease categories.

6. Description of Definition Files.

Each of the following files contains the definitions for the questions on its associated display page. The definition files are not encrypted and are sequential ASCII files. Each line has a single digit number followed by a comma and then a text string. The number refers to the relative question number on the page. The program will compare the single digit number on every line with the relative number of the question for which a definition is desired. For every number match, the corresponding text string is printed. When the number in the file exceeds the number of the question, all text associated with that question has been printed. The last line of the text file will be the last line of the highest numbered question on that page. Do not have extra carriage returns at the end of the file or the program will lock up. If your text editor places extra carriage returns at the end of the file, you can add another line at the end of the text that contains a number higher than the highest numbered question followed by a comma and several spaces or other characters. That line will never be printed, but will show the computer where the last question ends. A sample definition file for a page with three questions appears at the top of the next page.

```
    1,This is text associated
1,with the first question on the page.
2,This text goes with question 2
3,This text goes with question 3.
3,Notice that the definition may be longer than one line
3,and that the last line of the file must have a single
3,digit number greater that the number of questions on
3,the page.
4,This will never be printed.
```

6.1 H14.TXT

This is a sequential ASCII file containing the help text for page 1 of the History section. SITE OF PAIN(onset) and SITE OF PAIN(now) are defined.

6.2 H24.TXT

This is a sequential ASCII file containing the help text for page 2 of the History section. TYPE OF PAIN, SEVERITY OF PAIN, AGGRAVATING FACTORS, PROGRESS OF PAIN, DURATION OF PAIN, and RELIEVING FACTORS are defined.

6.3 H34.TXT

This is a sequential ASCII file containing the help text for page 3 of the History section. NAUSEA, VOMITING, BOWELS, APPETITE, JAUNDICE, and URINATION are defined.

6.4 H44.TXT

This is a sequential ASCII file containing the help text for page 4 of the History section. PREVIOUS INDIGESTION, PREVIOUS SIMILAR PAIN, PREVIOUS SURGERY, PREVIOUS ILLNESS, and TAKING MEDICATIONS are defined.

6.5 H54.TXT

This is a sequential ASCII file containing the help text for page 5 of the History section. Page 5 is displayed only if the patient is female. PERIODS, LAST PERIOD, VAGINAL D/C, PREGNANCY, FAINT/DIZZY, and PREV GYN HX are defined.

6.6 H15.TXT

This is a sequential ASCII file containing the help text for page 1 of the Physical Exam section. TEMPERATURE (F), PULSE RATE, BP (Systolic), and BP (Diastolic) are defined.

6.7 H25.TXT

This is a sequential ASCII file containing the help text for page 2 of the Physical Exam section. MOOD, COLOR, and WBC COUNT are defined.

6.8 H35.TXT

This is a sequential ASCII file containing the help text for page 3 of the Physical Exam section. INSPECTION, SCARS, GUARDING, RIGIDITY, BOWEL SOUNDS, DISTENTION, and MASSES are defined.

6.9 H45.TXT

This is a sequential ASCII file containing the help text for page 4 of the Physical Exam section. TENDERNESS is defined.

6.10 H55.TXT

This is a sequential ASCII file containing the help text for page 5 of the Physical Exam section. MURPHY'S SIGN, REBOUND TENDERNESS, RECTAL EXAM, and VAG EXAM are defined. The VAG EXAM definition is available only if the patient is female.

7. Description of Help Files.

The following files are help files used throughout the program. The files are BASICA random access files with a record length of 75 characters. Each record is encrypted. The method of decryption involves first shortening the record string by removing the spaces that were added by BASIC in order to pad each record string out to the proper length. Then each two character segment (equivalent to one word) of the shortened record string is exclusive OR'ed (XOR'ed) with each two corresponding characters of a 75 character "key" string and is also XOR'ed with the 2 byte word &H3A73. The resulting string is now decrypted and appears as a normal ASCII string.

A "|" in the first column of the decrypted string marks the end of a display page. A "|" in the second column marks the end of the text file. Columns 3-15 contain page information which is printed in the lower right corner of the screen. Columns > 15 contain the record number of the first line of the previous page. If the number is negative, then the current page is the first page.

The strings were encrypted by performing the XOR procedure above; i.e., XOR once to encrypt, XOR again to decrypt. The same procedure was used on the treatment protocol files.

The help files and tx protocols are maintained in Microsoft WORD.DOC format. Use the DAT.STY style sheet, especially the MD division style sheet. The files can be easily modified within WORD and printed to a file using the TTYFF printer driver. Then the print file can be entered into CONVTEXT program, which will read it, encrypt it, compute page pointers, and create the appropriate .DAT random access file for use by ABDX.

The TTYFF driver inserts a CR as the first character to cause printing to begin at the beginning of a line. The CONVTEXT program automatically removes it if present.

A temporary file T$EMP.$AT will be written, and it will be used for input for making the encrypted .DAT file. You can still use files with the '|'. Just make sure that you count correctly and end each file with '||'; make sure that there are no FF's in the file. If you use files with '|', no temporary file is created.

NOTE: You can use any word processor to maintain the files as long as you set the page length to 22 lines, set the page width to 74, and set the top, bottom, left, and right margins to 0. In addition, you need a printer driver which will print to disk using form feeds as end of page markers instead of filling out the end of the page with carriage return/line feed characters.

7.1 HABGY.DAT

This encrypted file contains general information for selecting the proper database to use if the patient is female.

7.2 HELP.DAT

This encrypted file contains general information for the program.

7.3 HP00.DAT

This encrypted file contains the help text for the Corpsman's Diagnosis Page.

7.4 HP10.DAT

This encrypted file contains the help text for the Diagnostic Summary Page.

7.5 HP11.DAT

This encrypted file contains the help text for the Treatment Summary Page.

7.6 HP12.DAT

This encrypted file contains the help text for the Main Options Page.

7.7 HP13.DAT

This encrypted file contains the help text for the Data Entry Options Page.

7.8 HPT10.DAT

This encrypted file contains the help text for the Training Diagnostic Summary Page.

7.9 HPT12.DAT

This encrypted file contains the help text for the Training Option Page.

7.10 HSF00.DAT

This encrypted file contains the help text for selecting the desired output device in the SF600 generation portion of the program.

8. Description of Treatment Protocol Files.

The following files contain the treatment protocols for each of the diagnoses considered in the program. The files are BASICA random access files with a record length of 75 characters. Each record is encrypted. The method of decryption involves first shortening the record string by removing the spaces that were added by BASIC in order to pad each record string out to the proper length. Then each two character segment (equivalent to one word) of the shortened record string is exclusive OR'ed (XOR'ed) with each two corresponding characters of a 75 character "key" string and is also

XOR'ed with the 2 byte word &H3A73. The resulting string is now decrypted and appears as a normal ASCII string.

A "|" in the first column of the decrypted string marks the end of a display page. A "|" in the second column marks the end of the text file. Columns 3-15 contain page information which is printed in the lower right corner of the screen. Columns > 15 contain the record number of the first line of the previous page. If the number is negative, then the current page is the first page.

The strings were encrypted by performing the XOR procedure above; i.e., XOR once to encrypt, XOR again to decrypt. The same procedure was used on the help files.

These files have been maintained in Microsoft WORD format. See 7. Description of Help Files for more information.

8.1 TX1.DAT

This is the encrypted treatment protocol for Appendicitis.

8.2 TX2.DAT

This is the encrypted treatment protocol for Non Specific Abdominal Pain.

8.3 TX3.DAT

This is the encrypted treatment protocol for Renal Colic.

8.4 TX4.DAT

This is the encrypted treatment protocol for Perforated Duodenal Ulcer.

8.5 TX5.DAT

This is the encrypted treatment protocol for Cholecystitis.

8.6 TX6.DAT

This is the encrypted treatment protocol for Small Bowel Obstruction.

8.7 TX7.DAT

This is the encrypted treatment protocol for Pelvic Inflammatory Disease.

8.8 TX8.DAT

This is the encrypted treatment protocol for Urinary Tract Infection.

8.9 TX9.DAT

This is the encrypted treatment protocol for Ovarian Cyst.

8.10 TX10.DAT

This is the encrypted treatment protocol for Ectopic Pregnancy.

8.11 TX11.DAT

This is the encrypted treatment protocol for Threatened Abortion.

9. Description of Miscellaneous Files.

The following files are grouped here for convenience.

9.1 ABDSX.DAT

This encrypted sequential file contains a list of the responses for the History and Physical Exam sections of the datasheet. It is called by ABDX.BAS to display the history items marked by the user if SHOW H & P or SHOW MISSED ITEMS is selected while on the Diagnostic Summary Page. It is created by CRYPTDAT.BAS with its ASCII version, ABDSX.ASC, as input.

9.2 ABDTRN.DAT

This binary data file contains history and physical examination information for each of the fifty training cases used in the program. Each case consumes 26 bytes, giving this file a total length of 1300 bytes. The compression method is the same one used when storing real and simulated cases. It is created by TRAINCAS.BAS.

9.3 BESTQUES.DAT

This encrypted sequential file contains questions or statements for the corpsman to consider in differentiating his diagnosis from the computer generated diagnosis, if the gynecological database is used. The format is:

```
xx yy zz Disease 1 vs Disease 2
text
text,
text
13  3 25 Disease 1 vs Disease 3
Consider Disease 1 if ...
Consider Disease 3 if ...
Consider neither if ...
xx yy zz Disease 1 vs Disease 4
text
etc.
```

xx is a number identifying the two diseases being contrasted. Since the seven gynecological diseases can be numbered sequentially, it is computed by taking the smaller number of the two diseases and placing it in the ten's position, while the larger number is placed in the unit's position. This ensures that the same text is chosen whether the corpsman's diagnosis is Disease 1 and the computer's diagnosis is Disease 2 or vice versa.

yy is the number of lines of text, and zz is the number of characters in the longest line of the text. The phrase "Disease 1 vs Disease 2" is just a heading for your convenience. It is not displayed in the program. It is created by CRYPTDAT.BAS with its ASCII version, BESTQUES.ASC, as input.

9.4 PHRASE.DAT

This encrypted sequential file is used by the SF-600 generation module FSF600.BAS to generate a patient narrative for the medical record. Each line contains a number occupying the first two positions of the line. The remainder of the line comprises a phrase associated with the response identified by the number. The file is created by CRYPTDAT.BAS from the input file PHRASE.ASC.

9.5 SUBPIC.BIN

This binary file contains the submarine or other vessel picture drawn at the beginning of the program. It is loaded directly into video RAM with the BASIC command BLOAD. The file was created by saving the desired graphics screen using BSAVE to dump the video RAM (&HB800 - &HBC00) to the file.

This file is unnecessary. If it is not present, the ABDX will display the name and vessel for whom the program has been configured.

10. Compilation/Linking Routine

The files with the .BAS extension will compile using the QuickBASIC 4.0 (or greater) compiler BC.EXE. The assembly language routines were compiled with the Microsoft MACRO Assembler Version 5.1, MASM.EXE, but they also compile with versions as early as 1.27 if the 5.0 specific .MODEL commands are replaced with the conventional model designations. Microsoft QuickC version 1.0 was used to compile CRYPT.C. LINK.EXE, the Microsoft linker, should be version 3.05 or greater to use the /E option which will reduce the disk storage size of the program.

MAKE is a neat utility used to compile the different files as necessary and link them together to create ABDX.EXE. MAKE allows you to re-compile just those files which have changed since you last compiled the program. MAKE was included with QuickC.

ABDX (no extension) is the MAKE script file used to compile ABDX.EXE. It is listed in Appendix G.

11. Installation Routines

Before distribution, all necessary files are compressed into a single file using PKZIP. The single file is then converted into ABDXPAK.EXE which when run, un-compresses itself into the original files.

The user receives a distribution disk containing three files. The first file is ABDXPAK.EXE. The second file is a complied QuickBASIC program INSTALL.EXE which the user runs to install ABDX on the target computer. The third file is an ASCII sequential file containing multiple lines (approximately 50) of the letter "Y" followed by carriage return/line feed characters and is used by the installation program if the program is installed in the same sub-directory as a previous version of the program.

For the user to install the program, he places the distribution diskette in a floppy drive and makes the floppy drive the default drive. Then he enters INSTALL and follows the directions. The install program will ask him for the drive and subdirectory in which to place the ABDX. It will create two batch files in the root directory of drive C:. ABDOMEN.BAT is used to start the program. ABDBKUP.BAT is used to backup the data files to a floppy in drive A:.

Acknowledgments

The original Abdominal Pain Diagnostic Program was produced by Dr. F. T. deDombal at the University of Leeds, England. Several editions of support manuals and versions of the diagnostic program were produced by J. V. Henderson, B. Ryack, G. Moeller, D. C. Arthur, R. Post, W. Schroeder, and S. F. Osborne for use on the Tektronix 4051. Without their efforts the program would not exist. The present revision has been completely rewritten for use with an MS-DOS microcomputer and the treatment protocols have been updated.

The authors would like to express their deepest appreciation for the advice and constructive criticism of Dr. George Moeller, Dr. Bernard Ryack, Mr. Ernest Noddin, Dr. Kendall Bryant, HMC(SS) Dan Johansen, HMC(SS) Dale Hamilton, Dr. Donald Tappan, CAPT Douglas Stetson, LT Barclay Caras, and HM1 Patrick Flaherty. In addition, Ms. Ellen Perkins and Mr. Harry Fiske provided superb technical support.

Appendix A
Distributed Program Listings

ABDX.BAS

```
DECLARE SUB UnPackDatabase (filename$, VARIABLE!(), APRIORI#(),
       arraywidth%, arraylength%)
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB CompareFemDXes (HMDX%, MAXNUM%)
DECLARE SUB TextPause ()
DECLARE SUB SetVideoMode (vm%)
DECLARE SUB experimental ()
DECLARE FUNCTION Translate% (HMDX%)
DECLARE SUB SF600 (BOAT1$, BOAT2$, HMNAM$, HMSSN$)
DECLARE SUB CompareAbdDXes (COMPAR%(), VARIABLE%(), MAXNUM%, HMDX%,
       QUESTPTR%(), QUESTIONS$())
DECLARE FUNCTION Centered% (s$)
DECLARE SUB SelectDatabase (lookatfemale%, VARIABLE%())
DECLARE SUB ComputeFinalProbs (NUMDISEASES%, MAXNUMBER%,
       MAXPROBABILITY%, PROB#(), FINALPROB#())
DECLARE SUB TXMenu (MAXNUM%, Ptsex$)
DECLARE SUB MaleGraph (FINPROB#())
DECLARE SUB DisplayEncryptedFile (TheFile$, ReturnPage%)
DECLARE SUB FemaleGraph (FINPROB#())
DECLARE SUB TextDxPause ()
DECLARE SUB PutCase (whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
DECLARE SUB ResetVariables (VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
DECLARE SUB MenuSummaryPage (menurow%, menucol%, NR%, resplength%,
       exitchar$, menuheading$, Choices$(), HELPFILE$)
DECLARE SUB MenuEntryPage (NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
DECLARE SUB GetCase (filnam$, whichcase%, VARIABLE%(), SSN$, AGE$,
       OTHER$, STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
DECLARE SUB PackArray (PackString$, thearray%())
DECLARE SUB ModifyNarrative (CASENUM%, escflag%)
DECLARE SUB UnPackArray (PackString$, thearray%())
DECLARE SUB DisplayHPgetstatments (SXloc%(), SXresp$(), abortHP%)
DECLARE SUB DisplayMissedHP (SSN$, STARTIME$, STARTDATE$, VARIABLE%(),
       THECASE%())
DECLARE SUB DisplayHProwcol (row%, col%, sxstrng$)
DECLARE SUB CenterString (infostring$)
DECLARE SUB DisplayHPFrame (TRAINING%, SIMULATE%, SSN$, STARTIME$,
       STARTDATE$)
DECLARE SUB DisplayHP (TRAINING%, SIMULATE%, SSN$, STARTIME$,
       STARTDATE$, VARIABLE%())
DECLARE SUB DisplayHPprint (HP%, SXloc%(), SXresp$(), VARIABLE%())
DECLARE SUB TextContinuePrompt ()
DECLARE SUB DisplayHPhelp ()
DECLARE SUB DisplayHPTitle (HP%)
DECLARE FUNCTION Exists% (FIL$)
DECLARE SUB PaintGraph (VAR%, WhichOne%)
DECLARE SUB DrawGraph (WhichOne%)
DECLARE FUNCTION VideoMode% ()
DECLARE SUB DataEntryPage (exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
DECLARE SUB PutCursor (quest%, resp%, Choices$())
DECLARE SUB UpdateAsterisk (FirstRow%, FirstCol%, NonePtr%,
       VariablePtr%, GraphFlag%, Numresp%, OffSet%, VARIABLE%())
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB HPframe ()
DEFINT A-Z
DECLARE SUB SexSSNAgeDate (STFLAG%, TRAINING%, SIMULATE%, sex$, SSN$,
       AGE$, STARTDATE$, STARTIME$, VARIABLE%())
DECLARE SUB InitializeColors (graphmode$, monmode$)
DECLARE SUB GetGraphMode (graphmode$, monmode$)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB hpresponsePRINT (a$)
DECLARE SUB textPRINT (a$)
DECLARE SUB questionPRINT (a$)
DECLARE SUB headingPRINT (a$)
DECLARE SUB responsePRINT (a$)
DECLARE SUB SetTrainingColors (TRAINING%)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB SetScreenMode (ScrnMode%)
DECLARE SUB GetKey (a$)
DECLARE SUB narrative (NR%, CASENUM%, THECASE%())
DECLARE SUB LoadTrainingCase (CASENUM%, THECASE%())
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB GetBoatStuff (BOAT1$, BOAT2$, HMNAM$, HMSSN$,
       VersionNumber$)
DECLARE SUB Disclaimer (VERSION$)
DECLARE SUB GraphContinuePrompt ()
DECLARE SUB BoxSelections (actyl%, XPOINT%, NumOfResp%, Xwidth%)


' $INCLUDE: 'include.bas'
black% = 0
```

```
blue% = 1
green% = 2
cyan% = 3
red% = 4
magenta% = 5
brown% = 6
white% = 7
gray% = 8
ltblue% = 9
ltgreen% = 10
ltcyan% = 11
ltred% = 12
ltmagenta% = 13
yellow% = 14
hiwhite% = 15

'       Dummy values so that variables are declared in main module.  They
        are
'        modified shortly by SetTrainingColors.
headingcolor% = 1
framecolor% = 1
frametype% = 1

  REM Copyright (C) 1985,1986,1987,1988 Navy Submarine Medical Research
       Laboratory
  KEY OFF
      RANDOMIZE TIMER
  DEFINT A-Z
REM $DYNAMIC
   DIM VARIABLE%(200)
   DIM question$(10), MULTIP%(10), Numresp%(10)
   DIM None(10), GraphFlag(10), SKIPBLANK(10), VariablePtr(10)
   DIM Choices$(10, 14)
   DIM QUESTPTR%(44), QUESTIONS$(44), DIFFER%(15)
   DIM BAYES!(154, 7), PROB#(7), FINPROB#(7), APRIORI#(7)
   DIM FEMBAYES!(174, 7), FEMPROB#(7), FEMFINPROB#(7), FEMAPRIORI#(7)
   DIM actrow%(MAXQUESTIONS%)
      DIM COMPAR%(6, 6, 15)

   DIM THECASE%(200), INARRAY%(7), OUTARRAY%(7), bargraph%(7)

'   male = 0
'   female = 1

' the following is added for testing purposes. but allows the program
' to be used normally, if "test" is not used at the command prompt.
' ie, ABDX test <CR>.
IF COMMAND$ = "TEST" THEN
  davidflag% = 1
```

```
ELSE
  davidflag% = 0
END IF

   VersionNumber$ = "3.00"
   VERSION$ = " (ver " + VersionNumber$ + ")"
      TRAINING = 0             ' 0 = main   ;  1 = training
      STFLAG = 0               ' 0 = no changes; 1 = changes have been
       made
      SIMULATE = 1             ' 1 = main   ;  0 = simulated (I know, I
       know!)

'      CALL SetTrainingColors(TRAINING)
      REM main prog > TRAINING=0; training prog > TRAINING=1

      CALL GetGraphMode(graphmode$, monmode$)

'       Set up graphics mode default (checked for CGA or EGA in
       graphmode$)
      CALL InitializeColors(graphmode$, monmode$)
'       Initialize display page colors
      CALL SetTrainingColors(TRAINING)

      ' BIOS to appropriate 80 col text mode
      IF monmode$ = "C" THEN
      '  CALL SetVideoMode(3)    'screen 0, 80 col, color
      ELSE
      '  CALL SetVideoMode(2)    'screen 0, 80 col, B&W
      END IF


'       Get name of vessel, user's name, and display submarine if
       present.
      CALL GetBoatStuff(BOAT1$, BOAT2$, HMNAM$, HMSSN$, VersionNumber$)
'            Go to subroutine to enter Bayesian probabilities,
' response and category names,and clear array variable.
      GOSUB 60000
      CALL GetKey(a$)
'       Subprogram which displays warning/disclaimer.
      CALL Disclaimer(VERSION$)

   SELECT CASE GRAPHICS
     CASE 9
       Vertbits = 14
       YOffsetpict = 4
     CASE 2
       Vertbits = 8
       YOffsetpict = 6
     CASE ELSE
```

```
      Vertbits = 8
      YOffsetpict = 6
  END SELECT

  GOTO 31000

'   REM   Enter SS#, Age.
30  CALL SexSSNAgeDate(STFLAG, TRAINING, SIMULATE, sex$, SSN$, AGE$,
       STARTDATE$, STARTIME$, VARIABLE%())

100 REM Data Entry Option Page
    ' Set/Reset flag for frame drawing on H&P pages
    FrameFlag = 0

    ' Choices for Main Option Page.
    Choices$(1, 1) = "GO TO HISTORY PAGES        "
    Choices$(1, 2) = "GO TO PHYSICAL EXAM PAGES "
    Choices$(1, 3) = "MAKE DIAGNOSIS            "
    Choices$(1, 4) = "GO TO SSN/AGE/TIME PAGE   "
    Choices$(1, 5) = "RETURN TO MAIN OPTION PAGE"
    IF TRAINING = 1 THEN
      Choices$(1, 5) = "GO TO TRAINING OPTION PAGE"
      TYP$ = "Training "
    ELSE
      TYP$ = ""
    END IF

    ' New method for menu
    NR% = 1
    resplength% = 5
    DATAHEADING$ = "Abdominal Pain Diagnosis " + TYP$ + "Program" +
       VERSION$
    menuheading$ = "Data Entry Options:"
    HELPFILE$ = "HP13.DAT"

    CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
    ON NR GOTO 1000, 5000, 50000, 30, 31000
    GOTO 100


1000 REM PAGE 1 of Hx.  Similar to above page.
     IF sex$ = FEMALE$ THEN
       MaxHxPages$ = "5"
     ELSE
       MaxHxPages$ = "4"
     END IF

1010 NUMQUEST = 2
```

```
     NUMCOL1QUESTS = 1
     TOPROW = 4
     TOPCOL = 19
     HELPFILE$ = "H14.TXT"
     DATAHEADING$ = "        History        "
     PAGEOF$ = "1"
     OFPAGE$ = MaxHxPages$

     FOR i = 1 TO 2
       question$(i) = QUESTIONS$(i)
       MULTIP%(i) = 1
       Numresp%(i) = QUESTPTR%(i)
       None(i) = 13
       GraphFlag(i) = i
       SKIPBLANK(i) = 2
Choices$(i, 1) = "   RUQ     ": Choices$(i, 2) = "    LUQ      "
Choices$(i, 3) = "   RLQ     ": Choices$(i, 4) = "    LLQ      "
Choices$(i, 5) = "UPPER HALF ": Choices$(i, 6) = "LOWER HALF   "
Choices$(i, 7) = "RIGHT HALF ": Choices$(i, 8) = "LEFT HALF    "
Choices$(i, 9) = "  CENTRAL  ": Choices$(i, 10) = " GENERAL    "
       Choices$(i, 11) = "RIGHT FLANK": Choices$(i, 12) = "LEFT FLANK "
       Choices$(i, 13) = " NO PAIN   "
     NEXT i

       VariablePtr(1) = 11
       VariablePtr(2) = 24

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)

     IF exitchar$ = "P" THEN 100
     IF exitchar$ = "X" THEN 100

2000 REM Page 2 of Hx
' Multip (Whether Multiple Responses are Possible (1)
' or Not (0)); and Numresp (Number of Symptoms in Each Category).

     NUMQUEST = 6
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H24.TXT"
     DATAHEADING$ = "        History        "
     PAGEOF$ = "2"
     OFPAGE$ = MaxHxPages$
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 2)
```

```
NEXT i

question$(1) = QUESTIONS$(3)
  Choices$(1, 1) = "Intermittent"
  Choices$(1, 2) = "Steady      "
  Choices$(1, 3) = "Colicky     "
  VariablePtr(1) = 37
  MULTIP(1) = 0
  None(1) = 0
  GraphFlag(1) = 0
  SKIPBLANK(1) = 0

question$(2) = QUESTIONS$(4)
  Choices$(2, 1) = "Moderate    "
  Choices$(2, 2) = "Severe      "
  VariablePtr(2) = 40
  MULTIP(2) = 0
  None(2) = 0
  GraphFlag(2) = 0
  SKIPBLANK(2) = 1

question$(3) = QUESTIONS$(5)
  Choices$(3, 1) = "Movement    "
  Choices$(3, 2) = "Cough       "
  Choices$(3, 3) = "Breathing   "
  Choices$(3, 4) = "Food        "
  Choices$(3, 5) = "Other       "
  Choices$(3, 6) = "None        "
  VariablePtr(3) = 42
  MULTIP(3) = 1
  None(3) = 6
  GraphFlag(3) = 0
  SKIPBLANK(3) = 3

question$(4) = QUESTIONS$(6)
  Choices$(4, 1) = "Better      "
  Choices$(4, 2) = "Same        "
  Choices$(4, 3) = "Worse       "
  VariablePtr(4) = 48
  MULTIP(4) = 0
  None(4) = 0
  GraphFlag(4) = 0
  SKIPBLANK(4) = 0

question$(5) = QUESTIONS$(7)
  Choices$(5, 1) = "12h or less "
  Choices$(5, 2) = "12-24h      "
  Choices$(5, 3) = "24-48h      "
  Choices$(5, 4) = "48+h        "
```

```
       VariablePtr(5) = 51
       MULTIP(5) = 0
       None(5) = 0
       GraphFlag(5) = 0
       SKIPBLANK(5) = 1

     question$(6) = QUESTIONS$(8)
       Choices$(6, 1) = "Lying Still "
       Choices$(6, 2) = "Vomiting    "
       Choices$(6, 3) = "Antacids    "
       Choices$(6, 4) = "Food        "
       Choices$(6, 5) = "Other.      "
       Choices$(6, 6) = "None        "
       VariablePtr(6) = 55
       MULTIP(6) = 1
       None(6) = 6
       GraphFlag(6) = 0
       SKIPBLANK(6) = 1

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 1010
       IF exitchar$ = "X" THEN 100

3000 REM Page 3 of Hx

     NUMQUEST = 6
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H34.TXT"
     DATAHEADING$ = "History - Other Symptoms"
     PAGEOF$ = "3"
     OFPAGE$ = MaxHxPages$
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 8)
       GraphFlag(i) = 0
       question$(i) = QUESTIONS$(8 + i)
     NEXT i

     Choices$(1, 1) = "Present       ": Choices$(1, 2) = "Absent
       "
     VariablePtr(1) = 61
     MULTIP(1) = 0
     None(1) = 0
     SKIPBLANK(1) = 1
     Choices$(2, 1) = "Present        ": Choices$(2, 2) = "Absent
       "
     VariablePtr(2) = 63
     MULTIP(2) = 0
     None(2) = 0
     SKIPBLANK(2) = 1

     Choices$(3, 1) = "Normal         ": Choices$(3, 2) = "Constipated
       "
     Choices$(3, 3) = "Diarrhea       ": Choices$(3, 4) = "Blood in
       Stool"
     Choices$(3, 5) = "Mucus in Stool"
     VariablePtr(3) = 65
     MULTIP(3) = 1
     None(3) = 1
     SKIPBLANK(3) = 1

     Choices$(4, 1) = "Decreased      ": Choices$(4, 2) = "Normal
       "
     VariablePtr(4) = 70
     MULTIP(4) = 0
     None(4) = 0
     SKIPBLANK(4) = 1

     Choices$(5, 1) = "Present        ": Choices$(5, 2) = "Absent
       "
     VariablePtr(5) = 72
     MULTIP(5) = 0
     None(5) = 0
     SKIPBLANK(5) = 1

     Choices$(6, 1) = "Normal         ": Choices$(6, 2) = "Frequency
       "
     Choices$(6, 3) = "Painful        ": Choices$(6, 4) = "Dark Urine
       "
     Choices$(6, 5) = "Blood in Urine"
     VariablePtr(6) = 74
     MULTIP(6) = 1
     None(6) = 1
     SKIPBLANK(6) = 1

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 2000
       IF exitchar$ = "X" THEN 100

4000 REM PAGE 4 of Hx
```

```
     NUMQUEST = 5
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H44.TXT"
     DATAHEADING$ = " History - Past History "
     PAGEOF$ = "4"
     OFPAGE$ = MaxHxPages$

     FOR i = 1 TO NUMQUEST
       MULTIP%(i) = 0
       Numresp%(i) = QUESTPTR%(i + 14)
       question$(i) = QUESTIONS$(14 + i)
       Choices$(i, 1) = "Yes": Choices$(i, 2) = "No"
       None(i) = 0
       GraphFlag(i) = 0
       VariablePtr(i) = 79 + (i - 1) * 2
       SKIPBLANK(i) = 1
     NEXT i



     CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
     IF exitchar$ = "P" THEN 3000
     IF exitchar$ = "X" THEN 100



4500 IF sex$ = FEMALE$ THEN
       REM PAGE 5 of Hx  Only shown if patient is female.
       NUMQUEST = 6
       NUMCOL1QUESTS = 3
       TOPROW = 4
       TOPCOL = 7
       HELPFILE$ = "H54.TXT"
       DATAHEADING$ = "    History - OB/GYN     "
       PAGEOF$ = "5"
       OFPAGE$ = MaxHxPages$
       FOR i = 1 TO NUMQUEST
 Numresp%(i) = QUESTPTR%(i + 38)
 GraphFlag(i) = 0
 question$(i) = QUESTIONS$(i + 38)
 MULTIP(i) = 0
 SKIPBLANK(i) = 1
 None(i) = 0
       NEXT i
       Choices$(1, 1) = "Not Started ": Choices$(1, 2) = "Ceased"
       Choices$(1, 3) = "Regular": Choices$(1, 4) = "Irregular"
       VariablePtr(1) = 160

       Choices$(2, 1) = "Normal": Choices$(2, 2) = "Late/Overdue"
  '     CHOICES$(2, 3) = "Overdue"  -- combined with LATE in the
       database.
       VariablePtr(2) = 164

       Choices$(3, 1) = "Yes": Choices$(3, 2) = "No"
       VariablePtr(3) = 166

       Choices$(4, 1) = "Impossible": Choices$(4, 2) = "Possible"
       Choices$(4, 3) = "Confirmed"
       VariablePtr(4) = 168

       Choices$(5, 1) = "Yes": Choices$(5, 2) = "No"
       VariablePtr(5) = 171
       SKIPBLANK(5) = 2

       Choices$(6, 1) = "Yes": Choices$(6, 2) = "No"
       VariablePtr(6) = 173

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 4000
       ' "X" and "N" will fall through the END IF and will goto 100
       ' automatically.
     END IF
     GOTO 100

5000 REM Page 5; Page 1 of PE.
     NUMQUEST = 4
     NUMCOL1QUESTS = 2
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H15.TXT"
     DATAHEADING$ = " Physical - Vital Signs  "
     PAGEOF$ = "1"
     OFPAGE$ = "5"

     FOR i = 1 TO NUMQUEST
       MULTIP%(i) = 0
       Numresp%(i) = QUESTPTR%(i + 19)
       None(i) = 0
       GraphFlag(i) = 0
  question$(i) = QUESTIONS$(i + 19)
NEXT i

Choices$(1, 1) = "< 98.6      ": Choices$(1, 2) = "98.6 - 100.2"
Choices$(1, 3) = "100.3 - 102 ": Choices$(1, 4) = "> 102       "
VariablePtr(1) = 89
SKIPBLANK(1) = 1

Choices$(2, 1) = "< 80        ": Choices$(2, 2) = "80 - 99     "
Choices$(2, 3) = "> 99        "
VariablePtr(2) = 93
SKIPBLANK(2) = 1

Choices$(3, 1) = "< 90        ": Choices$(3, 2) = "90 - 129    "
Choices$(3, 3) = "> 129       "
VariablePtr(3) = 96
SKIPBLANK(3) = 1

Choices$(4, 1) = "< 70        ": Choices$(4, 2) = "70 - 89     "
Choices$(4, 3) = "> 89        "
VariablePtr(4) = 99
SKIPBLANK(4) = 2

CALL DataEntryPage(exitchar$, question$(), Choices$(),
  VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
  GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
  NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
IF exitchar$ = "P" THEN 100
IF exitchar$ = "X" THEN 100


5500 REM Page 6; Page 2 of PE.

NUMQUEST = 3
NUMCOL1QUESTS = 2
TOPROW = 4
TOPCOL = 7
HELPFILE$ = "H25.TXT"
DATAHEADING$ = " Physical - General/Lab  "
PAGEOF$ = "2"
OFPAGE$ = "5"
FOR i = 1 TO NUMQUEST
  Numresp%(i) = QUESTPTR%(i + 23)
  question$(i) = QUESTIONS$(i + 23)
  GraphFlag(i) = 0
NEXT i
```

```
     Choices$(1, 1) = "Normal         ": Choices$(1, 2) = "Distressed
       "
     Choices$(1, 3) = "Anxious        "
     VariablePtr(1) = 102
     MULTIP(1) = 0
     None(1) = 0
     SKIPBLANK(1) = 1

     Choices$(2, 1) = "Normal         ": Choices$(2, 2) = "Pale
       "
     Choices$(2, 3) = "Flushed        ": Choices$(2, 4) = "Jaundiced
       "
     Choices$(2, 5) = "Cyanotic       "
     VariablePtr(2) = 105
     MULTIP%(2) = 1
     None(2) = 1
     SKIPBLANK(2) = 2

     Choices$(3, 1) = "< 8,000        ": Choices$(3, 2) = " 8,000 -
       10,000"
     Choices$(3, 3) = "10,100 - 12,000": Choices$(3, 4) = "12,100 -
       15,000"
     Choices$(3, 5) = "> 15,000       "
     VariablePtr(3) = 110
     MULTIP%(3) = 0
     None(3) = 0
     SKIPBLANK(3) = 1

     CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
     IF exitchar$ = "P" THEN 5000
     IF exitchar$ = "X" THEN 100


6000 REM Page 7; Page 3 of PE.

     NUMQUEST = 7
     NUMCOL1QUESTS = 4
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H35.TXT"
     DATAHEADING$ = "Physical - Abdominal Exam"
     PAGEOF$ = "3"
     OFPAGE$ = "5"
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 26)
       question$(i) = QUESTIONS$(i + 26)
```

```
  GraphFlag(i) = 0
  MULTIP(i) = 0
  SKIPBLANK(i) = 1
NEXT i

Choices$(1, 1) = "Normal                 "
Choices$(1, 2) = "Visible Perstalsis     "
Choices$(1, 3) = "Decreased Abd. Movement"
VariablePtr(1) = 115
None(1) = 0

Choices$(2, 1) = "Present                "
Choices$(2, 2) = "Absent                 "
VariablePtr(2) = 118
None(2) = 0

Choices$(3, 1) = "Present                "
Choices$(3, 2) = "Absent                 "
VariablePtr(3) = 120
None(3) = 0

Choices$(4, 1) = "Present                "
Choices$(4, 2) = "Absent                 "
VariablePtr(4) = 122
None(4) = 0

Choices$(5, 1) = "Normal                 "
Choices$(5, 2) = "Absent                 "
Choices$(5, 3) = "Hyperactive            "
VariablePtr(5) = 124
None(5) = 0

Choices$(6, 1) = "Present                "
Choices$(6, 2) = "Absent                 "
VariablePtr(6) = 127
None(6) = 0

Choices$(7, 1) = "Present                "
Choices$(7, 2) = "Absent                 "
VariablePtr(7) = 129
None(7) = 0

CALL DataEntryPage(exitchar$, question$(), Choices$(),
  VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
  GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
  NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
IF exitchar$ = "P" THEN 5500
IF exitchar$ = "X" THEN 100
```

```
7000 REM PAGE 8; Page 4 of PE.

     NUMQUEST = 1
     NUMCOL1QUESTS = 1
     TOPROW = 4
     TOPCOL = 19
     HELPFILE$ = "H45.TXT"
     DATAHEADING$ = "Physical - Abdominal Exam"
     PAGEOF$ = "4"
     OFPAGE$ = "5"

       question$(1) = QUESTIONS$(34)
       MULTIP%(1) = 1
       Numresp%(1) = QUESTPTR%(34)
       None(1) = 13
       GraphFlag(1) = 3
       SKIPBLANK(1) = 2
       Choices$(1, 1) = "   RUQ      ": Choices$(1, 2) = "   LUQ      "
       Choices$(1, 3) = "   RLQ      ": Choices$(1, 4) = "   LLQ      "
       Choices$(1, 5) = "UPPER HALF  ": Choices$(1, 6) = "LOWER HALF  "
       Choices$(1, 7) = "RIGHT HALF  ": Choices$(1, 8) = "LEFT HALF   "
       Choices$(1, 9) = "  CENTRAL   ": Choices$(1, 10) = " GENERAL    "
       Choices$(1, 11) = "RIGHT FLANK": Choices$(1, 12) = "LEFT FLANK "
       Choices$(1, 13) = " NO PAIN   "

       VariablePtr(1) = 131

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 6000
       IF exitchar$ = "X" THEN 100


8000 REM Page 9; Last page of PE.

     NUMQUEST = 3
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H55.TXT"
     DATAHEADING$ = "Physical - Abdominal Exam"
     PAGEOF$ = "5"
     OFPAGE$ = "5"
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 34)
       question$(i) = QUESTIONS$(i + 34)
```

```
  GraphFlag(i) = 0
  SKIPBLANK(i) = 1
NEXT i

Choices$(1, 1) = "Present           "
Choices$(1, 2) = "Absent            "
VariablePtr(1) = 144
MULTIP(1) = 0
None(1) = 0

Choices$(2, 1) = "Present           "
Choices$(2, 2) = "Absent            "
VariablePtr(2) = 146
MULTIP(2) = 0
None(2) = 0

Choices$(3, 1) = "Normal             "
Choices$(3, 2) = "Mass Felt          "
Choices$(3, 3) = "Left Tender        "
Choices$(3, 4) = "Right Tender       "
Choices$(3, 5) = "General Tenderness "
VariablePtr(3) = 148
MULTIP(3) = 1
None(3) = 1

IF sex$ = FEMALE$ THEN
  'increase width of cursor
  Choices$(1, 1) = Choices$(1, 1) + " "
  NUMQUEST = 4
  VariablePtr(4) = 153
  Numresp%(4) = QUESTPTR%(4 + 34)
  question$(4) = QUESTIONS$(38)
  GraphFlag(4) = 0
  SKIPBLANK(4) = 1
  MULTIP%(4) = 1
  None(4) = 1
  Choices$(4, 1) = "Normal             "
  Choices$(4, 2) = "Right Tenderness   "
  Choices$(4, 3) = "Left  Tenderness   "
  Choices$(4, 4) = "Cervical Tenderness"
  Choices$(4, 5) = "General Tenderness "
  Choices$(4, 6) = "Mass               "
  Choices$(4, 7) = "Blood (Clots)      "
END IF

  CALL DataEntryPage(exitchar$, question$(), Choices$(),
  VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
  GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
  NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
```

```
      IF exitchar$ = "P" THEN 7000
      IF exitchar$ = "X" THEN 100
      IF exitchar$ = "N" THEN 100



31000 REM  Primary Selection routine
      CALL SetTrainingColors(TRAINING)
      IF TRAINING = 0 THEN GOTO 32000

31004 REM Training Option Page

      'check for training case data file
      IF NOT Exists%("ABDTRN.DAT") THEN
SCREEN 0
   CLS
LOCATE 10, 10
PRINT "The Training Module is not available."
LOCATE 11, 10
PRINT "The data file for the training cases is not present."
TRAINING = 0
CALL TextPause
CALL SetTrainingColors(TRAINING)
GOTO 32000
      END IF
      Choices$(1, 1) = "Easy Case Narrative  "
      Choices$(1, 2) = "Harder Case Narrative"
      Choices$(1, 3) = "Enter DATA           "
      Choices$(1, 4) = "Exit Training Module "

      ' New method for menu
      NR% = 1
      resplength% = 4
      DATAHEADING$ = "Abdominal Pain Diagnosis Training Module" +
       VERSION$
      menuheading$ = "Training Options"
      HELPFILE$ = "HPT12.DAT"

      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

      '  Branch Depending on User's Selection (Case Narrative,
      '   Enter Data, Exit).
      SELECT CASE NR%
CASE 1, 2
  CALL narrative(NR, CASENUM, THECASE%())
CASE 3
  CALL ModifyNarrative(CASENUM%, escflag%)
  IF escflag <> 1 THEN
```

```
    CALL LoadTrainingCase(CASENUM, THECASE%())
    IF davidflag% = 1 THEN
      FOR i = 1 TO 190: VARIABLE%(i) = THECASE%(i): NEXT i
    END IF
    SSN$ = "000-00-0000"
    GOTO 30
  END IF
CASE 4
  TRAINING = 0
  CALL SetTrainingColors(TRAINING)
CASE ELSE
      END SELECT
      GOTO 31000


32000 REM  Primary Selection routine for main program.
32004 REM Main Option Page
      Choices$(1, 1) = "Real Case          "
      Choices$(1, 2) = "Simulated Case     "
      Choices$(1, 3) = "Training Module    "
      Choices$(1, 4) = "Last Real Case     "
      Choices$(1, 5) = "Last Simulated Case"
      Choices$(1, 6) = "Instructions - HELP"
      Choices$(1, 7) = "Generate SF600     "
      Choices$(1, 8) = "Display Treatment  "
      Choices$(1, 9) = "Exit Program       "

      ' New method for menu
      NR% = 1
      resplength% = 9
      DATAHEADING$ = "Abdominal Pain Diagnosis Program" + VERSION$
      menuheading$ = "Main Options"
      HELPFILE$ = "HP12.DAT"
      DO
CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

'             Branch to Main Option Selected.
'       ON NR GOTO 32320, 32330, 32340, 32350, 32350, 32600, 32800, 32900
SELECT CASE NR%
CASE 1                        ' Real Case
32320     SIMULATE = 1
  GOSUB 60135
  GOTO 30
CASE 2                        ' Simulated Case
32330     SIMULATE = 0
  GOSUB 60135
  SSN$ = "000-00-0000"
  GOTO 30
```

```
CASE 3                        ' Training Module
32340     GOSUB 60135
  ' SSN$ = "000-00-0000"
  TRAINING = 1
  CALL SetTrainingColors(TRAINING)
  GOTO 31000: REM TRAINING PROGRAM ROUTINE
CASE 4, 5                     'Last Real and Simulated Cases
  STFLAG = 0
32350     IF NR = 4 THEN
    filnam$ = "REAL.DAT"
    SIMULATE = 1
    SimulateString$ = "real"
  ELSE
    filnam$ = "SIMUL.DAT"
    SIMULATE = 0
    SimulateString$ = "simulated"
  END IF
  ' dummy value
  whichcase = 0
  ' open case, get data
  CALL GetCase(filnam$, whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
  IF whichcase% = 0 THEN
    SCREEN 0, 1, 3, 3
    CLS
    infostring$ = "No " + SimulateString$ + " cases have been saved."
    CALL LocateCenter(10, infostring$)
    PRINT infostring$;
    CALL TextPause
    SCREEN 0, 1, 0, 0
  ELSE
    '  set lookatfemale variable
    IF HMDX% > 7 THEN
      lookatfemale = 1
    END IF
    GOTO 100
  END IF
CASE 6                        ' Help - instructions
32600     filnam$ = "HELP.DAT"
  thispage% = VideoPage%
  CALL DisplayEncryptedFile(filnam$, thispage%)
  GOTO 32004
32800   CASE 7                        ' SF600 generation routine
  CALL SF600(BOAT1$, BOAT2$, HMNAM$, HMSSN$)
  CALL SetScreenMode(ScrnMode)
  CLS 0
  GOTO 31000
CASE 8                      ' Call treatment protocol routine
  '  User selects option to Display Treatment Protocol.
  '  Tell routine that pt is female so that all protocols are shown.
  Ptsex$ - FEMALE$
  CALL TXMenu(MAXNUM, Ptsex$)
  GOTO 31000
CASE 9                         ' Exit Program.
32900     CLS
  SCREEN 0, 1, 0, 0
   CLS
   END
CASE ELSE

END SELECT
' loop forever.  Will break out of loop when needed.
      LOOP UNTIL 1 - 2


50000 IF STFLAG <> 0 AND TRAINING - 0 THEN
CALL SelectDatabase(lookatfemale, VARIABLE())
      END IF



'              This portion of the program calculates diagnostic
       probabilities.
      REM Initial probabilities - DIV 10^25 to keep from getting an
       OVERFLOW ERROR.
```

a priori's

```
50020 'DATA 1.8D-25,5.4D-25,0.3D-25,0.01D-25,0.5D-25,0.3D-25,1.6D-25
     FOR i - 1 TO 7
       PROB#(i) - APRIORI#(i)
     NEXT i
     'PROB#(1) - 1.8D-25
     'PROB#(2) - 5.4D-25
     'PROB#(3) - 3D-26
     'PROB#(4) - 1D-27
     'PROB#(5) - 5D-26
     'PROB#(6) - 3D-26
     'PROB#(7) - 1.6D-25


'need: PROB#(),TFLAG,TRAINING,VARIABLE(),THECASE%(),BAYES!(),exitcode
'  have an exitcode and can remove several vars
       (TFLAGIF,TFLAGCASE,NUMCOUNT

      SCREEN 0, 1, 0, 0: CLS
      NUMCOUNT - 0
      REM All the checks for enough DATA entered can go here.
      REM Probabilities computed here
      TFLAG - 0: TFLAGCASE - 0: TFLAGIF - 0
      FOR i - 1 TO 152
```

```
IF TRAINING = 1 THEN
  IF THECASE%(i) = 1 THEN
    TFLAGCASE = TFLAGCASE + 1
    IF VARIABLE%(i) = THECASE%(i) THEN
      TFLAGIF = TFLAGIF + 1
    END IF
  END IF
  IF VARIABLE%(i) = THECASE%(i) THEN
    TFLAG = TFLAG + 1
  END IF
END IF
IF VARIABLE%(i) = 1 THEN
  NUMCOUNT = NUMCOUNT + 1
  FOR j = 1 TO 7
    PROB#(j) = PROB#(j) * BAYES!(i, j)
  NEXT j
END IF
      NEXT i

'NOTE NOTE !!!!!!!!!!!!! the next line is just for Ellen's version for
'                        entering data.
      IF davidflag% = 1 THEN GOTO 51040

51025 IF TRAINING = 1 THEN
 IF TFLAGIF > .75 * TFLAGCASE THEN
   GOTO 51040
  ELSE
   GOTO 51036
 END IF
       END IF

      IF NUMCOUNT > 32 THEN 51040
      PRINT "Insufficient DATA has been entered for accurate diagnosis."
      PRINT "Please enter more DATA."
      GOTO 51038
51036 PRINT "You have missed too many items.  Are you sure you have the
       right case?"
51038 LOCATE 25, 15
      CALL SetColor(infocolor)
      PRINT " To return to main menu, press any key";
      CALL SetColor(forecolor)
      CALL GetKey(a$)
      GOTO 100


'            Calculate final probabilitiies here (FINPROB).  Determine
' the disease (MAXNUM) with the greatest probability (MAXPROB).
```

Male calc

need 32 [illegible] replies to make Dx

```
51040 CALL ComputeFinalProbs(NUMDISEASES%, MAXNUM%, MAXPROB%, PROB#(),
       FINPROB#())
                                                       NSAP + "other"
      FINPROB#(2) - FINPROB#(2) + FINPROB#(7) <-
      IF MAXNUM - 7 THEN MAXNUM - 2  (other -> NSAP)
      IF FINPROB#(2) > MAXPROB THEN MAXNUM - 2
      IF MAXNUM - 2 THEN MAXPROB - FINPROB#(2)
      MALEMAXNUM% - MAXNUM

      'compute GYN probabilities.
      'skip all of the stuff about training checks and number of
       questions
      ' answered.
     REM Initial probabilities -  DIV 10^25 to keep from getting an
       OVERFLOW ERROR.
     FOR i - 1 TO 7
       FEMPROB#(i) - FEMAPRIORI#(i) set Di:
     NEXT i                              apriori


      IF sex$ - FEMALE$ THEN
FOR i - 1 TO 174                                    femcalc
  IF VARIABLE(i) - 1 THEN                      ↗
    FOR j - 1 TO 7
      IF FEMBAYES!(i, j) <> 0 THEN
        FEMPROB#(j) - FEMPROB#(j) * FEMBAYES!(i, j)
      END IF
    NEXT j
'  END IF
NEXT i

CALL ComputeFinalProbs(NUMDISEASES%, FEMMAXNUM%, FEMMAXPROB%,
       FEMPROB#(), FEMFINPROB#())
FEMMAXNUM% - FEMMAXNUM% + 7
      END IF

      IF sex$ - FEMALE$ AND lookatfemale - 1 THEN
MAXNUM - FEMMAXNUM%
      END IF

      REM PAGE 0 -- HM EVALUATION
'            Skip this page if the case is not new of if no response
' changes have been made.
      IF STFLAG - 0 THEN 52000


      'How about here is where the HM is asked re which database to use.

      'if female and appropriate responses marked, then
      ' ask which database to use.
      'On this page, can list all of the disease considered.
      'Will then need to move agree/disagree part to next screen.

      'Can have toggle on diagnostic summary page to switch back and
       forth
      ' between abd and gyn databases.


51100 IF lookatfemale - 0 THEN
Choices$(1, 1) - "APPENDICITIS                 "
Choices$(1, 2) - "NON-SPECIFIC ABDOMINAL PAIN"
Choices$(1, 3) - "RENAL COLIC                  "
Choices$(1, 4) - "PERFORATED DUODENAL ULCER    "
Choices$(1, 5) - "CHOLECYSTITIS                "
Choices$(1, 6) - "SMALL BOWEL OBSTRUCTION      "
Choices$(1, 7) - "OTHER                        "
resplength% - 7
menuheading$ - "Your Diagnosis"
      ELSE
Choices$(1, 1) - "APPENDICITIS                 "  'dx #  8
Choices$(1, 2) - "NON-SPECIFIC ABDOMINAL PAIN"  'dx #  9
Choices$(1, 3) - "PELVIC INFLAMMATORY DISEASE"  'dx # 10
Choices$(1, 4) - "URINARY TRACT INFECTION      "  'dx # 11
Choices$(1, 5) - "OVARIAN CYST                 "  'dx # 12
Choices$(1, 6) - "ECTOPIC PREGNANCY            "  'dx # 13
Choices$(1, 7) - "INCOMPLETE ABORTION          "  'dx # 14
Choices$(1, 8) - "OTHER                        "  'dx # 15
resplength% - 8
menuheading$ - "Your Gynecological Diagnosis"
      END IF

      ' New method for menu
      NR% - 1
      DATAHEADING$ - "Corpsman's Diagnosis Entry Page"
      HELPFILE$ - "HP00.DAT"

      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

'             Store Corpsman's Diagnosis variable HMDX (# of the
' Diagnosis) and HMDX$ (name of the diagnosis).
      IF lookatfemale - 0 THEN
HMDX - NR
      ELSE
HMDX - NR + 7
      END IF
      IF HMDX - 7 OR HMDX - 15 THEN
```

```
51250   LOCATE 16, 1: PRINT SPACE$(75);
LOCATE 16, 1: PRINT "Enter name of other diagnosis: ";
LINE INPUT OTHER$
IF OTHER$ = "" THEN
  SOUND 200, 1
  GOTO 51100
END IF
LOCATE 19, 10
PRINT "This database does not consider "; OTHER$;
LOCATE 20, 10
PRINT "in the differential diagnosis of acute abdominal pain."
CALL TextDxPause
GOTO 52000
      END IF

51350 IF HMDX = MAXNUM THEN
LOCATE 16, 1
PRINT "   The program-generated probabilities AGREE with your
       provisional";
LOCATE 17, 1: PRINT "diagnosis.";
CALL TextDxPause
GOTO 52000
      END IF

      ' Put all of this on another page.

      ' Check for questions to recheck if original database selected.
      IF lookatfemale = 0 THEN
CALL CompareAbdDXes(COMPAR%(), VARIABLE%(), MAXNUM%, HMDX%, QUESTPTR%(),
       QUESTIONS$())
      ELSE
CALL CompareFemDXes(HMDX%, MAXNUM%)
      END IF

52000 REM PAGE 14 -- Diagnostic Summary Page
52080 CALL SetScreenMode(ScrnMode)
      LOCATE 1, 18: headingPRINT ("Diagnostic Summary Page")
      LOCATE 1, 59: PRINT "SSN:  "; SSN$;
      LOCATE 2, 59: PRINT "Time: "; STARTIME$;
      LOCATE 3, 59: PRINT "Date: "; STARTDATE$;
      IF TRAINING = 1 THEN
LOCATE 5, 59: PRINT "Score:"; INT((TFLAG / 152) * 100); "%";
IF TFLAG = 152 THEN PRINT "!!!";
SELECT CASE HMDX
  CASE 1
   HMChosenDX$ = "APPEND"
  CASE 2
   HMChosenDX$ = "NONSAP"
  CASE 3
```

```
   HMChosenDX$ = "RENCOL"
  CASE 4
   HMChosenDX$ = "PERFDU"
  CASE 5
   HMChosenDX$ = "CHOLE"
  CASE 6
   HMChosenDX$ = "SMBOBS"
  CASE ELSE
   HMChosenDX$ = OTHER$
END SELECT
LOCATE 6, 59: PRINT "HM dx: "; HMChosenDX$;
LOCATE 17, 59: PRINT "TRAINING CASE";
LOCATE 18, 64: PRINT "#"; CASENUM;
      ELSE
LOCATE 17, 59
IF SIMULATE = 0 THEN
  PRINT "SIMULATED CASE";
  ELSE
  PRINT "REAL CASE";
END IF
      END IF
      LOCATE 20, 59: PRINT BOAT1$;
      LOCATE 21, 59: PRINT BOAT2$;

      IF Vertbits = 14 THEN
WINDOW SCREEN (0, 0)-(639, 199)
      END IF
      IF sex$ = FEMALE$ AND lookatfemale = 1 THEN
MAXNUM = FEMMAXNUM
CALL FemaleGraph(FEMFINPROB#())
      ELSE
MAXNUM = MALEMAXNUM
CALL MaleGraph(FINPROB#())
      END IF



'      LINE (0, 150)-(449, 150), hpframecolor
'      LINE (0, 28)-(449, 28), hpframecolor, , &HF00F
'      LINE (0, 82)-(449, 82), hpframecolor, , &HF00F


      IF TRAINING = 0 THEN
resplength% = 5
HELPFILE$ = "HP10.DAT"
Choices$(1, 1) = "CHANGE INPUT DATA"
Choices$(1, 2) = "ANOTHER DIAGNOSIS"
Choices$(1, 3) = "DISPLAY TREATMENT"
Choices$(1, 4) = "DISPLAY H & P    "
```

```
Choices$(1, 5) = "END INTERACTION  "
IF sex$ = FEMALE$ THEN
  resplength% = resplength% + 1
  IF lookatfemale = 1 THEN
    Choices$(1, resplength%) = "DISPLAY ABD DX'ES"
  ELSE
    Choices$(1, resplength%) = "DISPLAY GYN DX'ES"
  END IF
END IF
      ELSE
resplength% = 6
HELPFILE$ = "HPT10.DAT"
Choices$(1, 1) = "CHANGE INPUT DATA"
Choices$(1, 2) = "ANOTHER CASE      "
Choices$(1, 3) = "DISPLAY TREATMENT"
Choices$(1, 4) = "DISPLAY H & P     "
Choices$(1, 5) = "END INTERACTION  "
Choices$(1, 6) = "SHOW MISSED ITEMS"

      END IF


      NR% = 1
      menuheading$ = "Options"

    DO
      CALL MenuSummaryPage(10, 59, NR%, resplength%, exitchar$,
       menuheading$, Choices$(), HELPFILE$)
      ' Help text is written in text mode.  Have to redraw whole screen.
      IF exitchar$ = "?" THEN EXIT DO

' for testing only
IF davidflag% = 1 THEN
  LOCATE 1, 1
  PRINT USING "###### ###### ######"; FRE(0), FRE(""), FRE(-1)
END IF


'52440 ON NR GOTO 52450, 52501, 54000, 53000, 52500, 55000
      SELECT CASE NR%
CASE 1                    ' Change Input Data
  ' save it only if real case and changes have been made
  IF (TRAINING = 0 AND STFLAG = 1 AND SIMULATE = 1) THEN
    filnam$ = "REAL.DAT"
    CALL PutCase(0, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
    STFLAG = 0
  END IF
    GOTO 100
```

```
CASE 2                  ' Another Dx/Case
  IF TRAINING = 0 AND STFLAG = 1 THEN
    ' Save both real and simulated cases,
    '  if not training, and changes were made.
    CALL PutCase(0, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
  END IF
  CALL ResetVariables(VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
  STFLAG = 0
  lookatfemale = 0      'may not be necessary
  GOTO 31000

CASE 3                  ' Display Treatment
  '  User selects option to Display Treatment Protocol.
  CALL TXMenu(MAXNUM, sex$)
   CLS
  GOTO 52000

CASE 4                  ' Display H&P
  CALL DisplayHP(TRAINING, SIMULATE, SSN$, STARTIME$, STARTDATE$,
       VARIABLE%())
  GOTO 52000

CASE 5                  ' End
  IF TRAINING = 0 AND STFLAG = 1 THEN
    'save main real or simulated cases if changes made.
    CALL PutCase(0, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
  END IF
  SCREEN 0, 1, 0, 0
   CLS
   END

CASE 6                  ' Show Missed Items
  IF TRAINING = 1 THEN
    CALL DisplayMissedHP(SSN$, STARTIME$, STARTDATE$, VARIABLE%(),
       THECASE%())
    GOTO 52000
  ELSE                  ' Should only get here with real case if female
    IF lookatfemale = 1 THEN
      CALL MaleGraph(FINPROB#())
      Choices$(1, resplength%) = "DISPLAY GYN DX'ES"
      MAXNUM = MALEMAXNUM
      lookatfemale = 0
    ELSE
      CALL FemaleGraph(FEMFINPROB#())
      Choices$(1, resplength%) = "DISPLAY ABD DX'ES"
   MAXNUM = FEMMAXNUM
   lookatfemale = 1
  END IF
 END IF
CASE ELSE
   END SELECT
  LOOP UNTIL NR% <> 6 AND sex$ <> FEMALE$
  GOTO 52000

60000 REM READ IN BAYESIAN DATA

   'RESTORE 61000
   'FOR i = 1 TO 152
   'FOR j = 1 TO 7: READ BAYES!(i, j): NEXT j: NEXT i
   CALL UnPackDatabase("REGABD.DAT", BAYES!(), APRIORI#(), 7, 152)

   'RESTORE 62000
   'FOR i = 1 TO 174
   'READ dumDavid
   'READ dumDavid1$
   'FOR j = 1 TO 7: READ FEMBAYES!(i, j): NEXT j: NEXT i
   CALL UnPackDatabase("FEMABD.DAT", FEMBAYES!(), FEMAPRIORI#(), 7,
    174)

   RESTORE 60300
   FOR i = 1 TO 44: READ QUESTIONS$(i): NEXT i
   RESTORE 60410
   FOR i = 1 TO 44: READ QUESTPTR%(i): NEXT i
   RESTORE 60500
   FOR i = 1 TO 6
FOR j = 1 TO 6
 IF i <> j THEN
  FOR N = 1 TO 15
   READ COMPAR%(i, j, N)
   IF COMPAR%(i, j, N) < 1 THEN EXIT FOR
  NEXT N
 END IF
NEXT j
   NEXT i

'  VARIABLE(I) = 0 when the response has not been entered
' and VARIABLE(I) = 1 when the response has been entered.
60135 CALL ResetVariables(VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
    STARTDATE$)
   RETURN

   'QUESTIONS$()
60300 DATA SITE OF PAIN(onset),SITE OF PAIN(now)
   DATA TYPE OF PAIN,SEVERITY OF PAIN,AGGRAVATING FACTORS
```

```
      DATA PROGRESS OF PAIN,DURATION OF PAIN,RELIEVING FACTORS
      DATA NAUSEA,VOMITING,BOWELS,APPETITE,JAUNDICE,URINATION
      DATA PREVIOUS INDIGESTION,PREVIOUS SIMILAR PAIN,PREVIOUS SURGERY
      DATA PREVIOUS ILLNESS,TAKING MEDICATIONS
      DATA "TEMPERATURE (F):","PULSE RATE:","BP (Systolic):","BP
       (Diastolic):"
      DATA "MOOD:","COLOR:","WBC COUNT:"
      DATA "INSPECTION:","SCARS:","GUARDING:","RIGIDITY:","BOWEL
       SOUNDS:"
      DATA "DISTENTION:","MASSES:"
      DATA TENDERNESS,"MURPHY'S SIGN:",REBOUND TENDERNESS,RECTAL EXAM
      DATA "VAGINAL EXAMINATION"
      DATA PERIODS,"LAST PERIOD","VAGINAL DISCHARGE",PREGNANCY
      DATA "FAINT/DIZZY","PREV GYN HISTORY"
      'QUESTPTR%()
60410 DATA 13,13,3,2,6,3,4,6,2,2,5,2,2,5,2,2,2,2,2,4,3,3,3
      DATA 3,5,5,3,2,2,2,3,2,2,13,2,2,5,7,4,2,2,3,2,2
      'COMPAR%()
60500 DATA 13,19,26,32,38,42,43,50,70,107,120,133,146,151,0
      DATA 26,38,40,42,43,74,107,120,133,146,151,0
      DATA 19,26,40,102,107,115,133,151,0
      DATA 26,40,42,43,107,133,145,146,151,0
      DATA 26,38,40,42,43,65,102,107,120,133,146,151,0
      DATA 37,47,71,105,121,147,148,0
      DATA 40,74,0
      DATA 37,40,48,102,105,115,148,0
      DATA 37,40,65,145,133,0
      DATA 48,65,102,0
      DATA 21,22,34,35,39,41,47,75,76,105,106,107,141,142,147
      DATA 15,19,20,28,41,79,100,121,135,147,148,0
      DATA 21,22,34,35,39,75,76,102,105,115,128,0
      DATA 21,22,34,35,75,76,141,142,145,0
      DATA 21,22,34,35,41,65,75,76,128,140,0
      DATA 11,15,24,28,41,82,106,131,144,148,0
      DATA 32,33,39,41,47,66,83,103,106,121,127,140,147,148,0
      DATA 21,22,34,35,41,75,76,141,142,0
      DATA 15,20,28,33,42,55,103,106,117,140,145,0
      DATA 15,20,28,33,38,41,65,117,0
      DATA 15,20,28,33,38,0
      DATA 11,24,38,41,63,66,79,120,131,144,0
      DATA 19,32,39,49,66,83,103,106,127,139,0
      DATA 15,20,28,39,74,104,135,0
      DATA 11,24,38,107,131,144,0
      DATA 11,15,24,28,74,81,120,131,144,0
      DATA 19,32,105,115,135,148,0
      DATA 11,24,102,105,115,131,144,0
      DATA 19,32,39,66,80,83,103,127,140,0
      DATA 19,32,39,83,103,127,140,145,0
```

```
REM $STATIC
SUB CompareFemDXes (HMDX%, MAXNUM%)
'    This routine displays the Best Questions to Ask if the HM's
'    diagnosis differs from the computer's (GYN database only).
'    See CompareAbdDXes for abd database.

  CLS
  TheHMDX = HMDX - 7
  TheMAXNUM = MAXNUM - 7
  row = 10
  col = 10

  filnam$ = "bestques.dat"
  IF NOT Exists%(filnam$) THEN
   EXIT SUB
  END IF

  ' draw window here.

  IF TheHMDX = TheMAXNUM THEN
   'the dx'es agree.  already commented on in ABDX.BAS
  ELSEIF TheHMDX = 8 THEN
   'other dx not considered.  already commented on in ABDX.BAS
  ELSE
   'create a number such that the smaller of TheHMDX or MAXNUM is in
   'the tens' position and the other is in the unit's position.
   IF TheHMDX < TheMAXNUM THEN
combinednumber = TheHMDX * 10 + TheMAXNUM
   ELSE
combinednumber = TheMAXNUM * 10 + TheHMDX
   END IF


   filenum = FREEFILE
   OPEN filnam$ FOR INPUT AS filenum


   foundflag = 0
   stringnumber = 0
   DO WHILE stringnumber < combinednumber
LINE INPUT #filenum, quest$
'decipher string
CALL decipher(quest$)
stringnumber = VAL(LEFT$(quest$, 2))
   LOOP
   IF stringnumber = combinednumber THEN
numofquests = VAL(MID$(quest$, 4, 2))
questlength = VAL(MID$(quest$, 7, 2))
col = (80 - questlength) \ 2
```

```
foundflag = 1
LOCATE 3, 5: PRINT "  At this time the computer-generated probabilities
       DO NOT AGREE with";
LOCATE 4, 5: PRINT "your preliminary diagnosis.  The following
       categories are particularly";
LOCATE 5, 5: PRINT "useful in differentiating your diagnosis from the
       others.  It may be";
LOCATE 6, 5: PRINT "helpful to review your input in these areas and make
       any changes you";
LOCATE 7, 5: PRINT "consider appropriate.";

    DATAHEADING$ = "Best Questions to Ask"
    CALL LocateCenter(1, DATAHEADING$)
    CALL headingPRINT(DATAHEADING$)

'        CALL frame(row , col - 1, numofquests, questlength, frametyp)
CALL BoxSelections(row, col - 1, numofquests, questlength)

FOR i = 1 TO numofquests
  LINE INPUT #filenum, quest$
  'decipher string
  CALL decipher(quest$)
  LOCATE row, col
  PRINT quest$
  row = row + 1
NEXT i
      END IF
      CLOSE #filenum
      IF foundflag = 0 THEN
LOCATE row, col
PRINT "    At this time, the computer-generated probabilities DO NOT
       AGREE with";
PRINT "your preliminary diagnosis.  However, in this case, there are
       no";
PRINT "specific questions to ask which would differentiate your
       preliminary";
PRINT "diagnosis from the current program-generated diagnosis.";

      END IF
      CALL TextPause
    END IF
END SUB

SUB GetCase (filnam$, whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
'       This routine opens the .DAT file filnam$ and retrieves the
'       appropriate case 'whichcase'.
'       NOTE:  if whichcase = 0 then the case retrieved is the last
'               stored case.
'              If it is returned as 0, then file did not exist.

      'Globals
      '       - MALE$   (constant in include file )
      '       - FEMALE$ (constant in include file )


      OPEN filnam$ FOR RANDOM AS #1 LEN = 128
      '  File format for variables.
      FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$
'  If no previous case has been entered, beep, close
' the file and request more user input.
N% = LOF(1) / 128
IF N% = 0 THEN
  CLOSE #1
  whichcase% = 0
  EXIT SUB
END IF
IF whichcase% = 0 OR whichcase% > N% THEN
  whichcase% = N%
END IF

'  Get a record from the file.
GET #1, whichcase%

'   Put case data into variables.
SSN$ = LSSN$: AGE$ = LAGE$: a$ = LVAR$: OTHER$ = LOTH$
STARTIME$ = LTIM$: STARTDATE$ = LDAT$
HMDX = CVI(LHMD$): SIMULATE = CVI(LSIM$)
'   Close the file.
      CLOSE #1
      ' unpack data in a$ into VARIABLE%()
      CALL UnPackArray(a$, VARIABLE%())
      ' update sex
      IF VARIABLE%(2) = 1 THEN
sex$ = FEMALE$
      ELSE
sex$ = MALE$
      END IF


END SUB

REM $DYNAMIC
SUB InitializeColors (graphmode$, monmode$)

'GLOBAL - GRAPHICS, ScrnMode, forecolor, backcolor, infocolor
```

```
'GLOBAL - textcolor, questioncolor, responsecolor, hpresponsecolor
'GLOBAL - graphcolor, helpcolor, hpframecolor, bargraph()
'GLOBAL - red,green, brown, white, black, yellow
'GLOBAL - ltred, ltgreen, ltcyan, ltmagenta, ltblue
'GLOBAL - MAINHEADINGCOLOR%, MAINFRAMECOLOR%, MAINFRAME%
'GLOBAL - TRAININGHEADINGCOLOR%, TRAININGFRAMECOLOR%, TRAININGFRAME%


'       Set up graphics mode default (checked for CGA or EGA in
      graphmode$)
IF graphmode$ = "C" THEN
  GRAPHICS = 2
  ScrnMode = 2
  IF monmode$ = "C" THEN
    '  CGA and color monitor
    MAINHEADINGCOLOR% = red
    MAINFRAMECOLOR% = green
    MAINFRAME% = 1            'select single frame for default pages.
    TRAININGHEADINGCOLOR% = brown
    TRAININGFRAMECOLOR% = 1
    TRAININGFRAME% = 2            'select double frame for training
       pages.
    forecolor = white
    backcolor = black
    textcolor = white
    questioncolor = yellow
    responsecolor = white
    hpresponsecolor = white
    hpframecolor = 1
    infocolor = green
    graphcolor = white
    helpcolor = white
    bargraph(1) = 1
    bargraph(2) = 1
    bargraph(3) = 1
    bargraph(4) = 1
    bargraph(5) = 1
    bargraph(6) = 1
    bargraph(7) = 1
  ELSE
    '   CGA, but no color monitor
    MAINHEADINGCOLOR% = hiwhite
    MAINFRAMECOLOR% = white
    MAINFRAME% = 1            'select single frame for default pages.
    TRAININGHEADINGCOLOR% = white
    TRAININGFRAMECOLOR% = white
    TRAININGFRAME% = 2            'select double frame for training
       pages.
    forecolor = white
  backcolor = black
  textcolor = white
  questioncolor = white
  responsecolor = white
  hpresponsecolor = white
  hpframecolor = 1
  infocolor = white
  graphcolor = white
  helpcolor = white
  bargraph(1) = 1
  bargraph(2) = 1
  bargraph(3) = 1
  bargraph(4) = 1
  bargraph(5) = 1
  bargraph(6) = 1
  bargraph(7) = 1
END IF
ELSE
GRAPHICS = 9
ScrnMode = 9
IF monmode$ = "C" THEN
  '  EGA and color monitor
  MAINHEADINGCOLOR% = red
  MAINFRAMECOLOR% = green
  MAINFRAME% = 1            'select single frame for default pages.
  TRAININGHEADINGCOLOR% = brown
  TRAININGFRAMECOLOR% = brown
  TRAININGFRAME% = 2            'select double frame for training
     pages.
  forecolor = white
  backcolor = black
  textcolor = white
  questioncolor = yellow
  responsecolor = white
  hpresponsecolor = white
  hpframecolor = blue
  infocolor = green
  graphcolor = white
  helpcolor = white
  bargraph(1) = ltred
  bargraph(2) = ltgreen
  bargraph(3) = ltcyan
  bargraph(4) = ltmagenta
  bargraph(5) = ltblue
  bargraph(6) = yellow
  bargraph(7) = red
ELSE
  '   EGA, but no color monitor
  MAINHEADINGCOLOR% = hiwhite
```

```
    MAINFRAMECOLOR% = white
    MAINFRAME% = 1             'select single frame for default pages.
    TRAININGHEADINGCOLOR% = white
    TRAININGFRAMECOLOR% = white
    TRAININGFRAME% = 2             'select double frame for training
       pages.
    forecolor = white
    backcolor = black
    textcolor = white
    questioncolor = white
    responsecolor = white
    hpresponsecolor = white
    hpframecolor = white
    infocolor = white
    graphcolor = white
    helpcolor = white
    bargraph(1) = white
    bargraph(2) = white
    bargraph(3) = white
    bargraph(4) = white
    bargraph(5) = white
    bargraph(6) = white
    bargraph(7) = white
  END IF
END IF

END SUB

REM $STATIC
SUB PutCase (whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
'       This routine opens the .DAT file based on SIMULATE and saves the
'       appropriate case in record 'whichcase'.
'       NOTE:  if whichcase = 0 then the case is appended to the
'                stored cases.

      IF SIMULATE = 0 THEN
filnam$ = "SIMUL.DAT"
      ELSE
filnam$ = "REAL.DAT"
      END IF

      OPEN filnam$ FOR RANDOM AS #1 LEN = 128
      '  File format for variables.
      FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$

N% = LOF(1) / 128
```

```
' If 0, then append case to end of stored cases.
IF whichcase% = 0 THEN
  whichcase% = N% + 1
END IF

CALL PackArray(a$, VARIABLE%())

'   Left justifies the variables in the field and moves the
' DATA into a random buffer file.
LSET LSSN$ = SSN$: LSET LAGE$ = AGE$: LSET LVAR$ = a$
LSET LOTH$ = OTHER$: LSET LTIM$ = STARTTIME$
LSET LDAT$ = STARTDATE$: LSET LHMD$ = MKI$(HMDX)
LSET LSIM$ = MKI$(SIMULATE): LSET LNUM$ = MKI$(MAXNUM)
LSET LPRO$ = MKI$(MAXPROB)

'  Save the record.
PUT #1, whichcase%

      CLOSE #1

END SUB

 SUB SelectDatabase (lookatfemale, VARIABLE())
      'Check if female and ask which database if appropriate responses
      ' have been entered.

      'GLOBAL - frametype
      '       - FEMALE$ (constant in include file )

      SHARED sex$

      DIM checkvarlocation(6), Choices$(1, 2)

       lookatfemale = 0        'ABD database as a default for females
                'not meeting criteria and all males.

      IF sex$ = FEMALE$ THEN

checkvarlocation(1) = 32  ' Site of Pain at present - CENTRAL
checkvarlocation(2) = 29  ' Site of Pain at present - LOWER
checkvarlocation(3) = 26  ' Site of Pain at present - RLQ
checkvarlocation(4) = 27  ' Site of Pain at present - LLQ
checkvarlocation(5) = 34  ' Site of Pain at present - R FLANK
checkvarlocation(6) = 35  ' Site of Pain at present - L FLANK
CriteriaMet = 0
FOR i = 1 TO 6
  IF VARIABLE(checkvarlocation(i)) = 1 THEN
    CriteriaMet = 1
    EXIT FOR
  END IF
NEXT i
SCREEN 0, 1, 0, 0
   CLS
framtyp = frametype
IF CriteriaMet = 1 THEN
  ' female and gyn database looks good.
  CALL frame(7, 9, 3, 60, framtyp)
  LOCATE 8, 10
  PRINT "    Remember the gynacological database is available and is";
  LOCATE 9, 10
  PRINT "especially suitable for this patient.  It compares your case";
  LOCATE 10, 10
  PRINT "with 1,000 similar previous patients.";
  NR% = 2
  ELSE
  ' Female, but regular database looks better.
  CALL frame(7, 15, 4, 48, framtyp)
  LOCATE 8, 16
  PRINT "    At this time, your patient appears to have";
  LOCATE 9, 16
  PRINT "non-gynecological abdominal pain.  However, the";
  LOCATE 10, 16
  PRINT "gynecological database is available as an option";
  LOCATE 11, 16
  PRINT "on the Diagnostic Summary Page.";
  NR% = 1
END IF
  CALL TextContinuePrompt
  CALL GetKey(a$)
  menuheading$ = "Select Database"
  Choices$(1, 1) = "General Abdominal Database"
  Choices$(1, 2) = "Gynecological Database    "
  HELPFILE$ = "Habgy.dat"
  menurow = 16
  menucol = Centered(Choices$(1, 1))
  resplength% = 2
  LOCATE 25, 5
  PRINT SPACE$(70);
    DO
    CALL MenuSummaryPage(menurow, menucol, NR%, resplength%, exitchar$,
       menuheading$, Choices$(), HELPFILE$)
  LOOP UNTIL exitchar$ = ""
  PRINT "Choose desired database"
  IF NR% = 1 THEN
    lookatfemale = 0
  ELSE
    lookatfemale = 1
  END IF
```

```
      END IF

      ERASE checkvarlocation, Choices$

END SUB

REM $DYNAMIC
SUB SetTrainingColors (TRAINING)
'       This routine sets the different display colors between training
'        and other displays.

   ' Global vars  - headingcolor, framecolor, frametype
   ' Global vars  - MAINHEADINGCOLOR%, MAINFRAMECOLOR%, MAINFRAME%
   ' Global vars  - TRAININGHEADINGCOLOR%, TRAININGFRAMECOLOR%,
       TRAININGFRAME%


    IF TRAINING = 0 THEN
      headingcolor% = MAINHEADINGCOLOR%
      framecolor% = MAINFRAMECOLOR%
      frametype = MAINFRAME%
   ELSE
      headingcolor% = TRAININGHEADINGCOLOR%
      framecolor% = TRAININGFRAMECOLOR%
      frametype = TRAININGFRAME%
   END IF

END SUB

REM $STATIC
FUNCTION Translate% (HMDX)
'This routine translates the value of HMDX to a number between 1 and 7,
'the gyn database, otherwise it returns unchanged.
'       HMDX    the variable to modify
'       lookatfemale    flag for female database selection.

  SHARED lookatfemale

  IF lookatfemale = 0 THEN
    tempval = HMDX
  ELSE
    tempval = HMDX - 7
  END IF
  Translate = tempval

END FUNCTION

 SUB TXMenu (MAXNUM, Ptsex$)
```

```
'       This routine displays the Treatment menu. Upon selection, the
'       treatment is displayed.

      SHARED VERSION$

      DIM Choices$(1, 12)
30004 REM  Tx protocol routine
      Choices$(1, 1) = "APPENDICITIS               "
      Choices$(1, 2) = "NON-SPECIFIC ABDOMINAL PAIN"
      Choices$(1, 3) = "RENAL COLIC                "
      Choices$(1, 4) = "PERFORATED DUODENAL ULCER  "
      Choices$(1, 5) = "CHOLECYSTITIS              "
      Choices$(1, 6) = "SMALL BOWEL OBSTRUCTION    "
      IF Ptsex$ = MALE$ THEN
Choices$(1, 7) = "EXIT DISPLAY               "
resplength% = 7
      ELSE
Choices$(1, 7) = "PELVIC INFLAMMATORY DISEASE"  'dx # 10
Choices$(1, 8) = "URINARY TRACT INFECTION    "  'dx # 11
Choices$(1, 9) = "OVARIAN CYST               "  'dx # 12
Choices$(1, 10) = "ECTOPIC PREGNANCY          " 'dx # 13
Choices$(1, 11) = "INCOMPLETE ABORTION        " 'dx # 14
Choices$(1, 12) = "EXIT DISPLAY               " 'dx # 15
resplength% = 12
      END IF

    ' New method for menu
    SELECT CASE MAXNUM
      CASE 0, 7, 15     'none marked, or other dx
NR% = 1
      CASE 8            ' appi in gyn mode
NR% = 1
      CASE 9            'nonsap in gyn mode
NR% = 2
      CASE 10 TO 14     ' other gyn diseases
NR% = MAXNUM - 3
      CASE ELSE
NR% = MAXNUM
    END SELECT

    DATAHEADING$ = "Abdominal Pain Diagnosis Program" + VERSION$
    menuheading$ = "Treatment Summary"
    HELPFILE$ = "HP11.DAT"
    DO
      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
      ' decrypt and print treatment text.
      IF NR% <> resplength% THEN
TXfile$ = "TX" + MID$(STR$(NR%), 2) + ".DAT"
```

```
thispage% = VideoPage%
CALL DisplayEncryptedFile(TXfile$, thispage%)
      END IF
    LOOP UNTIL NR% = resplength%
    ERASE Choices$

END SUB
```

ABDXNARA.BAS

```
DECLARE SUB UnPackArray (PackString$, thearray%())
DECLARE FUNCTION Centered% (s$)
DECLARE FUNCTION Exists% (FIL$)
DECLARE SUB questionPRINT (a$)
DECLARE SUB templatehelp (helpstring$, a$, templatestring$, blankchar$,
       returncode%, errorstring$, errorflag%)
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE FUNCTION centeredlocation% (infostring$)
DECLARE SUB CenterString (infostring$)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB headingPRINT (a$)
DECLARE SUB GetKey (a$)
DECLARE SUB LoadTrainingCase (CASENUM%, THECASE%())
DECLARE SUB InitiateTHELOOP (THELOOP%())
DECLARE SUB NarrativeHelp (WritePage%, VisualPage%)
DECLARE SUB InitializeTrainingCase (NUMCASE%, DataString$)
' This contains the key input and training narrative routines.
DEFINT A-Z

' $INCLUDE: 'include.bas'

FUNCTION Centered% (s$)
'       This function returns the col location for printing the centered
'       string s$

  Centered% = (80 - LEN(s$)) \ 2

END FUNCTION

SUB CenterPrint (row%, TheString$)
'       This routine centers a string on the screen at row and prints
       it.
  CALL LocateCenter(row, TheString$)
  PRINT TheString$;

END SUB

SUB CenterString (infostring$)
'       This routine centers a string on the screen at the current row.
  crow = CSRLIN
  CALL LocateCenter(crow, infostring$)
  PRINT infostring$;
END SUB
```

```
'       Contains: GetKey(A$)                originally @ line   400
'                 narrative(NR, CASENUM,THECASE%() )          31320
'                 NarrativeHelp(WritePage,VisualPage)         30500
'                 LoadTrainingCase(CASENUM,THECASE%() )       31700
'
'
'
' ====================================================================
SUB GetKey (a$)
'       Waits for character input.
'       Originally a subroutine at line 400
    DO
      a$ = INKEY$
    LOOP WHILE a$ = ""
      IF a$ = CHR$(3) THEN STOP
      a$ = UCASE$(a$)
END SUB

' ====================================================================
SUB InitializeTrainingCase (NumOfTrainingCase, DataString$)
'   This routine loads the array TrainingCase() with the desired
'  training case in compacted form.

  'No need to check existence of data file.  It was checked when
       training
  'routine was first entered.

  filename$ = TRAININGCASEFILE$
  filenum = FREEFILE
  OPEN filename$ FOR RANDOM AS filenum LEN = 26
  FIELD #filenum, 26 AS CaseString$
  GET #filenum, NumOfTrainingCase
  DataString$ = CaseString$

  CLOSE #filenum
'       end of InitializeTrainingCase()
END SUB

'=======================================================================
       ====

SUB LoadTrainingCase (CASENUM, THECASE%())

'Gets, decompresses case and places it in THECASE%()
' this should probably be a subprogram, since it is called in 2 places:
'       1. in the narrative printing routine above,
'       2. in the do you want a different case routine below in line
       31880
```

```
'      DIM TrainingCase(12)

      ' get packed case string.
      CALL InitializeTrainingCase(CASENUM, DataString$)

      'Clear case storage array
31710 FOR i = 1 TO NUMBEROFITEMS
THECASE%(i) = 0
      NEXT i

      ' Unpack into THECASE%()
      CALL UnPackArray(DataString$, THECASE%())

'       end of LoadTrainingCase()
END SUB

 SUB LocateCenter (crow, infostring$)
'       This routine computes the location to write the string so that
       it
'       is centered on the 80 col wide screen.

  iscenter = Centered(infostring$)
  LOCATE crow, iscenter
END SUB

SUB ModifyNarrative (CASENUM%, escflag%)
'       This routine allows for the selection of a different case number
'       used in the training program.

CaseHeading$ = "Current Case [  ]"
IF CASENUM% = 0 THEN
  casenumber$ = "__"
ELSE
  casenumber$ = RTRIM$(LTRIM$(STR$(CASENUM%)))
  IF LEN(casenumber$) = 1 THEN casenumber$ = casenumber$ + "_"
END IF
blankstring$ = "_"
mainrow = 10
maincol = 20
SCREEN 0, 1, 0, 0
CLS
    Casehelp$ = "Enter the desired case number. Valid case numbers are
       1|"
Casehelp$ = Casehelp$ + " through 50. Press 'Enter' to select the
       displayed|"
Casehelp$ = Casehelp$ + " case. Press 'Esc' to exit without changing the
       case."
Caseerror$ = " You have to enter a number between 1 and 50, inclusive."
      CALL SetFrameColor
      framtyp = frametype
      CALL frame(mainrow - 1, maincol - 2, 1, 28, framtyp)
      CALL SetNormalColor
      LOCATE mainrow, maincol
      questionPRINT (CaseHeading$)
      LOCATE mainrow, maincol + 14
      PRINT casenumber$;
      noerror = 1
      DO
LOCATE mainrow, maincol + 14
CALL templatehelp(Casehelp$, casenumber$, "%%", blankstring$, rc,
       Caseerror$, errorcode)
'               rc = 0    CR
'                    1    Esc
'                    2    up arrow
'                    3    down arrow
'  Check for Escape. If so, then set exitval, and exit sub.
IF rc = 1 THEN
  escflag = 1
  EXIT SUB
  ELSE
  '  Check for valid number, ie, >0 and <51
  ' VAL stops looking at first non-number or space. ( _ stops it.)
  IF LEFT$(casenumber$, 1) = "_" THEN
    testcase = VAL(RIGHT$(casenumber$, 1))
  ELSE
    testcase = VAL(casenumber$)
  END IF
  IF testcase < 1 OR testcase > 50 THEN
    noerror = 0
    SOUND 200, 1
  END IF
END IF
      LOOP UNTIL noerror = 1
      CASENUM% = testcase
      escflag = 0
END SUB

'----------------------------------------------------------------------------
       ----

SUB NarrativeHelp (WritePage, VisualPage)
' help routine for narrative

30500 SCREEN 0, 1, 3, 3: CLS
      CALL SetColor(framecolor)
      CALL frame(10, 1, 5, 78, frametype)
      CALL SetColor(forecolor)
      LOCATE 11, 3, 0
```

ABDXNARA.BAS (cont'd)

```
30510 PRINT "Push 'N' to go to the next page. (If on last page, will go
       back to"
      LOCATE 12, 3, 0
30515 PRINT "    previous menu.)"
      LOCATE 13, 3, 0
30520 PRINT "Push 'P' to go to the previous page. (If on first page,
       will go"
      LOCATE 14, 3, 0
30522 PRINT "    back to previous menu.)"
      LOCATE 15, 3, 0
30530 PRINT "Push '?' for this help message."
30540 LOCATE 25, 26
      CALL SetColor(infocolor)
      PRINT " To continue, press any key";
      CALL SetNormalColor
30550 CALL GetKey(a$)
30560 SCREEN 0, 1, WritePage, VisualPage
END SUB
```

ABDXONLY.BAS

```
'This module contains routines specific only to ABDX.
DEFINT A-Z
DECLARE SUB SetColor (thecolor%)

' $INCLUDE: 'include.bas'

SUB ABDDrawGraph (WhichOne%)
'       This routine draws the abdominal graph at the proper
       coordinates.
'       WhichOne% can be 1  -  Pain at onset
'                        2  -  Pain at present
'                        3  -  Tenderness
'       XX and YY.
'       Shared variables: graphcolor
'                         YOffsetpict
'                         forecolor
     IF WhichOne% = 2 THEN
       xx = 565
       YY = 100
     ELSE
       xx = 75
       YY = 100
     END IF

     WINDOW SCREEN (0, 0)-(639, 199)
1680 CALL SetColor(graphcolor)
     LINE (xx, YY - YOffsetpict)-(xx, YY - 50)
     LINE (xx, YY + 50)-(xx, YY + YOffsetpict)
     LINE (xx - 50, YY + 20)-(xx - 40, YY - 20), , B
     LINE -(xx, YY - 50)
     LINE -(xx + 40, YY - 20)
     LINE -(xx + 50, YY + 20), , B
     LINE (xx + 40, YY + 20)-(xx, YY + 50)
     LINE -(xx - 40, YY + 20)
     LINE (xx - 40, YY)-(xx - 10, YY)
     LINE (xx + 10, YY)-(xx + 40, YY)
     CIRCLE (xx, YY), 10, , , , .6
     CALL SetColor(forecolor)

END SUB

SUB ABDPaintGraph (VAR%, WhichOne%)
'       This routine paints the appropriate sections of the abdomen
       graph.

'       WhichOne% can be 1  -  Pain at onset
'                        2  -  Pain at present
'                        3  -  Tenderness
'       XX and YY.

     IF WhichOne% = 2 THEN
       xx = 565
       YY = 100
     ELSE
       xx = 75
       YY = 100
     END IF

  SELECT CASE VAR
    CASE 1
      PAINT (xx - 1, YY - 48)
    CASE 2
      PAINT (xx + 1, YY - 48)
    CASE 3
      PAINT (xx - 1, YY + 48)
    CASE 4
      PAINT (xx + 1, YY + 48)
    CASE 5
      PAINT (xx - 1, YY - 48)
      PAINT (xx + 1, YY - 48)
    CASE 6
      PAINT (xx - 1, YY + 48)
      PAINT (xx + 1, YY + 48)
    CASE 7
      PAINT (xx - 1, YY - 48)
      PAINT (xx - 1, YY + 48)
    CASE 8
      PAINT (xx + 1, YY - 48)
      PAINT (xx + 1, YY + 48)
    CASE 9
      PAINT (xx, YY)
    CASE 10
      PAINT (xx - 1, YY - 48)
      PAINT (xx + 1, YY - 48)
      PAINT (xx - 1, YY + 48)
      PAINT (xx + 1, YY + 48)
      PAINT (xx, YY)
    CASE 11
      PAINT (xx - 41, YY - 19)
    CASE 12
      PAINT (xx + 41, YY - 19)
    CASE 12                     'no pain

    CASE ELSE
      'should never get here.
  END SELECT

END SUB

      SUB FemaleGraph (FINPROB#())
'       This routine draws the bar graph using female diseases.

      'uses bargraph()

      DIM dxptr(7)
      DIM DxString$(7)

      hpframecolor2 = bargraph(1)
      ' Clear center of graph
      LINE (1, 15)-(448, 167), 0, BF
      ' Draw frame for bar graph
      LINE (0, 14)-(449, 168), hpframecolor2, B
      ' These are printed inside the box
      LOCATE 5, 54: PRINT "90%";
      LOCATE 12, 54: PRINT "50%";

       dxptr(1) = 2: DxString$(1) = "APPEND"
      dxptr(2) = 10: DxString$(2) = "NONSAP"
      dxptr(3) = 20: DxString$(3) = "PID   "
      dxptr(4) = 28: DxString$(4) = "UTI   "
      dxptr(5) = 34: DxString$(5) = "OVCYST"
      dxptr(6) = 42: DxString$(6) = "ECTOPI"
      dxptr(7) = 50: DxString$(7) = "INC_AB"

      FOR i = 1 TO 7
'        LOCATE 20, 2 + ((i - 1) * 8): PRINT DxString$(i);
        LOCATE 20, dxptr(i): PRINT DxString$(i);

'        xoffset = ((i - 1) * 8 + 1) * 8
        xoffset = (dxptr(i) - 1) * 8
        IF i = 3 OR i = 4 THEN
          x1 = xoffset
          x2 = 23 + xoffset
        ELSE
          x1 = 8 + xoffset
          IF i = 7 THEN
            x2 = 30 + xoffset
          ELSE
            x2 = 31 + xoffset
          END IF
        END IF

        y1 = 150 - (FINPROB#(i) / 100 * 136)
'        y1 = 150 - (95 / 100 * 136)
```

```
        y2 = 150
        LINE (x1, y1)-(x2, y2), bargraph(i), BF
        theprob = INT(FINPROB#(i) * 10)
        theprob = theprob / 10

        LOCATE 21, 2 + (i - 1) * 8: PRINT USING "###.#"; FINPROB#(i);
        PRINT "%";
      NEXT i
      LINE (0, 150)-(449, 150), hpframecolor2
      LINE (0, 28)-(449, 28), hpframecolor2, , &HF00F
      LINE (0, 82)-(449, 82), hpframecolor2, , &HF00F

      ERASE dxptr, DxString$


END SUB

SUB MaleGraph (FINPROB#())
'       This routine draws the bar graph using male diseases.
      'uses hpframecolor%, bargraph()


      DIM DxString$(7)

      DxString$(1) = "APPEND": DxString$(2) = "NONSAP"
      DxString$(3) = "RCOLIC": DxString$(4) = "PERFDU"
      DxString$(5) = "CHOLE": DxString$(6) = "SMBOBS"

      ' Clear center of graph
      LINE (1, 15)-(448, 167), 0, BF
      ' Draw frame for bar graph
      LINE (0, 14)-(449, 168), hpframecolor, B

      ' These are printed inside the box
      LOCATE 5, 54: PRINT "90%";
      LOCATE 12, 54: PRINT "50%";

        'print male abd diseases bar graph and probs.
      FOR i = 1 TO 6
        LINE (19 + ((i - 1) * 72), (150 - (FINPROB#(i) / 100 * 136)))-
       (43 + ((i - 1) * 72), 150), bargraph(i), BF
        LOCATE 20, 2 + ((i - 1) * 9): PRINT DxString$(i);
        LOCATE 21, 2 + ((i - 1) * 9): PRINT USING "###.#"; FINPROB#(i);
        PRINT "%";
      NEXT i
      LINE (0, 150)-(449, 150), hpframecolor
      LINE (0, 28)-(449, 28), hpframecolor, , &HF00F
      LINE (0, 82)-(449, 82), hpframecolor, , &HF00F
```

```
    ERASE DxString$

END SUB
```

ABDXSHAR.BAS

```
DECLARE SUB HPframe ()
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB SetScreenMode (smode%)
DECLARE SUB HelpDataEntry (HLPFIL$, quest%)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB questionPRINT (a$)
DECLARE SUB headingPRINT (a$)
DECLARE SUB responsePRINT (a$)
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB CenterString (infostring$)
DECLARE SUB GetKey (a$)
DECLARE SUB LoadTrainingCase (CASENUM%, THECASE%())
DECLARE SUB NarrativeHelp (WritePage%, VisualPage%)
DECLARE SUB TrainingHPHeading (IBEGIN%, CASENUM%, NR%, NAM$)
DECLARE SUB InitiatePhrase (EasyHard%, PHRASE$())
DECLARE SUB InitiateTHELOOP (THELOOP%())
DECLARE SUB TextContinuePrompt ()
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB mainkeyroutine (exitstring$, quest%, resp%, NUMQUEST%,
       Numresp%(), NUMCOL1QUESTS%, VariablePtr%(), VARIABLE%(),
       MULTIP%(), None%(), actrow%(), GRAPHFLAG%(), offset%, STFLAG%,
       Choices$())
DECLARE SUB UpdateAsterisk (FirstRow%, Firstcol%, NonePtr%, VariPtr%,
       GraphFlg%, NumberofResp%, offset%, VARIABLE%())
DECLARE SUB PutCursor (quest%, resp%, Choices$())
DECLARE SUB DrawGraph (WhichOne%)
DECLARE SUB ABDPaintGraph (VAR%, WhichOne%)
DECLARE SUB ABDDrawGraph (WhichOne%)
DECLARE SUB DisplayHProwcol (row%, col%, sxstrng$)
DECLARE SUB DisplayHPTitle (HP%)
'This module contains routines which although have common names with
'routines in CPDX, etc, have been modified specifically for ABDX.

' $INCLUDE: 'include.bas'

REM $DYNAMIC
DEFINT A-Z
SUB BlankGraph (WhichOne%)
'       This routine blanks a block containing the Abd graph, so that it
'       can be written over fresh.

     IF WhichOne% = 2 THEN
       XX = 565
       YY = 100
     ELSE
```

```
          XX = 75
          YY = 100
        END IF
        WINDOW SCREEN (0, 0)-(639, 199)
        LINE (XX - 51, YY + 51)-(XX + 51, YY - 51), backcolor, BF

END SUB

SUB DataEntryPage (exitchar$, question$(), Choices$(), VariablePtr%(),
       MULTIP%(), SKIPBLANK%(), Numresp%(), None%(), GRAPHFLAG%(),
       VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%, NUMQUEST%,
       DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
'        This is a generic routine to enter data for the H&P pages.
'        Hopefully, it can be used with other programs easily.

'        exitchar$        ch returned upon exit, will be "XNP"
'        question$(x)     list of questions for this page. x=[1-NUMQUEST]
'        Choices$(x,y)    list of responses for each question
'                              x =[1-NUMQUEST],y=[1-numresp(x)]
'        VariablePtr%(x)  list of locations in VARIABLE for each question
'                             x's responses.
'        MULTIP%(x)       =0  only one answer allowed in question x
'                         =1  multiple answers allowed.
'        SKIPBLANK%(x)    number of lines to skip before question x
'                                 used to line up questions side-by-side
'        Numresp%(x)      number of responses for question x
'        None%(x)         response number of any normal/no pain item for
'                                 question x.  Used to reset other
       responses
'                                 to 0 if it is selected.
'        GRAPHFLAG(x)     = 0  --  any question not requiring graphics
'        GRAPHFLAG(x)     = 1  --  Pain at onset question
'        GRAPHFLAG(x)     = 2  --  pain at present question
'        GRAPHFLAG(x)     = 3  --  Tenderness question
'        VARIABLE%()      the data array
'        NUMCOL1QUESTS%   the number of questions in the first column
'        TOPROW%          row of first response of first question
'        TOPCOL%          col of asterisks for first response of first
'                                 question.  Is also the starting col
'                                 for printing out the question.
'        NUMQUEST%        number of questions on page
'        DATAHEADING$     title printed at top of screen
'        PAGEOF$          Number of current page (string form)
'        OFPAGE$          Maximum number of pages in grouping (string
       form)
'        HELPFILE$        Help file for H&P.  Remember, that it is
       sequential
'                         and not encrypted.
'        STFLAG%          =0 if no changes made to data, =1 if CR pressed.
' Global variable
'       FrameFlag = 0 if frmaework has not been drawn
'                 = 1 if framework has been drawn.  No need to re-draw.

  '     initialize stuff
  'initialize location for first response of each question
  actrow(1) = TOPROW + SKIPBLANK(1) + 1
  FOR i = 2 TO NUMCOL1QUESTS
    actrow(i) = actrow(i - 1) + SKIPBLANK(i) + Numresp(i - 1) + 1
  NEXT i
  IF NUMCOL1QUESTS < NUMQUEST THEN
    actrow(NUMCOL1QUESTS + 1) = TOPROW + SKIPBLANK(i) + 1
    FOR i = NUMCOL1QUESTS + 2 TO NUMQUEST
      actrow(i) = actrow(i - 1) + Numresp(i - 1) + SKIPBLANK(i) + 1
    NEXT i
  END IF

  '     All Choices$()'s should be the same length, so max length is the
  '     length of any of them; therefore, use the first one.
  Xwidth = LEN(Choices$(1, 1))

  ShareTOPCOL = TOPCOL
  'Set up column pointer for graphics(page 1 of hx, pain at present
       should
  'not be 40 greater than pain at onset; would overlie abd graph on
       right.
  IF NUMQUEST > 1 AND GRAPHFLAG(2) = 2 THEN
    ShareTOPCOL2 = 46
  ELSE
    ShareTOPCOL2 = TOPCOL + 40
  END IF
  ShareNUMCOL1QUESTS = NUMCOL1QUESTS

  curquest = 1
  curresp = 1
 'main loop
 DO
  '     headings, frames, info
  'headings
  IF FrameFlag = 0 THEN
    CALL SetScreenMode(ScrnMode)
  END IF
  CALL LocateCenter(2, DATAHEADING$)
  headingPRINT (DATAHEADING$)
  Pageheading$ = "Page " + PAGEOF$ + " of " + OFPAGE$
  LOCATE 2, (78 - LEN(Pageheading$))
  headingPRINT (Pageheading$)
```

```
IF FrameFlag = 0 THEN
  CALL HPframe

  '  This portion of the subroutine prints the help message at the
  'bottom of the screen.
  SetColor (infocolor)
  LOCATE 23, 1: PRINT "Use the TAB key or arrow keys to move the
     cursor to the desired position.";
  LOCATE 24, 1: PRINT "Push RETURN to select the desired response or
     (P)revious page, (N)ext page,";
  LOCATE 25, 1, 0
  PRINT "e(X)it, or '?' for more information on that response.";
  SetColor (forecolor)
ELSE
  'frame already drawn is OK; need to blankd response areas.
  'clear left side
  LINE (5, 21)-(314, 171), 0, BF
  'clear right side
  LINE (317, 21)-(635, 171), 0, BF

END IF

' Set flag for frame drawing above.
FrameFlag = 1
' print questions, choices
actcol = TOPCOL
FOR questnum = 1 TO NUMQUEST
  IF questnum > NUMCOL1QUESTS THEN
    actcol = ShareTOPCOL2
  END IF
  ' Print appropriate question heading.
  SELECT CASE GRAPHFLAG(questnum)
    CASE 0            ' non-graphics questions
      LOCATE actrow(questnum) - 1, actcol
      questionPRINT (question$(questnum))
    CASE 1            ' pain at onset question
      LOCATE 4, 5: questionPRINT ("SITE OF PAIN")
      LOCATE 5, 7: questionPRINT ("At ONSET")
      ' LOCATE 4, 39: questionPRINT ("SITE")
      CALL DrawGraph(1)

    CASE 2            ' pain at present question
      LOCATE 4, 66: questionPRINT ("SITE OF PAIN")
      LOCATE 5, 67: questionPRINT ("At PRESENT")
      CALL DrawGraph(2)

    CASE 3            ' Tenderness question
      LOCATE 4, 6: questionPRINT (question$(1))
      CALL DrawGraph(3)
      CASE ELSE

    END SELECT

    FOR rowptr = 0 TO Numresp%(questnum) - 1
      LOCATE actrow(questnum) + rowptr, actcol + 4, 0
      responsePRINT (Choices$(questnum, rowptr + 1))
    NEXT rowptr

    'Update astericks
    FirstRow = actrow(questnum)
    Firstcol = actcol
    NonePtr = None(questnum)
    VariPtr = VariablePtr(questnum)
    GraphFlg = GRAPHFLAG(questnum)
    NumberofResp = Numresp(questnum)
    offset = 0
    CALL UpdateAsterisk(FirstRow, Firstcol, NonePtr, VariPtr, GraphFlg,
       NumberofResp, offset, VARIABLE())

  NEXT questnum

  'initialize cursor on page at first question, first response
  CALL PutCursor(curquest, curresp, Choices$())

  'now comes the key entry routine.
  CALL mainkeyroutine(exitstring$, curquest%, curresp%, NUMQUEST,
       Numresp(), NUMCOL1QUESTS, VariablePtr%(), VARIABLE(), MULTIP(),
       None%(), actrow(), GRAPHFLAG(), offset, STFLAG%, Choices$())
  exitchar$ = exitstring$

  'print help for H&P questions
  IF exitstring$ = "?" THEN
    CALL HelpDataEntry(HELPFILE$, curquest%)
    ' After displaying help file, will need to redraw fully the H&P
       page.
    FrameFlag = 0
  END IF

 LOOP UNTIL INSTR("NPX", exitchar$) <> 0

END SUB

REM $STATIC
SUB Disclaimer (VERSION$)
'       This subprogram displays the warning/disclaimer.
'       Do not need to save variables; Therefore, not STATIC.
64290 SCREEN 0, 1, 0, 0: CLS
```

```
      CALL SetFrameColor
      CALL frame(1, 1, 23, 78, 1)
      CALL SetNormalColor
64295 LOCATE 1, 16: PRINT "Abdominal Pain Diagnosis Program"; VERSION$
64300 LOCATE 3, 4: PRINT "    This computer-assisted diagnosis and
       medical support program can"
64301 LOCATE , 4: PRINT "reliably aid the corpsman in differentiating
       the five illness which"
64302 LOCATE , 4: PRINT "represent the most common serious causes of
       acute abdominal pain."
64304 LOCATE , 4: PRINT "In addition, a sixth category, non-specific
       abdominal pain, is"
64305 LOCATE , 4: PRINT "intended to represent those cases which are
       non-surgical, not life"
64306 LOCATE , 4: PRINT "threatening, and therefore, not reasons for
       evacuation."
64307 LOCATE , 4: PRINT
64308 LOCATE , 4: PRINT "    IMPORTANT - Not all diseases causing acute
       abdominal pain are"
64309 LOCATE , 4: PRINT "considered.  Input of symptom complexes
       associated with other diseases"
64310 LOCATE , 4: PRINT "will result in a diagnosis of one of the six
       categories most closely"
64311 LOCATE , 4: PRINT "resembling that disease."
64312 LOCATE , 4: PRINT
64313 LOCATE , 4: PRINT "    THE CORPSMAN'S JUDGEMENT MUST TAKE
       PRECEDENCE WHEN ANY DOUBT"
64314 LOCATE , 4: PRINT "EXISTS.  Remember that the computer does not
       have the capability"
64319 LOCATE , 4: PRINT "to think or make the subjective evaluations
       which are so important"
64320 LOCATE , 4: PRINT "in medical diagnosis."
      LOCATE , 4: PRINT
      LOCATE , 4: PRINT "    NOTE - This program routinely stores
       patient information which"
      LOCATE , 4: PRINT "requires protection under the Privacy Act of
       1974.  Users should"
      LOCATE , 4: PRINT "ensure that files created by this program are
       not subject to "
      LOCATE , 4: PRINT "unauthorized disclosure."


' This paragraph was in the above.  Had to pull it to insert the privacy
      act
'   stuff.
'
'  The computer will consider only its questions in making a
' diagnosis.  For example, severe hematemesis would surely play a part
' in the diagnosis and management of the patient, but the computer will
' be ignorant of the finding, since it will not ask any questions
' concerning hematemesis. Remember that the computer does not have the
' capability to think or make the subjective evaluations which are so
' important in medical diagnosis.

64330 CALL TextContinuePrompt
64980 CALL GetKey(a$)
END SUB

REM $DYNAMIC
SUB DisplayHPprint (HP%, SXloc%(), SXresp$(), VARIABLE%())
'       This routine prints the synopses for the hx or pe, depending on
'       the value of HP.  HP=0  HX;  HP=1  PE

  'uses backcolor

  'initialize
  IF HP = 0 THEN        'history listing
    IBEGIN = 1
    iend = 88
    fembegin = 160
    femend = 174
  ELSE
    IBEGIN = 89
    iend = 152
    fembegin = 153
    femend = 159
  END IF


  '     Print correct heading
  CALL DisplayHPTitle(HP)
  CALL scrollup(4, 2, 22, 79, 0, backcolor)
  row = 0
  col = 0

  'cycle through and check responses
  FOR i = IBEGIN TO iend
    IF VARIABLE(i) = 1 THEN
      CALL DisplayHProwcol(row, col, SXresp$(i))
    END IF
  NEXT i
  'female questions
  FOR i = fembegin TO femend
    IF VARIABLE(i) = 1 THEN
      CALL DisplayHProwcol(row, col, SXresp$(i))
    END IF
  NEXT i
```

```
END SUB

SUB DrawGraph (WhichOne%)
' This routine allows mainkeyroutine to be used by all
'programs(ABDX,CPDX,etc), by modifiying only the next line to call the
'proper subroutine.

  CALL ABDDrawGraph(WhichOne%)
END SUB

REM $STATIC
'------------------------------------------------------------------------
     ----
SUB InitiatePhrase (EasyHard, PHRASE$())
' Initializes the array PHRASE$() with easy or hard case narrative
     phrases.
'     EasyHard = 1  -> Easy case narrative phrases
'     EasyHard = 2  -> Hard case narrative phrases
  IF EasyHard = 1 THEN
      REM Data for easy case narrative data phrases.
    PHRASE$(11) = " the right upper quadrant"
    PHRASE$(12) = " the left upper quadrant"
    PHRASE$(13) = " the right lower quadrant"
    PHRASE$(14) = " the left lower quadrant"
    PHRASE$(15) = " the upper half"
    PHRASE$(16) = " the lower half"
    PHRASE$(17) = " the right half"
    PHRASE$(18) = " the left half"
    PHRASE$(19) = " the central part of his belly"
    PHRASE$(20) = " a rather vague area over the whole abdomen"
    PHRASE$(21) = " the right flank area"
    PHRASE$(22) = " the left flank area"
    PHRASE$(23) = "  "
    PHRASE$(24) = " and is now located in the right upper quadrant"
    PHRASE$(25) = " and is now located in the left upper quadrant"
    PHRASE$(26) = " and is now located in the right lower quadrant"
    PHRASE$(27) = " and is now located in the left lower quadrant"
    PHRASE$(28) = " and is now located in the upper half"
    PHRASE$(29) = " and is now located in the lower half"
    PHRASE$(30) = " and is now located in the right half"
    PHRASE$(31) = " and is now located in the left half"
    PHRASE$(32) = " and is now located centrally"
    PHRASE$(33) = " and is now located over the whole abdomen"
    PHRASE$(34) = " and is now located in the right flank area"
    PHRASE$(35) = " and is now located in the left flank area"
    PHRASE$(36) = "  "
    PHRASE$(37) = " and seems to come and go."
```

```
PHRASE$(38) = " and seems to be fairly constant."
PHRASE$(39) = " and seems to come and go but never really goes away
   entirely."
PHRASE$(40) = " not very bad right now"
PHRASE$(41) = " a really intense pain"
PHRASE$(42) = "  Movement seems to make the pain worse and"
PHRASE$(43) = "  Coughing seems to make the pain worse and"
PHRASE$(44) = "  Breathing seems to make the pain worse and"
PHRASE$(45) = "  Food seems to make the pain worse and"
PHRASE$(46) = "  Applying ice to the area of pain seems to make the
   pain worse and"
PHRASE$(47) = "  Nothing the patient does makes the pain worse and"
PHRASE$(48) = " and it seems to be getting better."
PHRASE$(49) = " and it seems to be about the same as when it first
   began."
PHRASE$(50) = " and it seems to be getting worse."
PHRASE$(51) = "  The pain began less than 12 hours ago and is"
PHRASE$(52) = "  The pain began 14 hours ago and is"
PHRASE$(53) = "  The pain began some time yesterday and is"
PHRASE$(54) = "  The pain began two days ago and is"
PHRASE$(55) = " lying still makes the pain a little better."
PHRASE$(56) = " vomiting relieves the pain a little."
PHRASE$(57) = " antacids seems to relieve the pain a little."
PHRASE$(58) = " eating bland foods seems to relieve the pain a
   little."
PHRASE$(59) = " applying heat to the area of pain seems to help a
   little."
PHRASE$(60) = " nothing he does makes the pain any better."
PHRASE$(61) = "  He has felt sick to his stomach all day"
PHRASE$(62) = "  He hasn't been nauseated today"
PHRASE$(63) = " and has vomited twice."
PHRASE$(64) = " and has not been vomiting."
PHRASE$(65) = "  His bowels have been relatively normal"
PHRASE$(66) = "  He has been experiencing some constipation
   recently"
PHRASE$(67) = "  He has had some diarrhea recently"
PHRASE$(68) = "  He has noticed some blood in his bowel movements
   recently"
PHRASE$(69) = "  He has noticed some stringy white material in his
   stool recently"
PHRASE$(70) = "  He states that he has not felt like eating today
   because of his discomfort."
PHRASE$(71) = "  He states that he still feels like eating in spite
   of the discomfort."
PHRASE$(72) = "  One of his friends has told him that his eyes have
   been turning a yellow color recently."
PHRASE$(73) = "  He hasn't noticed any change in the color of his
   skin or eyes recently."
PHRASE$(74) = " and his urinary habits have been normal."
```

```
PHRASE$(75) = " and he complains of having to urinate more often
   than usual."
PHRASE$(76) = " and he has had some discomfort when he urinates."
PHRASE$(77) = " and his urine has been darker than usual."
PHRASE$(78) = " and he has noticed a red tint to his urine
   recently."
PHRASE$(79) = "  He has been bothered by minor G-I upset from time
   to time"
PHRASE$(80) = "  There is no history of previous G-I upset"
PHRASE$(81) = " and he relates an episode of pain very similar to
   this a couple of months ago."
PHRASE$(82) = " and he cannot recall ever having a pain like this
   before."
PHRASE$(83) = "  An appendectomy was performed when he was very
   young and he doesn't remember any other hospitalization."
PHRASE$(84) = "  Repair of a hernia has been his only
   hospitalization."
PHRASE$(85) = "  The patient relates a history of many episodes of
   abdominal illness"
PHRASE$(86) = "  The patient denies a history of G-I illnesses"
PHRASE$(87) = " and is now taking aspirin and Maalox for his pain."
PHRASE$(88) = " and is not taking any medication for his pain."
PHRASE$(89) = " 98.4,"
PHRASE$(90) = " 100.2,"
PHRASE$(91) = " 101.1,"
PHRASE$(92) = " 102.6,"
PHRASE$(93) = " pulse 74,"
PHRASE$(94) = " pulse 86,"
PHRASE$(95) = " pulse 110,"
PHRASE$(96) = " blood pressure 86/"
PHRASE$(97) = " blood pressure 122/"
PHRASE$(98) = " blood pressure 144/"
PHRASE$(99) = "62,"
PHRASE$(100) = "80,"
PHRASE$(101) = "94,"
PHRASE$(102) = "    Your examination reveals a patient who is well
   developed and well nourished, in no apparent distress, and who"
PHRASE$(103) = "    Your examination reveals a patient who is in
   obvious distress from his pain and who"
PHRASE$(104) = "    Your examination reveals a patient who is
   concerned about his condition but doesn't appear to be in
   distress and who"
PHRASE$(105) = "se color appears normal."
PHRASE$(106) = " appears pale."
PHRASE$(107) = " appears flushed."
PHRASE$(108) = " appears to have a yellow color to his skin and
   eyes."
PHRASE$(109) = "se skin has a slight blue color."
PHRASE$(110) = " and his white blood cell count is 6,800."
```

```
PHRASE$(111) = " and his white blood cell count is 8,800."
PHRASE$(112) = " and his white blood cell count is 10,800."
PHRASE$(113) = " and his white blood cell count is 13,000."
PHRASE$(114) = " and his white blood cell count is 16,200."
PHRASE$(115) = "  Inspection of the abdomen reveals no
   abnormalities."
PHRASE$(116) = "  Inspection of the abdomen reveals visible
   peristaltic waves."
PHRASE$(117) = "  The patient experienced difficulty in raising his
   belly to touch your hand when requested to during the abdominal
   inspection."
PHRASE$(118) = "  A surgical scar is present in the midline"
PHRASE$(119) = "  There are no surgical scars on the abdomen"
PHRASE$(120) = "  The patient reflexively tenses his abdominal
   muscles during palpation and"
PHRASE$(121) = "  The patient doesn't appear to be guarding and"
PHRASE$(122) = " there is some residual muscle spasm throughout the
   examination."
PHRASE$(123) = " the abdomen is soft during palpation."
PHRASE$(124) = "  Bowel sounds are normal."
PHRASE$(125) = "  No bowel sounds could be appreciated."
PHRASE$(126) = "  Bowel sounds are markedly increased."
PHRASE$(127) = " and there is a generalized swelling of the entire
   abdomen."
PHRASE$(128) = " and there is no apparent distention."
PHRASE$(129) = "  There is a mass appreciated centrally and"
PHRASE$(130) = "  There are no masses and"
PHRASE$(131) = "  Tenderness is noted in the right upper quadrant"
PHRASE$(132) = "  Tenderness is noted in the left upper quadrant"
PHRASE$(133) = "  Tenderness is noted in the right lower quadrant"
PHRASE$(134) = "  Tenderness is noted in the left lower quadrant"
PHRASE$(135) = "  Tenderness is noted in the upper half of the
   abdomen"
PHRASE$(136) = "  Tenderness is noted in the lower half of the
   abdomen"
PHRASE$(137) = "  Tenderness is noted in the right half of the
   abdomen"
PHRASE$(138) = "  Tenderness is noted in the left half of the
   abdomen"
PHRASE$(139) = "  Tenderness is noted in the middle of the abdomen"
PHRASE$(140) = "  Tenderness is noted over the entire abdomen"
PHRASE$(141) = "  Tenderness is noted in the right flank area"
PHRASE$(142) = "  Tenderness is noted in the left flank area"
PHRASE$(143) = "  Tenderness is not noted"
PHRASE$(144) = " Murphy's sign is present."
PHRASE$(145) = " Murphy's sign is not present."
PHRASE$(146) = " and rebound tenderness is appreciated."
PHRASE$(147) = " and rebound tenderness is not appreciated."
PHRASE$(148) = "  The rectal examination is non-revealing."
```

```
  PHRASE$(149) = "  A mass is appreciated anteriorly on rectal
     examination."
  PHRASE$(150) = "  Tenderness is elicited on the left side during
     rectal examination."
  PHRASE$(151) = "  Tenderness is elicited on the right side during
     rectal examination."
  PHRASE$(152) = "  Rectal examination reveals generalized
     tenderness."
ELSE
    REM Harder case narrative data phrases.
  PHRASE$(11) = " the right side of his belly below his ribs"
  PHRASE$(12) = " the left side of his belly below his ribs"
  PHRASE$(13) = " the lower part of his belly and mostly on the right
     side"
  PHRASE$(14) = " the lower part of his belly and mostly on the left
     side"
  PHRASE$(15) = " the upper part of his belly"
  PHRASE$(16) = " the lower part of his belly"
  PHRASE$(17) = " the middle area of his belly but mostly on the
     right"
  PHRASE$(18) = " the middle area of his belly but mostly on the left"
  PHRASE$(19) = " the area right around his navel"
  PHRASE$(20) = " the middle of his belly but spread out all over"
  PHRASE$(21) = " his right side mostly around the back"
  PHRASE$(22) = " his left side but mostly around the back"
  PHRASE$(23) = " "
  PHRASE$(24) = " and is now located on the right side mostly just
     below the ribs"
  PHRASE$(25) = " and is now located on the left side mostly just
     below the ribs"
  PHRASE$(26) = " and is now located on the right side below his
     navel"
  PHRASE$(27) = " and is now located on the left side below his navel"
  PHRASE$(28) = " and is now located mostly in a vague area above his
     navel"
  PHRASE$(29) = " and is now located mostly in a vague area below his
     navel"
  PHRASE$(30) = " and is now located mostly on the right side of his
     belly"
  PHRASE$(31) = " and is now located mostly on the left side of his
     belly"
  PHRASE$(32) = " and is now located mostly in the middle (he puts his
     finger on his navel when asked to point to the pain)"
  PHRASE$(33) = " and it now hurts everywhere"
  PHRASE$(34) = " and it now hurts toward the back and over his right
     hip"
  PHRASE$(35) = " and it now hurts toward the back and over his left
     hip"
  PHRASE$(36) = " "
```

```
PHRASE$(37) = " but he doesn't have it right now."
PHRASE$(38) = " and it just won't seem to go away."
PHRASE$(39) = " and although it seems a little better right now,
   it's still there."
PHRASE$(40) = "  It isn't giving him a great deal of discomfort
   right now"
PHRASE$(41) = "  It has him doubled-over most of the time"
PHRASE$(42) = "  The pain is worse when he tries to climb ladders
   and"
PHRASE$(43) = "  The pain is worse when he coughs and"
PHRASE$(44) = "  The pain seems to be accentuated when he exhales
   deeply and"
PHRASE$(45) = "  He says that the pain seems to get worse when he
   eats and"
PHRASE$(46) = "  Pressing on the region of the pain seems to make it
   worse and"
PHRASE$(47) = "  He cannot think of anything that makes the pain
   worse and"
PHRASE$(48) = " and it seems to be easing up a little now."
PHRASE$(49) = " and it really hasn't changed very much."
PHRASE$(50) = " and he hopes you can ease the pain a little since
   his discomfort is increasing."
PHRASE$(51) = "  When he began his watch 8 hours ago, he was free of
   pain."
PHRASE$(52) = "  He has only noticed the pain during the last day."
PHRASE$(53) = "  He was in his usual state of good health until
   yesterday."
PHRASE$(54) = "  Because of the pain, he has not felt like standing
   watch for the last three days."
PHRASE$(55) = " he prefers to stay in his rack since the pain is
   less then."
PHRASE$(56) = " vomiting relieves the pain a little."
PHRASE$(57) = " the one drink of Maalox you just gave him seems to
   make the pain a little better."
PHRASE$(58) = " milk seems to relieve the pain a little."
PHRASE$(59) = " applying heat to the area of pain seems to help."
PHRASE$(60) = " he hasn't been able to find anything which will
   relieve the pain."
PHRASE$(61) = "  He has felt sick to his stomach all day and"
PHRASE$(62) = "  He hasn't really felt sick to his stomach and"
PHRASE$(63) = " has vomited a couple of times."
PHRASE$(64) = " has not vomited."
PHRASE$(65) = "  He hasn't noticed any change in his bowels"
PHRASE$(66) = "  He has been unable to have a bowel movement in the
   past three days"
PHRASE$(67) = "  He has had loose stools for the last four days"
PHRASE$(68) = "  He has noticed a red color in the water after a
   bowel movement"
```

```
PHRASE$(69) = "  He has seen some white, stringy material in his
   stool lately"
PHRASE$(70) = "  He states that food is distinctly unappealing to
   him."
PHRASE$(71) = "  He states that he was able to eat a regular meal
   earlier today."
PHRASE$(72) = "  While shaving yesterday, he noticed a yellow tint
   to his eyes."
PHRASE$(73) = "  He hasn't appreciated any change in the color of
   his eyes."
PHRASE$(74) = " and has not had a problem with urination."
PHRASE$(75) = " and he complains of having to urinate every 45
   minutes."
PHRASE$(76) = " and he complains of burning when he urinates."
PHRASE$(77) = " and his urine has darkened so much that it looks
   like tea."
PHRASE$(78) = " and his urine has been slightly red-colored
   recently."
PHRASE$(79) = "  He has frequently been bothered by stomach upset
   after meals"
PHRASE$(80) = "  He hasn't been bothered by stomach upset"
PHRASE$(81) = " and he has occasionally had episodes of pain like
   this in the last two months."
PHRASE$(82) = " and he has never had this same pain before."
PHRASE$(83) = "  Correction of pyloric stenosis as a child has been
   his only hospitalization."
PHRASE$(84) = "  His only hospitalizations were for pneumonia and a
   tonsillectomy."
PHRASE$(85) = "  The patient states that he has a long history of
   undiagnosed abdominal pains"
PHRASE$(86) = "  The patient hasn't had any major illnesses since
   childhood"
PHRASE$(87) = " and is now taking probanthine, Maalox and Gaviscon."
PHRASE$(88) = " and denies the need for medication at this time."
PHRASE$(89) = " 98.2,"
PHRASE$(90) = " 100.2,"
PHRASE$(91) = " 101.1,"
PHRASE$(92) = " 102.6,"
PHRASE$(93) = " pulse 74,"
PHRASE$(94) = " pulse 86,"
PHRASE$(95) = " pulse 110,"
PHRASE$(96) = " blood pressure 86/"
PHRASE$(97) = " blood pressure 122/"
PHRASE$(98) = " blood pressure 144/"
PHRASE$(99) = "62,"
PHRASE$(100) = "80,"
PHRASE$(101) = "94,"
```

```
PHRASE$(102) = "    Your examination reveals a well-developed and
   well-nourished patient who is calm and is in no apparent
   distress"
PHRASE$(103) = "    Your examination reveals a patient who is
   obviously in severe pain"
PHRASE$(104) = "    Your examination reveals a patient who is
   concerned about his condition but is not in severe pain"
PHRASE$(105) = " and whose color is normal."
PHRASE$(106) = " and who appears a little pale."
PHRASE$(107) = " and the color of his face seems to be redder than
   usual."
PHRASE$(108) = " and his skin and sclera are obviously icteric."
PHRASE$(109) = " and whose skin looks a little blue."
PHRASE$(110) = " and his white blood cell count is 6,800."
PHRASE$(111) = " and his white blood cell count is 8,800."
PHRASE$(112) = " and his white blood cell count is 10,800."
PHRASE$(113) = " and his white blood cell count is 13,000."
PHRASE$(114) = " and his white blood cell count is 16,200."
PHRASE$(115) = "  Inspection of the abdomen reveals normal movement
   without obvious peristaltic waves."
PHRASE$(116) = "  Inspection of the abdomen reveals visible wave-
   like movements on the  surface."
PHRASE$(117) = "  Inspection of the abdomen is normal except that
   the patient has some trouble raising his belly to your hand."
PHRASE$(118) = " and a well-healed surgical scar is noted on the
   anterior abdominal wall."
PHRASE$(119) = " and there are no marks or scars on the abdomen."
PHRASE$(120) = "  The patient reflexively tenses his abdominal wall
   muscles during palpation"
PHRASE$(121) = "  The patient is easily able to relax during the
   examination"
PHRASE$(122) = " and there is some residual muscle spasm throughout
   the examination."
PHRASE$(123) = " and the abdomen is soft during palpation."
PHRASE$(124) = "  On auscultation of the abdomen, two gurgles were
   noted per minute"
PHRASE$(125) = "  No bowel sounds could be appreciated"
PHRASE$(126) = "  On auscultation of the abdomen, constant gurgles
   were heard"
PHRASE$(127) = "  The abdomen appears slightly bloated."
PHRASE$(128) = "  The abdomen is flat."
PHRASE$(129) = "  There appears to be a localized swelling near the
   umbilicus"
PHRASE$(130) = "  There don't appear to be any localized swellings"
PHRASE$(131) = " on the right side of his belly below his ribs"
PHRASE$(132) = " on the left side of his belly below his ribs"
PHRASE$(133) = " on the lower part of his belly, mostly on the right
   side"
```

```
    PHRASE$(134) = " on the lower part of his belly, mostly on the left
       side"
    PHRASE$(135) = " on the upper part of his belly"
    PHRASE$(136) = " on the lower part of his belly"
    PHRASE$(137) = " in the middle part of his belly but mostly on the
       right"
    PHRASE$(138) = " in the middle part of his belly but mostly on the
       left"
    PHRASE$(139) = " in the area right around his navel"
    PHRASE$(140) = " in the middle of his belly but spread out all over"
    PHRASE$(141) = " on his right side but mostly around the back"
    PHRASE$(142) = " on his left side but mostly around the back"
    PHRASE$(143) = " "
    PHRASE$(144) = " and you notice that the patient experiences pain
       when you are palpating below the right ribs.  Palpation of the
       abdomen reveals an area of tenderness"
    PHRASE$(145) = " and no pain or reflex inhibition of inspiration is
       noted when palpating below the right costal margin.  Palpation of
       the abdomen reveals an area of tenderness"
    PHRASE$(146) = " with a marked increase in the pain when quickly
       withdrawing your hand to the level of the skin."
    PHRASE$(147) = " with no increase in the pain when quickly
       withdrawing your hand to the level of the skin."
    PHRASE$(148) = "  The rectal examination fails to reveal masses,
       tenderness, or occult blood."
    PHRASE$(149) = "  The rectal examination reveals a nodule on the
       rectal wall."
    PHRASE$(150) = "  The rectal examination reveals slight tenderness
       on the left side."
    PHRASE$(151) = "  The rectal examination reveals slight tenderness
       on the right side."
    PHRASE$(152) = "  The rectal examination reveals tenderness on both
       sides, but more on the right side."
  END IF
END SUB

'------------------------------------------------------------------------
     ---
SUB InitiateTHELOOP (THELOOP%())
'   Routine initiates the THELOOP%() array with beginning and ending
' response numbers.
29800 THELOOP%(0, 1) = 11: THELOOP%(0, 2) = 23
      THELOOP%(1, 1) = 24: THELOOP%(1, 2) = 39
      THELOOP%(2, 1) = 51: THELOOP%(2, 2) = 54
      THELOOP%(3, 1) = 40: THELOOP%(3, 2) = 41
      THELOOP%(4, 1) = 48: THELOOP%(4, 2) = 50
      THELOOP%(5, 1) = 42: THELOOP%(5, 2) = 47
      THELOOP%(6, 1) = 55: THELOOP%(6, 2) = 64
      THELOOP%(7, 1) = 70: THELOOP%(7, 2) = 73
      THELOOP%(8, 1) = 65: THELOOP%(8, 2) = 69
      THELOOP%(9, 1) = 74: THELOOP%(9, 2) = 88
      THELOOP%(10, 1) = 89: THELOOP%(10, 2) = 101
      THELOOP%(11, 1) = 110: THELOOP%(11, 2) = 114
      THELOOP%(12, 1) = 102: THELOOP%(12, 2) = 109
      THELOOP%(13, 1) = 115: THELOOP%(13, 2) = 117
      THELOOP%(14, 1) = 124: THELOOP%(14, 2) = 126
      THELOOP%(15, 1) = 118: THELOOP%(15, 2) = 119
      THELOOP%(16, 1) = 127: THELOOP%(16, 2) = 128
      THELOOP%(17, 1) = 120: THELOOP%(17, 2) = 123
      THELOOP%(18, 1) = 129: THELOOP%(18, 2) = 130
      THELOOP%(19, 1) = 144: THELOOP%(19, 2) = 145
      THELOOP%(20, 1) = 131: THELOOP%(20, 2) = 143
      THELOOP%(21, 1) = 146: THELOOP%(21, 2) = 152

END SUB

' ====================================================================
SUB narrative (NR, CASENUM, THECASE%())

        DIM THELOOP%(21, 2), PHRASE$(152)

'   Initialize PHRASE$() array.
'      NR = 1  -> Easy case narrative phrases
'      NR = 2  -> Hard case narrative phrases
      CALL InitiatePhrase(NR, PHRASE$())

31330 LOCATE 15, 10, 1: PRINT "Enter the desired case number (1-50): ";
31340 LINE INPUT a$
31342 LOCATE , , 0
31345 IF a$ = "" THEN
        ERASE THELOOP%, PHRASE$
        EXIT SUB
      END IF
31350 CASENUM = VAL(LEFT$(a$, 3))

'            Check Validity of the Number Entered.  If Invalid, Beep and
       Get Another Case Number.
31360 IF CASENUM < 1 OR CASENUM > 50 THEN
        SOUND 100, 4
        LOCATE 15, 47: PRINT SPACE$(10)
        GOTO 31330
      END IF

'load desired case
       CALL LoadTrainingCase(CASENUM, THECASE%())
'31370 GOSUB 31700
```

```
'create array with beginning and ending response numbers for each
       question.
      CALL InitiateTHELOOP(THELOOP%())
'31430 RESTORE 29800
'31440 FOR I-0 TO 21:FOR J-1 TO 2:READ THELOOP%(I,J):NEXT J:NEXT I

31445 IBEGIN - 0: NAM$ - "History"
      CALL TrainingHPHeading(IBEGIN, CASENUM, NR, NAM$)

31450 FOR i - 3 TO 10
        IF THECASE%(i) <> 0 THEN GOTO 31460
      NEXT i
31460 IF i - 4 THEN AGE - 19 ELSE AGE - 10 * (i - 3) + INT(RND * (10))
31470 a$ - "    This patient is a" + STR$(AGE) + " year old sailor who
       presents with pain in his abdomen which began in"
31480 GOSUB 20050
31490 K1 - 0: K2 - 9: GOSUB 20200: a$ - "": GOSUB 20090: LOCATE 1, 1, 0
31500 GOTO 31570
31540 IBEGIN - 1: NAM$ - "Physical": BENTHER - 1
      CALL TrainingHPHeading(IBEGIN, CASENUM, NR, NAM$)
31545 a$ - "    On physical examination of your patient, he is noted to
       have a temperature of": GOSUB 20050
31550 K1 - 10: K2 - 11: GOSUB 20200: a$ - "": GOSUB 20090
31560 K1 - 12: K2 - 21: GOSUB 20200: a$ - "": GOSUB 20090: LOCATE 1, 1,
       0
31570 CALL GetKey(a$)

      IF a$ - "X" THEN
        ERASE THELOOP%, PHRASE$
        EXIT SUB
      END IF

31580 IF a$ - "P" THEN
        IF IBEGIN - 1 THEN
          IBEGIN - 0
          SCREEN 0, 1, 0
          GOTO 31570
        ELSE
          ERASE THELOOP%, PHRASE$
          EXIT SUB
        END IF
      END IF
'                    display directions
31590 IF a$ - "?" THEN
        CALL NarrativeHelp((IBEGIN), (IBEGIN))
        GOTO 31570
      END IF
31600 IF a$ <> "N" THEN SOUND 100, 4: GOTO 31570
31604 IF IBEGIN - 1 THEN EXIT SUB
```

```
31606 IF BENTHER = 0 THEN
          GOTO 31540
        ELSE
          IBEGIN = 1
          SCREEN 0, 1, 1
          GOTO 31570
        END IF

20050 B$ = "": COUNTER = 1: ACOUNT = 1
20090 IF a$ = "" THEN
        LOCATE , 3
        PRINT B$
        B$ = "": COUNTER = 1: ACOUNT = 1
        RETURN
      END IF
20100 B$ = B$ + MID$(a$, ACOUNT, 1)
20110 ACOUNT = ACOUNT + 1
      COUNTER = COUNTER + 1
      IF ACOUNT > LEN(a$) AND COUNTER >= 72 THEN 20150
20120 IF ACOUNT > LEN(a$) THEN ACOUNT = 1: RETURN
20130 IF COUNTER > 72 THEN 20150
20140 GOTO 20100
20150 COUNTER = COUNTER - 1: ACOUNT = ACOUNT - 1
20160 IF MID$(B$, COUNTER, 1) = " " THEN
        LOCATE , 3
        PRINT MID$(B$, 1, COUNTER - 1)
        COUNTER = 1
        ACOUNT = ACOUNT + 1
        B$ = ""
        GOTO 20100
      END IF
20170 COUNTER = COUNTER - 1: ACOUNT = ACOUNT - 1
20180 GOTO 20160
20200 FOR i = K1 TO K2
20205   IF i = 0 OR i = 20 THEN
          GOSUB 20300
          GOTO 20235
        END IF
20210   FOR J = THELOOP%(i, 1) TO THELOOP%(i, 2)
20220     IF THECASE%(J) <> 0 THEN
            a$ = PHRASE$(J)
            GOSUB 20090
          END IF
20230   NEXT J
20235 NEXT i
20240 RETURN

20300 MULFLAG = 0
20310 FOR J = THELOOP%(i, 1) TO THELOOP%(i, 2)
```

```
20320 IF THECASE%(J) = 0 THEN 20350
20330 a$ = PHRASE$(J): IF MULFLAG <> 0 THEN a$ = " and" + a$
20340 GOSUB 20090: MULFLAG = 1
20350 NEXT J: RETURN

END SUB

REM $DYNAMIC
SUB PaintGraph (VAR%, WhichOne%)
' This routine allows mainkeyroutine to be used by all
'programs(ABDX,CPDX,etc), by modifiying only the next line to call the
'proper subroutine.

  CALL ABDPaintGraph(VAR%, WhichOne%)

END SUB

REM $STATIC
SUB TrainingHPHeading (IBEGIN, CASENUM, NR, NAM$)
      'Draws heading for Training case narrative H&P

20000 SCREEN 0, 1, IBEGIN, IBEGIN
      CLS
20010 LOCATE 1, 2
      dash$ = "Training Case number:" + STR$(CASENUM)
      dash$ = dash$ + "  Difficulty:" + STR$(NR)
      headingPRINT (dash$)
20020 LOCATE 2, 30
      PRINT NAM$;
      CALL SetFrameColor
      framtyp = frametype
      CALL frame(3, 1, 20, 78, framtyp)
      CALL SetNormalColor
20025 LOCATE 25, 8
      SetColor (infocolor%)
      infostring$ = "Enter (P)revious, (N)ext, e(X)it, or (?) for help."
      CALL CenterString(infostring$)
      SetColor (forecolor%)
20030 LOCATE 4, 1, 0

END SUB

SUB UpdateAge (agevar%, VARIABLE%())
'This routine looks at agevar, and checks it for legality.  Then, the
' VARIABLE() array corresponding to age is zeroed and it is then updated
' at the proper element.
'NOTE:  agevar initially is the age.  On return, it is the pointer to
' the age in VARIABLE().
```

```
    IF agevar > AGEMAXIMUM THEN agevar = AGEMAXIMUM
    agevar = INT(agevar / 10) + 3
    FOR i = 3 TO 10
      VARIABLE(i) = 0
    NEXT i
    VARIABLE(agevar) = 1

END SUB
```

ABDXSUB1.BAS

```
DECLARE SUB SetColor (thecolor%)
DECLARE FUNCTION VideoMode% ()
DECLARE SUB CenterPrint (row%, TheString$)
DECLARE SUB TextPause ()
DECLARE FUNCTION Centered% (s$)
DECLARE SUB inversegraph (row%; col%, StrWidth%)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE FUNCTION FileIsPresent& (nam$)
DECLARE SUB GetUCResponse (ch$, filter$)
DECLARE SUB TextContinuePrompt ()
DECLARE SUB UpdatePtrMinus (ptr%)
DECLARE SUB UpdatePtrPlus (ptr%)
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB GetKey (a$)
DECLARE SUB ContinuePrompt ()
DEFINT A-Z
' $INCLUDE: 'include.bas'
'━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

'━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

'       ABDXSUB1.BAS
'
'       Contains: ContinuePrompt          originally @ line   40216
'
'                 SSNAgeDate(STFLAG,TRAINING,SIMULATE,SSN$,
'                              AGE$,AGE,STARTDATE$,STARTIME$,
'                              PT%(0).VARIABLE%(0))              63000
'
'
'                 GetBoatStuff(BOAT1$,BOAT2$,HMNAM$,HMSSN$)   64000
'
'
'                 Disclaimer                                     64290
'
'━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

FUNCTION FileIsPresent& (nam$)
'       This routine returns a 0 if file is not present and the length
'        of the file if it is present.
'
      filenum = FREEFILE
      OPEN "R", #1, nam$, filenum
      N& = LOF(1)
      CLOSE filenum
      FileIsPresent& = N&
END FUNCTION

'=================================================================
SUB frame (ulr, ulc, numlines, length, frametyp)
'       Draws frame about coordinates given which form the corner of
'     the frame.  Numlines and length do not include frame itself.
'       Also, can have several types of frames.
'     1 = single frame, 2 = double frame; 3-5 = block frames.

' Will use line statement if in screen 2.  It seems that some of the
' straight CGA cards to not have the high ASCII graphics characters
       while
' in screen 2. (At least that is the case for the IBM CGA board.)

IF VideoMode = 6 THEN   'screen 2
  ulxcoor = (ulc - 1) * 8
  ulycoor = (ulr - 1) * 8
  lrxcoor = (ulc + length) * 8
  lrycoor = (ulr + numlines) * 8

  SELECT CASE frametyp
    CASE 1          'single line
      LINE (ulxcoor + 4, ulycoor + 4)-(lrxcoor + 4, lrycoor + 4), 1, B
      LINE (ulxcoor + 5, ulycoor + 4)-(lrxcoor + 3, lrycoor + 4), 1, B

    CASE ELSE   'throw everything else as 2 lines.
      LINE (ulxcoor + 1, ulycoor + 2)-(lrxcoor + 5, lrycoor + 4), 1, B
      LINE (ulxcoor + 2, ulycoor + 2)-(lrxcoor + 4, lrycoor + 4), 1, B
      LINE (ulxcoor + 4, ulycoor + 4)-(lrxcoor + 2, lrycoor + 2), 1, B
      LINE (ulxcoor + 5, ulycoor + 4)-(lrxcoor + 1, lrycoor + 2), 1, B

    END SELECT
ELSE

  SELECT CASE frametyp
    CASE 1
      tlc$ = CHR$(218)
      trc$ = CHR$(191)
      llc$ = CHR$(192)
      lrc$ = CHR$(217)
      horiz = 196
      vert$ = CHR$(179)
    CASE 2
      tlc$ = CHR$(201)
      trc$ = CHR$(187)
      llc$ = CHR$(200)
      lrc$ = CHR$(188)
      horiz = 205
      vert$ = CHR$(186)
    CASE 3
      tlc$ = CHR$(176)
      trc$ = CHR$(176)
      llc$ = CHR$(176)
      lrc$ = CHR$(176)
      horiz = 176
      vert$ = CHR$(176)
    CASE 4
      tlc$ = CHR$(177)
      trc$ = CHR$(177)
      llc$ = CHR$(177)
      lrc$ = CHR$(177)
      horiz = 177
      vert$ = CHR$(177)
    CASE 5
      tlc$ = CHR$(178)
      trc$ = CHR$(178)
      llc$ = CHR$(178)
      lrc$ = CHR$(178)
      horiz = 178
      vert$ = CHR$(178)
    CASE ELSE
      tlc$ = CHR$(219)
      trc$ = CHR$(219)
      llc$ = CHR$(219)
      lrc$ = CHR$(219)
      horiz = 219
      vert$ = CHR$(219)
  END SELECT

  'reality checks for coordinates -- add later.

  horiz$ = STRING$(length, horiz)
  topstring$ = tlc$ + horiz$ + trc$
  bottomstring$ = llc$ + horiz$ + lrc$
  LOCATE ulr, ulc
  PRINT topstring$;
  lrc = ulc + length + 1
  lrr = ulr + numlines + 1
  FOR r = ulr + 1 TO lrr - 1
    LOCATE r, ulc: PRINT vert$;
    LOCATE r, lrc: PRINT vert$;
  NEXT r
  LOCATE lrr, ulc
  PRINT bottomstring$;
END IF

END SUB
```

```
SUB GetBoatStuff (BOAT1$, BOAT2$, HMNAM$, HMSSN$, VersionNumber$)
'       This routine checks for SHIP.DAT and gets info from it.  If
'       it is not present, then it asks questions and creates it.
'       It also checks for SUBPIC.DAT and prints it if found.
64000 REM  TITLE PAGE ROUTINE
64010
      REM This routine checks for presence of the SHIP.DAT file.
       SHIP.DAT contains corpsman's name and SSN and Boat Name.
64020 NAMEOFFILE$ = "SHIP.DAT"
'            SHIP.DAT file does not exist.  Routine to create file.
      IF FileIsPresent&(NAMEOFFILE$) = 0 THEN
SCREEN 0, 1, 0, 0
   CLS
64050   PRINT "The SHIP.DAT file is not present on the program disk."
PRINT " This file contains your name,and boat.  You will now"
PRINT " create it."
 PRINT
'            Enter Boat Name and Hull Number.
PRINT "Enter your boat name. (ex: USS MISSISSIPPI (do not include hull
       #) )"
PRINT " BOAT NAME > ";
LINE INPUT BOAT1$
 PRINT
PRINT "Enter your boat's hull number. (ex: SSBN 999 BLUE) "
PRINT " HULL NUMBER > ";
LINE INPUT BOAT2$
'            Enter Corpsman's Name.
 PRINT
PRINT "Enter your name as signed on an SF-600."
PRINT " NAME > ";
LINE INPUT HMNAM$
 PRINT
PRINT "Enter your SSN.  If you do not desire your SSN to be printed"
PRINT " beneath your name on an SF-600 entry, just press the ENTER"
PRINT " key by itself."
 PRINT
PRINT " SSN > ";
LINE INPUT HMSSN$


'            Display Boat Name, Hull Number, Corpsman's Name and SSN.
   CLS
LOCATE 10, 10: PRINT "Boat Name    -  "; BOAT1$
LOCATE 11, 10: PRINT "Hull Number -  "; BOAT2$
LOCATE 12, 10: PRINT "Your Name    -  "; HMNAM$
LOCATE 13, 10: PRINT "Your SSN     -  "; HMSSN$
```

```
'            Ask user if information is correct.  If correct, then store
       information in file SHIP.DAT.  If incorrect, get new information.
LOCATE 17, 10: PRINT "Is this correct? [Y/N] ";
64060   CALL GetUCResponse(a$, "YN")
IF a$ = "N" THEN CLS : GOTO 64050
OPEN "SHIP.DAT" FOR OUTPUT AS #1
PRINT #1, BOAT1$
PRINT #1, BOAT2$
PRINT #1, HMNAM$
PRINT #1, HMSSN$
CLOSE #1
   CLS
      ELSE
64200   OPEN "I", #1, "SHIP.DAT"
64210   LINE INPUT #1, BOAT1$
64220   LINE INPUT #1, BOAT2$
64230   LINE INPUT #1, HMNAM$
64235   LINE INPUT #1, HMSSN$
64240   CLOSE #1
      END IF
64250 NAMEOFFILE$ = "SUBPIC.BIN"
      REM This routine checks for presence of the SUBPIC.DAT file.
      IF FileIsPresent&(NAMEOFFILE$) = 0 THEN
'            File does not exist.  Display the title of the program and
       Boat Name, Hull Number, Corpsman's Name and SSN.  Display
       instructions at the bottom of the screen.
SCREEN 0, 1, 0, 0
   CLS
      CALL SetFrameColor
      CALL frame(1, 1, 23, 78, 1)
      CALL SetNormalColor
'64295 LOCATE 1, 16: PRINT "Chest Pain Diagnosis Program"; VERSION$

CALL SetColor(MAINHEADINGCOLOR%)
        ConfigureTitle$ = PROGRAMTYPE$ + " Diagnostic Program"
CALL CenterPrint(1, ConfigureTitle$)
ConfigureVersion$ = "Version " + VersionNumber$
CALL CenterPrint(2, ConfigureVersion$)
CALL SetColor(forecolor)
CALL CenterPrint(10, "Configured For:")
CALL SetColor(hiwhite)
CALL CenterPrint(12, BOAT1$)
CALL CenterPrint(13, BOAT2$)

CALL SetColor(questioncolor)
CALL CenterPrint(20, "Naval Submarine Medical Research Laboratory")
CALL CenterPrint(21, "Box 900, Subase New London")
CALL CenterPrint(22, "Groton, CT   06349-5900")
CALL CenterPrint(23, "AV 241-3668, COMM (203) 449-3668")
CALL SetColor(forecolor)

CALL TextContinuePrompt
'            SUBPIC.BIN exists (picture of submarine).  Display picture,
       Boat Name and Hull Number.
      ELSE
SCREEN 2, 0, 0, 0: OUT &H3D9, 7
DEF SEG = &HB800
BLOAD NAMEOFFILE$, 0
DEF SEG
64255   LOCATE 15, 30: PRINT BOAT1$;
64260   LOCATE 16, 30: PRINT BOAT2$;
      END IF
END SUB

SUB GetGraphMode (GRAPHICS$, MONITOR$)
      CALL returnadapter(adapt)
      SELECT CASE adapt
CASE 1
  GRAPHICS$ = "C"
CASE 2
  GRAPHICS$ = "E"
CASE ELSE
   CLS
  LOCATE 10, 5
  PRINT "According to my sensors, your computer does not support";
  LOCATE 11, 5
  PRINT "CGA, EGA, or VGA graphics.  Therefore this program cannot";
  LOCATE 12, 5
  PRINT "continue.";
   END
      END SELECT

      NAMEOFFILE$ = "ABDGRAPH.DAT"
      IF FileIsPresent&(NAMEOFFILE$) = 0 THEN
SCREEN 0, 1, 0, 0
   CLS
LOCATE 10, 5, 1
PRINT "Do you have a color monitor (Y/N)? [ ]";
col = POS(0)
col = col - 2
LOCATE , col, 1
CALL GetUCResponse(a$, "YN")
IF a$ = "Y" THEN
  MONITOR$ = "C"
  ELSE
  MONITOR$ = "M"
END IF
```

```
OPEN NAMEOFFILE$ FOR OUTPUT AS #1
PRINT #1, MONITOR$
CLOSE #1
   CLS
      ELSE
OPEN "I", #1, NAMEOFFILE$
LINE INPUT #1, MONITOR$
CLOSE #1
      END IF
END SUB

     SUB GetUCResponse (ch$, filter$)
'   This routine returns the character ch$ in uppercase, chosen from a
'   single character in the filter string filter$.  GetKey() is in
       __DXNARA.
    DO
 CALL GetKey(a$)
LOOP UNTIL INSTR(filter$, a$) <> 0
ch$ - a$
PRINT ch$;
END SUB

'-----------------------------------------------------------------
SUB GraphContinuePrompt
'       This routine displays the "To continue prompt in inverse."
'       No need to save variables, so not STATIC.
      CP$ - "To continue, press any key"
      CPlen - LEN(CP$)
      LOCATE 25, 26
      PRINT CP$;
      CALL inversegraph(25, 26, CPlen)
END SUB

SUB PackArray (PackString$, thearray%())
'       This routine packs the data stored in the array VARIABLE() into
'       the string a$
      PackString$ - ""
      FOR i - 0 TO 12
N% - 0
FOR j - 0 TO 14
  K% - thearray%(1 + j + i * 15)
  N% - N% OR (K% * 2 ^ j)
NEXT j
PackString$ - PackString$ + MKI$(N%)
      NEXT i



END SUB
```

```
SUB TextContinuePrompt

'       This routine displays the "To continue prompt in inverse."
'       This routine is used when in text mode.
'       ContinuePrompt is used when in graphics mode.
'       No need to save variables, so not STATIC.
      COLOR backcolor, framecolor
      LOCATE 25, 26
      PRINT "To continue, press any key";
      LOCATE 1, 1, 0
      COLOR forecolor, backcolor
END SUB

SUB UnPackArray (PackString$, thearray%())
'       This routine unpacks the data stored in the string Packstring$
'       into the array VARIABLE().

      FOR i = 0 TO 12
N% = CVI(MID$(PackString$, i * 2 + 1, 2))
FOR j = 0 TO 14
  IF (N% AND 2 ^ j) <> 0 THEN
    thearray%(1 + j + i * 15) = 1
  ELSE
    thearray%(1 + j + i * 15) = 0
  END IF
NEXT j
      NEXT i

END SUB

SUB UpdatePtrMinus (ptr)
' Used by SSNAgeDAte subprogram
  ptr = ptr - 1
  SELECT CASE ptr
    CASE 4
      ptr = 3
    CASE 7
      ptr = 6
    CASE IS < 1
      ptr = 1
    CASE ELSE
'  pass thru with no changes
  END SELECT

END SUB

SUB UpdatePtrPlus (ptr)
' Used by SSNAgeDAte subprogram
```

ABDXSUB1.BAS (cont'd)

```
  ptr = ptr + 1
  SELECT CASE ptr
    CASE 4
      ptr = 5
    CASE 7
      ptr = 8
    CASE IS > 11
      ptr = 11
    CASE ELSE
'  pass thru with no changes
  END SELECT
END SUB
```

ABDXSUB2.BAS

```
DECLARE SUB TextPause ()
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB DecryptClearWindow ()
DECLARE FUNCTION VideoPage% ()
DECLARE SUB GetKey (a$)
DECLARE SUB NarrativeHelp (WritePage%, VisualPage%)
DECLARE SUB TextContinuePrompt ()
DECLARE SUB decryptstring (instring$, outstring$)
DECLARE SUB DisplayEncryptedFile (TheFile$, ReturnPage%)
DECLARE FUNCTION Centered% (s$)
DECLARE SUB textPRINT (a$)
DECLARE SUB BoxSelections (actyl%, XPOINT%, NumOfResp%, internalwidth%)
DECLARE SUB HelpDataEntry (HLPFIL$, quest%)
DECLARE SUB PaintGraph (VAR%, WhichOne%)
DECLARE SUB BlankGraph (WhichOne%)
DECLARE SUB DrawGraph (WhichOne%)
DECLARE SUB mainkeyroutine (exitstring$, exitquest%, exitresp%,
       NUMQUEST%, Numresp%(), NUMCOL1QUESTS%, VariablePtr%(),
       VARIABLE%(), MULTIP%(), None%(), actrow%(), GRAPHFLAG%(),
       offset%, STFLAG%, Choices$())
DECLARE SUB UpdateCursors (oldquest%, oldresp%, quest%, resp%,
       Choices$())
DECLARE SUB RemoveCursor (quest%, resp%, Choices$())
DECLARE FUNCTION getkeycode% ()
DECLARE SUB versetext (row%, col%, theresponse$)
DECLARE SUB inversegraph (row%, col%, StrWidth%)
DECLARE FUNCTION VideoMode% ()
DECLARE SUB inversetext (row%, col%, theresponse$)
DECLARE SUB HPframe ()
DECLARE SUB SetColor (thecolor%)
DECLARE SUB questionPRINT (a$)
DECLARE SUB headingPRINT (a$)
DECLARE SUB responsePRINT (a$)
DECLARE SUB SetScreenMode (SMode%)
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB UpdateAsterisk (FirstRow%, Firstcol%, NonePtr%, VariPtr%,
       GraphFlg%, NumberofResp%, offset%, VARIABLE%())
DECLARE SUB PutCursor (quest%, resp%, Choices$())
DECLARE FUNCTION VideoMode% ()
DEFINT A-Z

' $INCLUDE: 'include.bas'

REM $DYNAMIC
```

```
SUB DecryptClearWindow
'       This routine clears the text window for the decrypted text
'       print out.
 'for now, later will need scrollup to save frame.
 '  Draw frame stuff here.
   CALL scrollup(2, 2, 23, 78, 0, backcolor)

END SUB

SUB DisplayEncryptedFile (TheFile$, ReturnPage%)
'       This routine will decrypt and display the help file TheFile$.
'       ReturnPage is the visual page to be set on exiting this routine.

   ' IF a$ = "?" THEN filnam$ = "HP11.DAT": GOSUB 30252: GOTO 30004
 ' move to display page 2
 SCREEN 0, 1, 2, 2
 CLS

30252 OPEN "R", 1, TheFile$, 75
   FIELD #1, 75 AS B$
   IF LOF(1) < 1 THEN
    CLOSE #1
    LOCATE 10, 10
    PRINT "File not found"
    SOUND 200, 1
    CALL TextPause
    SCREEN 0, 1, ReturnPage%, ReturnPage%
    EXIT SUB
   END IF
   RecNum = 1
   ExitFlag = 0
   ' Draw frame stuff here.
   framtyp = frametype
   SetColor (framecolor)
   CALL frame(1, 1, 22, 78, framtyp)
   SetColor (forecolor)
   SetColor (infocolor)
   infostring$ = " Enter (P)revious, (N)ext, e(X)it, or (?) for
    help."
   LOCATE 25, Centered%(infostring$) - 4
   PRINT infostring$;
   SetColor (forecolor)
   DO
    rowptr = 2
    DO
30280     GET #1, RecNum
     decryptedstring$ = B$
     'decipher string
     CALL decipher(decryptedstring$)
```

```
  IF MID$(decryptedstring$, 1, 1) <> "|" THEN
    LOCATE rowptr, 3
    PRINT decryptedstring$;
    rowptr = rowptr + 1
  END IF
  'check to see if not finding '|', and therefore, file is not
  'in proper format.
  IF rowptr > 24 THEN
    CLOSE #1
    CALL DecryptClearWindow
    LOCATE 10, 10
    PRINT "File is in improper format.  Cannot read."
    SOUND 200, 1
    CALL TextPause
    SCREEN 0, 1, ReturnPage%, ReturnPage%
    EXIT SUB
  END IF
  RecNum = RecNum + 1
LOOP UNTIL LEFT$(decryptedstring$, 1) = "|"
'print bottom header
SetColor (infocolor)
LOCATE 25, 65, 0
PRINT " " + MID$(decryptedstring$, 3, 13);
SetColor (forecolor)
COLOR 7, 0

DO
  CALL GetKey(a$)
  SELECT CASE a$
    CASE "N"
      IF MID$(decryptedstring$, 2, 1) = "|" THEN
        ExitFlag = 1
      ELSE
        CALL DecryptClearWindow
      END IF
    CASE "P"
      RecNum = VAL(MID$(decryptedstring$, 16, 3))
      IF RecNum < 0 THEN
        ExitFlag = 1
      ELSE
        CALL DecryptClearWindow
      END IF
    CASE "X", CHR$(27)
      ExitFlag = 1
      a$ = "X"
    CASE "?"
      CALL NarrativeHelp(2, 2)
    CASE ELSE
      SOUND 200, 1
```

```
     END SELECT
    LOOP UNTIL INSTR("NPX", a$) <> 0
   LOOP UNTIL ExitFlag = 1
   CLOSE #1
   SCREEN 0, 1, ReturnPage%, ReturnPage%
END SUB

SUB inversegraph (row, col, StrWidth)
'    This routine inverses strwidth characters at location row and
    col
'    while in graphics mode.
   ' $DYNAMIC
   REDIM INVERSEARRAY%(1000)
   temrow = row - 1
   temcol = col
   IF Vertbits = 14 THEN WINDOW
   GET ((temcol - 1) * 8 - 1, temrow * Vertbits)-((temcol + StrWidth -
    1) * 8, (temrow + 1) * Vertbits - 1), INVERSEARRAY%(0)
   PUT ((temcol - 1) * 8 - 1, temrow * Vertbits), INVERSEARRAY%(0),
    PRESET
   IF Vertbits = 14 THEN
    WINDOW SCREEN (0, 0)-(639, 199)
   END IF
END SUB

REM $STATIC
SUB inversetext (row, col, theresponse$)
'    This routine prints theresponse$ in inverse print at location
'    row and col.
 COLOR backcolor, textcolor
 LOCATE row, col, 0
 PRINT theresponse$;
 COLOR textcolor, backcolor

END SUB

SUB mainkeyroutine (exitstring$, quest%, resp%, NUMQUEST%, Numresp%(),
    NUMCOL1QUESTS%, VariablePtr%(), VARIABLE%(), MULTIP%(), None%(),
    actrow%(), GRAPHFLAG%(), offset%, STFLAG%, Choices$())
'    This is the main key entry loop for moving the cursor and
    updating
'    VARIABLE during data entry.
' exitstring$ - character returned by routine upon exiting (prob key
    pressed.)
' quest%      - returns relative question number where cursor is upon
    exiting.
' resp%         - returns response for question quest as above.
' NUMQUEST%     - number of questions on the display page.
' Numresp%()      - number of responses for each question on the display
     page.
' NUMCOL1QUESTS% - number of questions in the first column.
' VariablePtr%() - pointer to position in array VARIABLE() for first
'                  response of question ().
' VARIABLE%() - array containing the entered case data to check and
     update.
' MULTIP%()   - 1 if multpile responses allowed, 0 if not.
' None%()     - # of response to 0 out all others.  0 if no item clears
     others.
' actrow%()   - array containing the actual row of the first reponse of
     question.
' GRAPHFLAG%() - 0 if no graph drawn, 1 if graph is drawn.
' offset%     - offset for printing asterisks in UpdateAsterisk.
' STFLAG%     - flags changes made to data.  0 if no changes, 1 if
     changes made.
' Choices$()  - 2-d array containing responses for the display page.

'initialize stuff
  CONST homekey = &H4700
  CONST endkey = &H4F00
  CONST uparrow = &H4800
  CONST downarrow = &H5000
  CONST leftarrow = &H4B00
  CONST rightarrow = &H4D00
  CONST tabkey = 9
  CONST shifttabkey = &HF00
  CONST esc = 27
  CONST CR = 13
  IF resp > 0 THEN
    oldresp = resp
  ELSE
    oldresp = 1
    resp = 1
  END IF
  IF quest > 0 THEN
    oldquest = quest
  ELSE
    oldquest = 1
    quest = 1
  END IF
  exitstring$ = ""

  DO
    keynum = getkeycode
    SELECT CASE keynum

    CASE homekey
      resp = 1
  quest = 1
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE endkey
  quest = NUMQUEST
  resp = Numresp(quest)
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE uparrow
  resp = resp - 1
  IF resp < 1 THEN
    quest = quest - 1
    IF quest < 1 THEN
      quest = NUMQUEST
    END IF
    resp = Numresp(quest)
  END IF
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE downarrow
  resp = resp + 1
  IF resp > Numresp(quest) THEN
    resp = 1
    quest = quest + 1
    IF quest > NUMQUEST THEN
      quest = 1
    END IF
  END IF
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE leftarrow, rightarrow
  'Will check to see if a response is at the same row in the
  ' different col. (will use same col if only one col or if
   currently
  ' in 2nd col.) (Note - col here means one of the two col's of
   questions.
  'determine which column to use
  IF (quest <= NUMCOL1QUESTS) AND (NUMQUEST% > NUMCOL1QUESTS) THEN
    ' use 2nd column
    QuestStart = NUMCOL1QUESTS + 1
    QuestFinish = NUMQUEST%
  ELSE
    'use first column.
    QuestStart = 1
    QuestFinish = NUMCOL1QUESTS
  END IF
  'get current row
  thisrow = actrow(quest) + resp - 1
  FOR LRQuestPtr = QuestStart TO QuestFinish
```

```
    FOR LRRespPtr = 0 TO Numresp%(LRQuestPtr) - 1
      ' compare current row to test row
      IF thisrow = (actrow(LRQuestPtr) + LRRespPtr) THEN
        quest = LRQuestPtr
        resp = LRRespPtr + 1
      END IF
    NEXT LRRespPtr
  NEXT LRQuestPtr

  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE tabkey
  quest = quest + 1
  IF quest > NUMQUEST THEN quest = 1
  resp = 1
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE shifttabkey
  quest = quest - 1
  IF quest < 1 THEN quest = NUMQUEST
  resp = 1
  CALL UpdateCursors(oldquest, oldresp, quest, resp, Choices$())

CASE esc, ASC("X"), ASC("x")
  exitstring$ = "X"

CASE ASC("N"), ASC("n"), ASC("P"), ASC("p")
 exitstring$ = UCASE$(CHR$(keynum))

CASE ASC("?")
 exitstring$ = CHR$(keynum)

CASE CR
  'First check if questions update variables.
  VariPtr = VariablePtr%(quest)
  IF VariPtr = 0 THEN
    exitcode = resp
    EXIT SUB
  END IF
  SpecificVarPtr = VariPtr + resp - 1
  IF VARIABLE(SpecificVarPtr) = 1 THEN
    VARIABLE(SpecificVarPtr) = 0
  ELSE
    VARIABLE(SpecificVarPtr) = 1
  END IF
  STFLAG = 1
  'chck if multip answers not allowed.  If so, clear other responses
  'in VARIABLE()
  IF VARIABLE(SpecificVarPtr) = 1 AND MULTIP(quest) = 0 THEN
  ' response was set and other responses are not allowed
  FOR i = 1 TO Numresp(quest)
   IF i <> resp THEN VARIABLE(VariPtr + i - 1) = 0
  NEXT i
 END IF

 'if not noneptr, then clear noneptr in VARIABLE(), so that
 ' updateasterisks routine will work properly.
 NonePtr = None%(quest)
 IF VARIABLE(SpecificVarPtr) = 1 THEN
  IF NonePtr <> 0 AND NonePtr <> resp THEN
   VARIABLE(VariPtr + NonePtr - 1) = 0
  END IF
 END IF

 'clean up graph if present.
 'If VARIABLE(SpecificVarPtr)=0 and here, then it was just set.
 'Therefore, need to erase corresponding portion of graph.
 IF VARIABLE(SpecificVarPtr) = 0 AND GRAPHFLAG(quest) THEN
  WhichOne = GRAPHFLAG(quest)
  'first start with clean graph frame
  CALL BlankGraph(WhichOne)
  CALL DrawGraph(WhichOne)
  'now paint all selected responses
  FOR tempresp = 1 TO Numresp(quest)
   IF VARIABLE(VariPtr + tempresp - 1) = 1 THEN
    CALL PaintGraph(tempresp, WhichOne%)
   END IF
  NEXT tempresp

 END IF

 'update asterisks
 FirstRow = actrow(quest)
 IF quest > NUMCOL1QUESTS THEN
  Firstcol = ShareTOPCOL2
 ELSE
  Firstcol = ShareTOPCOL
 END IF
 GraphFlg = GRAPHFLAG(quest)
 NumberofResp = Numresp(quest)
 offset = 0
 CALL UpdateAsterisk(FirstRow, Firstcol, NonePtr, VariPtr,
  GraphFlg, NumberofResp, offset, VARIABLE())

CASE ELSE
  error
 SOUND 999, 1
```

```
    END SELECT
  LOOP UNTIL exitstring$ <> ""

END SUB

 SUB MenuEntryPage (NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
'        This routine displays a single menu on the screen and
'        obtains the input.  It is analagous to the DataEntryPage for the
'        H&P section.
'        NR%             - response upon exit. [0, 1-resplength%]
'                        - on entance, cursor placed at value
'        resplength%     - number of responses for the menu
'        exitchar$       - char upon exit, X, or null
'        DATAHEADING$    - Display page heading
'        menuheading$    - heading over menu box
'        Choices$()      - list.of responses, here will be (1,[1-
       resplength%]
'        HELPFILE$       - Help file name. File is encrypted.

    DIM Numresp%(2), dumvar%(2) ' for mainkeyentry routine below

    HeadingRow = 7      'location of first response of menu
    MenuWidth = LEN(Choices$(1, 1))
    CenterCol = Centered(Choices$(1, 1))


    SCREEN 0, 1, 0, 0
    CLS
    CALL LocateCenter(1, DATAHEADING$)
    CALL headingPRINT(DATAHEADING$)
    CALL LocateCenter(HeadingRow - 2, menuheading$)
    CALL textPRINT(menuheading$)

    FOR i = 0 TO resplength% - 1
      LOCATE HeadingRow + i, CenterCol
      CALL responsePRINT(Choices$(1, i + 1))
    NEXT i

    ' Box the menu.
    CALL BoxSelections(HeadingRow, CenterCol - 1, resplength%,
       MenuWidth)

    ' Information.
    SetColor (infocolor)
    LOCATE 23, 1: PRINT "Use the arrow keys to move the cursor to the
       desired position.  Push RETURN";
    LOCATE 24, 1: PRINT "to select the desired response or '?' for more
       information.";
    SetColor (forecolor)


    ' Set up so that can use it with PutCursor and RemoveCursor
       routines.
    Xwidth = MenuWidth
    ShareTOPCOL = CenterCol - 4
    ShareTOPCOL2 = ShareTOPCOL
    actrow(1) = HeadingRow
    NUMQUEST = 1
    Numresp(1) = resplength%
    curquest = 1
    curresp = NR%
    ShareNUMCOL1QUESTS = 1
    dumvar(1) = 0
    offset = 0
    dummySTFLAG% = 0

    'initialize cursor on page at first question, first response
    CALL PutCursor(curquest, curresp, Choices$())

    'now comes the key entry routine.
    DO
      CALL mainkeyroutine(exitstring$, curquest%, curresp%, NUMQUEST,
       Numresp(), NUMCOL1QUESTS, dumvar%(), dumvar(), dumvar(),
       dumvar%(), actrow(), dumvar(), offset, STFLAG%, Choices$())
      exitchar$ = exitstring$
      'print help for H&P questions
      IF exitstring$ = "?" THEN
        thispage% = VideoPage%
        CALL DisplayEncryptedFile(HELPFILE$, thispage%)
      END IF
    LOOP UNTIL (curresp <> 0 AND exitstring$ = "")
    NR% = curresp

    ERASE Numresp%, dumvar%

END SUB

SUB MenuSummaryPage (MenuRow, MenuCol, NR%, resplength%, exitchar$,
      menuheading$, Choices$(), HELPFILE$)
'       This routine displays a single menu on the summary page and
'       obtains the input.  It is analagous to the DataEntryPage for the
'       H&P section, and is very similar to MenuEntryPage.
'       MenuRow          - row location of first response of menu
'       MenuCol          - col location of first response of menu
'       NR%              - response upon exit. [0, 1-resplength%]
'                        - on entance, cursor placed at value
'       resplength%      - number of responses for the menu
'       exitchar$        - char upon exit, X, or null
'       menuheading$     - heading over menu box
'       Choices$()       - list of responses, here will be (1,[1-
      resplength%]
'       HELPFILE$        - Help file name. File is encrypted.

    DIM Numresp%(2), dumvar%(2) ' for mainkeyentry routine below

    MenuWidth = LEN(Choices$(1, 1))
    HeadingRow = MenuCol + (MenuWidth - LEN(menuheading$)) \ 2
    LOCATE MenuRow - 2, HeadingRow
    CALL textPRINT(menuheading$)

    FOR i = 0 TO resplength% - 1
      LOCATE MenuRow + i, MenuCol
      CALL responsePRINT(Choices$(1, i + 1))
    NEXT i

    ' Box the menu.
    CALL BoxSelections(MenuRow, MenuCol - 1, resplength%, MenuWidth)

    ' Information.
    SetColor (infocolor)
    LOCATE 23, 1: PRINT "Use the arrow keys to move the cursor to the
       desired position.  Push RETURN";
    LOCATE 24, 1: PRINT "to select the desired response or '?' for more
       information.";
    SetColor (forecolor)


    ' Set up so that can use it with PutCursor and RemoveCursor
       routines.
    Xwidth = MenuWidth
    ShareTOPCOL = MenuCol - 4
    ShareTOPCOL2 = ShareTOPCOL
    actrow(1) = MenuRow
    NUMQUEST = 1
    Numresp(1) = resplength%
    curquest = 1
    curresp = NR%
    ShareNUMCOL1QUESTS = 1
    dumvar(1) = 0
    offset = 0
    dummySTFLAG% = 0

    'initialize cursor on page at first question, first response
    CALL PutCursor(curquest, curresp, Choices$())

    'now comes the key entry routine.
    DO
      CALL mainkeyroutine(exitstring$, curquest%, curresp%, NUMQUEST,
       Numresp(), NUMCOL1QUESTS, dumvar%(), dumvar(), dumvar(),
       dumvar%(), actrow(), dumvar(), offset, STFLAG%, Choices$())
      exitchar$ = exitstring$
      'print help for H&P questions
      IF exitstring$ = "?" THEN
        thispage% = VideoPage%
        CALL DisplayEncryptedFile(HELPFILE$, thispage%)
      END IF
    LOOP UNTIL (curresp <> 0 AND exitstring$ = "") OR exitstring$ = "?"
    NR% = curresp

    ERASE Numresp%, dumvar%

END SUB

SUB PutCursor (quest, resp, Choices$())
'       This routine places the high-lighted cursor at the appropriate
'       response.  It checks to see if in graphics or text mode and uses
'       the appropriate method for inversing the text.

'       Variables needed from DataEntryPage routine.
'               actrow(),Xwidth,ShareNUMCOL1QUESTS,ShareTOPCOL2

' compute row and col
  row = actrow(quest) + resp - 1
  IF quest <= ShareNUMCOL1QUESTS THEN
    col = ShareTOPCOL
  ELSE
    col = ShareTOPCOL2
  END IF
  col = col + 4
' check screen mode
'       VideoMode =3    screen 0, width 80
'                  6    screen 2
'                  16   screen 9

  IF VideoMode = 3 THEN
    ' text mode
    theresponse$ = LEFT$(Choices$(quest, resp) + SPACE$(Xwidth), Xwidth)
    CALL inversetext(row, col, theresponse$)
  ELSE
    'else graphics mode
    StrWidth = Xwidth
    CALL inversegraph(row, col, StrWidth)
  END IF
END SUB
```

```
SUB RemoveCursor (quest, resp, Choices$())
'       This routine clears the high-lighted cursor at the appropriate
'       response.  It checks to see if in graphics or text mode and uses
'       the appropriate method for versing the text.
'       NOTE - If in graphics mode, it just inverses the inverse text.
'       If used on normal text, it will inverse it.  In text mode,
       nothing
'       would happen.

'       Variables needed from DataEntryPage routine.
'          Xwidth

' compute row and col
  row = actrow(quest) + resp - 1
  IF quest <= ShareNUMCOL1QUESTS THEN
    col = ShareTOPCOL
  ELSE
    col = ShareTOPCOL2
  END IF
  col = col + 4

' check screen mode
'       VideoMode =3     screen 0, width 80
'                  6     screen 2
'                  16    screen 9

  IF VideoMode = 3 THEN
    ' text mode
    theresponse$ = LEFT$(Choices$(quest, resp) + SPACE$(Xwidth), Xwidth)
    CALL versetext(row, col, theresponse$)
  ELSE
    'else graphics mode
    StrWidth = Xwidth
    CALL inversegraph(row, col, StrWidth)
  END IF

END SUB

SUB UpdateAsterisk (FirstRow%, Firstcol%, NonePtr%, VariPtr%, GraphFlg%,
       NumberofResp%, offset%, VARIABLE())
'       This routine updates the asterisks for the responses by checking
'       the appropriate areas of VARIABLE().  offset is the offset for
'       printing the asterisks.  offset can be + or -, but is usually 0.
'needs:
       VARIABLE(),firstrow,firstcol,noneptr,VariPtr,GraphFlg,NumberofRes
       p,offset
'       firstrow,firstcol  - location of asterisks for first choice
'       noneptr            - location of none/no pain if present
'       VariPtr        - ptr to appropriate area of VARIABLE()
'       GraphFlg                 - flag for graphics routines
'       NumberofResp                   - number of choices
'       offset                    - as above
'       VARIABLE()                - array containing the data to check.

   blankasterisk$ = "  "
   asterisk$ = "**"

   ' check first for no/none response
   IF NonePtr <> 0 AND VARIABLE(VariPtr + NonePtr - 1) = 1 THEN
     FOR cnt = 0 TO NumberofResp - 1
       LOCATE FirstRow + cnt, Firstcol + offset, 0
       IF cnt <> NonePtr - 1 THEN
         VARIABLE(VariPtr + cnt) = 0
         PRINT blankasterisk$;
       ELSE
         PRINT asterisk$;
         IF GraphFlg > 0 THEN           're-init graph frame
           WhichOne = GraphFlg
           CALL BlankGraph(WhichOne)
           CALL DrawGraph(WhichOne)
         END IF

       END IF
     NEXT cnt
   ELSE
     FOR cnt = 0 TO NumberofResp - 1
       LOCATE FirstRow + cnt, Firstcol + offset, 0
       IF VARIABLE(VariPtr + cnt) = 1 THEN
         PRINT asterisk$;
         IF GraphFlg > 0 THEN
           WhichOne = GraphFlg
           CALL PaintGraph(cnt + 1, WhichOne%)
         END IF
       ELSE
         PRINT blankasterisk$;
       END IF
     NEXT cnt
   END IF

END SUB

SUB UpdateCursors (oldquest, oldresp, quest, resp, Choices$())
'       This routine clears the response at oldquest,oldresp and
'       inverses the response number resp, question quest.

       CALL RemoveCursor(oldquest, oldresp, Choices$())
       CALL PutCursor(quest, resp, Choices$())
       oldquest = quest
```

```
      oldresp = resp

END SUB

  SUB versetext (row, col, theresponse$)
'       This routine prints theresponse$ in normal "verse" print at
      location
'       row and col.

  COLOR textcolor, backcolor
  LOCATE row, col, 0
  PRINT theresponse$;

END SUB

FUNCTION VideoMode%
'       This function returns the current video mode.

'       VideoMode =1    screen 0, width 40
'                  3    screen 0, width 80
'                  6    screen 2
'                 16    screen 9

  DIM reg AS RegType

  reg.ax = &HF00
  CALL interrupt(&H10, reg, reg)
  VideoMode = reg.ax AND &HFF
END FUNCTION

FUNCTION VideoPage%
'       This function returns the current video page.

'       VideoPage =0    this is the default.  Max number will depend
'                       on the mode.  For screen 0, the values are
'                       0 - 3.

  DIM reg AS RegType

  reg.ax = &HF00
  CALL interrupt(&H10, reg, reg)
  thepage% = (CLNG(reg.bx) AND &HFF00) \ 256
  VideoPage = thepage%
END FUNCTION
```

ABDXSUB3.BAS

```
DEFINT A-Z
DECLARE FUNCTION VideoMode% ()
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB TextDxPause ()
DECLARE SUB SetColor (thecolor%)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB GetKey (a$)
DECLARE SUB TextContinuePrompt ()
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB DisplayHPgetstatments (SXloc%(), SXresp$(), abortHP%)
DECLARE SUB DisplayHProwcol (row%, col%, sxstrng$)
DECLARE SUB DisplayHPFrame (TRAINING%, SIMULATE%, SSN$, STARTIME$,
       STARTDATE$)
DECLARE SUB DisplayHPprint (HP%, SXloc%(), SXresp$(), VARIABLE%())
DECLARE SUB DisplayHPhelp ()
DECLARE SUB DisplayHPTitle (HP%)
DECLARE FUNCTION Exists% (filnam$)

' $INCLUDE: 'include.bas'

SUB CompareAbdDXes (COMPAR%(), VARIABLE%(), MAXNUM%, HMDX%, QUESTPTR%(),
       QUESTIONS$())
'       This routine searches the compar array for questions to recheck.
'       This is used only for the original abdominal pain database.

      ' uses questioncolor, forecolor, infocolor

  DIM DIFFER%(15)

51390 COMFLAG = 0
      FOR i = 1 TO 15
IF COMPAR%(MAXNUM, HMDX, i) = 0 THEN EXIT FOR
IF VARIABLE%(COMPAR%(MAXNUM, HMDX, i)) = 0 THEN
  COMFLAG = COMFLAG + 1
  DIFFER%(COMFLAG) = COMPAR%(MAXNUM, HMDX, i)
END IF
      NEXT i
51430 IF COMFLAG = 0 THEN
LOCATE 15, 1: PRINT "    At this time, the computer-generated
       probabilities DO NOT AGREE with";
LOCATE 16, 1: PRINT "your preliminary diagnosis.  However, in this case,
       there are no";
```

```
LOCATE 17, 1: PRINT "specific categories which would differentiate your
       preliminary";
LOCATE 18, 1: PRINT "diagnosis from the current program-generated
       diagnosis.";
CALL TextDxPause
      ELSE
SUM = 11: J = 1: N = 1: COMFLAG1 = COMFLAG
FOR i = 1 TO 37
  SUM = SUM + QUESTPTR%(i)
  IF DIFFER%(J) < SUM THEN
    DIFFER%(N) = i
    J = J + 1: N = N + 1
  END IF
51472     IF J > COMFLAG THEN GOTO 51500
  IF DIFFER%(J) < SUM THEN
    J = J + 1
    COMFLAG1 = COMFLAG1 - 1
    GOTO 51472
  END IF
  IF N > COMFLAG1 THEN GOTO 51500
NEXT i
51500   LOCATE 15, 1: PRINT "  At this time the computer-generated
       probabilities DO NOT AGREE with";
LOCATE 16, 1: PRINT "your preliminary diagnosis.  The following
       categories are particularly";
LOCATE 17, 1: PRINT "useful in differentiating your diagnosis from the
       others.  It may be";
LOCATE 18, 1: PRINT "helpful to review your input in these areas and
       make any changes you";
LOCATE 19, 1: PRINT "consider appropriate.";
FOR i = 23 TO 25: LOCATE i, 1: PRINT SPACE$(75); : NEXT i
dummyc = questioncolor
CALL SetColor(dummyc)
FOR i = 1 TO COMFLAG1
  XI = i: YI = 1
  IF i > 4 THEN XI = i - 4: YI = 26
  IF i > 10 THEN XI = i - 10: YI = 51
  LOCATE 20 + XI, YI: PRINT QUESTIONS$(DIFFER%(i));
NEXT i

'Maybe do away with this, because the HM has the option of changing
'his responses on the summary page.
'        dummyc = forecolor
'        CALL SetColor(dummyc)
'        SetColor (infocolor)
'        infostring$ = "Would you like to make any changes? (Y or N) [
       ]"
'        CALL LocateCenter(25, infostring$)
'        PRINT infostring$;
'        SetColor (forecolor)
'        col = POS(0) - 2
'        row = CSRLIN
'        LOCATE row, col, 1
'        DO
'          CALL GetKey(a$)
'          SELECT CASE a$
'            CASE "Y"                  ' Wants to make changes.
'              PRINT a$
'              GOTO 100
'            CASE "N"                  ' No changes desired.
'              PRINT a$
'              GOTO 52000
'            CASE ELSE                 ' Can't follow directions.
'              SOUND 200, 1
'          END SELECT
'        LOOP UNTIL 1 = 2              ' loop forever. Will exit when
       necessary.

      END IF
      CALL TextContinuePrompt
      CALL GetKey(a$)

      ERASE DIFFER%

END SUB

REM $DYNAMIC
 SUB DisplayHP (TRAINING, SIMULATE, SSN$, STARTIME$, STARTDATE$,
       VARIABLE%())
'        This routine was orginally at 53000.  It displays the brief
       summary
'        of findings on a history page and a physical exam page.

  DIM SXloc%(1 TO 180)
  DIM SXresp$(1 TO 180)

  CONST HX = 0
  CONST PE = 1
  HP = HX
  ' get comments for each response
  CALL DisplayHPgetstatments(SXloc%(), SXresp$(), abortHP%)
  IF abortHP = 1 THEN
    ERASE SXloc%, SXresp$
    EXIT SUB
  END IF


  'draw heading and frame
  CALL DisplayHPFrame(TRAINING, SIMULATE, SSN$, STARTIME$, STARTDATE$)

  'display HX items first
  CALL DisplayHPprint(HP, SXloc%(), SXresp$(), VARIABLE%())

  QuitDo = 0
  DO
    DO
      CALL GetKey(a$)
      'filter for PNX?
      PNX$ = "PNX?" + CHR$(27)
    LOOP UNTIL INSTR(PNX$, a$) <> 0
    SELECT CASE a$
      CASE "X", CHR$(27)
QuitDo = 1

      CASE "?"
CALL DisplayHPhelp
      CASE "P"
IF HP = HX THEN
  QuitDo = 1
  ELSE
  HP = HX
  CALL DisplayHPprint(HP, SXloc%(), SXresp$(), VARIABLE%())
END IF
      CASE "N"
IF HP = HX THEN
  HP = PE
  CALL DisplayHPprint(HP, SXloc%(), SXresp$(), VARIABLE%())
  ELSE
  QuitDo = 1
END IF

      CASE ELSE
'should never get here

    END SELECT
  LOOP UNTIL QuitDo <> 0

  ERASE SXloc%, SXresp$

END SUB

 SUB DisplayHPFrame (TRAINING, SIMULATE, SSN$, STARTIME$, STARTDATE$)
'       This routine draws the frame for the H&P synopsis pages.  It is
'       called only once, at the beginning of the displayH&P sub.

      ' uses frametype%, infocolor, forecolor
      SCREEN 0, 1, 0, 0
      CLS
      IF TRAINING = 1 THEN
TYP$ = "Training"
leftend = 199
rightend = 182
      ELSE
TYP$ = "Simulated"
leftend = 195
rightend = 180
IF SIMULATE = 1 THEN
  TYP$ = "Real"
END IF
      END IF

      CALL SetFrameColor
      framtyp = frametype%
      CALL frame(1, 1, 21, 78, framtyp)
      'print cross bar
      LOCATE 3, 1
      PRINT CHR$(leftend) + STRING$(78, 196) + CHR$(rightend);
      CALL SetNormalColor

      LOCATE 2, 40: PRINT "SSN: "; SSN$;
      LOCATE 2, 11: PRINT " Summary ("; TYP$; " case)";
      LOCATE 2, 63: PRINT STARTIME$; "  "; STARTDATE$;

      CALL SetColor(infocolor)
      info$ = "Enter (P)revious, (N)ext, e(X)it, or (?) for help."
      CALL LocateCenter(24, info$)
      PRINT info$;
      SetColor (forecolor)

END SUB

SUB DisplayHPgetstatments (SXloc%(), SXresp$(), abortHP%)
'       This routine opens the file ___SX.DAT and loads SXloc() and
       SXresp$().
'       SXloc(x) is the location in the VARIABLE() of response
       SXresp$(x).
'       ___ = "ABD" , "CPD". etc.

'  TYPE SXformat
'    SXlocation AS INTEGER
'    SXstring AS STRING * 21
'  END TYPE

'  DIM SX AS SXformat
```

```
  'Check for existence of file
  abortHP = 0
  filnam$ = SXDATAFILE$
  IF NOT Exists%(filnam$) THEN
    CLS
    LOCATE 10, 10
    PRINT "File "; filnam$; " not found!"
    CALL TextContinuePrompt
    CALL GetKey(a$)
    abortHP = 1
    EXIT SUB
  END IF


  OPEN filnam$ FOR INPUT AS #1
  i = 1
  DO WHILE NOT EOF(1)
    LINE INPUT #1, a$
    'decipher string
    CALL decipher(a$)
    SXloc(i) = i
    SXresp$(i) = RTRIM$(a$)
    i = i + 1
  LOOP
  CLOSE #1
  ' Check for proper number of items read in.
  IF i <> NUMBEROFITEMS + 1 THEN
    abortHP = 1
  END IF

END SUB

 SUB DisplayHPhelp
'       This routine prints the help statment for the H&P summary.

  ' uses frametype%

'frame based on length of a$
  IF TRAINING = 0 THEN          'not training case
    wid = 33
    hght = 1
  ELSE                          'training case
    wid = 38
    hght = 3
  END IF

  row = 10
  col = (80 - wid) \ 2
  SCREEN 0, 1, 1, 1
  CLS
  CALL SetFrameColor
  framtyp = frametype%
  CALL frame(row, col, hght, wid, framtyp)
  CALL SetNormalColor

  LOCATE row + 1, col + 1
  PRINT "These are the items you selected."
  PRINT a$;
  IF TRAINING = 1 THEN
    LOCATE row + 2, col + 1
    PRINT "Responses you should have chosen, but"
    LOCATE row + 3, col + 1
    PRINT "did not, are flagged with '(omitted)'."
  END IF

  CALL TextContinuePrompt

'help heading

CALL GetKey(a$)
SCREEN 0, 1, 0, 0
END SUB

  SUB DisplayHProwcol (row, col, sxstrng$)
'       This routine computes the proper location for printing the SX's
'       in the display H&P routine.

  minrow = 4
  maxrow = 22
  colincrement = 34
  IF row = 0 THEN row = minrow - 1
  IF col = 0 THEN col = 4
  row = row + 1
  IF row > maxrow THEN
    row = minrow
    col = col + 34
  END IF
  LOCATE row, col, 0
  PRINT sxstrng$;
END SUB

 SUB DisplayHPTitle (HP)
'       This prints the heading for the H&P display routine at 53000.
      IF HP = 0 THEN
heading$ = " History"
      ELSE
heading$ = "Physical"
      END IF
      LOCATE 2, 3
      PRINT heading$;

END SUB

REM $STATIC
SUB DisplayMissedHP (SSN$, STARTIME$, STARTDATE$, VARIABLE%(),
       THECASE%())
'       This routine displays any inconsistencies between the responses
'       the HM entered and the test case.

  DIM SXloc%(1 TO 180)
  DIM SXresp$(1 TO 180)

  ' get comments for each response
  CALL DisplayHPgetstatments(SXloc%(), SXresp$(), abortHP%)
  IF abortHP = 1 THEN
    ERASE SXloc%, SXresp$
    EXIT SUB
  END IF
  'draw heading and frame
  CALL DisplayHPFrame(1, 0, SSN$, STARTIME$, STARTDATE$)
  heading$ = "Incorrect Items"
  LOCATE 2, 4
  PRINT heading$;
  ibegin = 1
  iend = 180
  row = 0
  col = 0
  'cycle through and check responses
  missedsomeflag = 0
  FOR i = ibegin TO iend
    IF VARIABLE%(i) <> THECASE%(i) THEN
      CALL DisplayHProwcol(row, col, SXresp$(i))
      IF THECASE%(i) = 1 THEN
PRINT " (omitted)";
      END IF
      missedsomeflag = 1
    END IF
  NEXT i
  IF missedsomeflag = 0 THEN
    'didn't miss any, so let it be known.
    LOCATE 10, 10
    PRINT "Congratulations! No items were missed.";
  END IF
  QuitDo = 0
  DO
    DO
      CALL GetKey(a$)
      'filter for PNX?
    LOOP UNTIL INSTR("PNX?", a$) <> 0
    SELECT CASE a$
      CASE "P", "N", "X"
QuitDo = 1

      CASE "?"
CALL DisplayHPhelp
      CASE ELSE
'should never get here
 SOUND 200, 1
    END SELECT
  LOOP UNTIL QuitDo <> 0

  ERASE SXloc%, SXresp$

END SUB

REM $DYNAMIC
SUB HPframe
'       This routine draws the frame for each of the history and PE
       pages.
   'uses forecolor, hpframecolor

   SetColor (hpframecolor)
   CALL frame(1, 1, 20, 78, 1)
   IF VideoMode = 6 THEN   'screen 2
     'horiz line just below heading
     LINE (4, 20)-(636, 20), 1

     'vertical line in middle
     LINE (316, 20)-(316, 172), 1
     LINE (315, 20)-(315, 172), 1

   ELSE
     'horiz line just below heading
     topstring$ = CHR$(195) + STRING$(78, 196) + CHR$(180)
     LOCATE 3, 1
     PRINT topstring$;

     'vertical line in middle
     LOCATE 3, 40
     PRINT CHR$(194);
     FOR i% = 4 TO 21
       LOCATE i%, 40
       PRINT CHR$(179);
     NEXT i%
     LOCATE 22, 40
     PRINT CHR$(193);
    END IF
    SetColor (forecolor)

END SUB
```

ABDXSUB4.BAS

```
DECLARE SUB CenterString (infostring$)
DECLARE SUB TextPause ()
DECLARE SUB GetKey (a$)
DECLARE SUB TextContinuePrompt ()
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB GraphContinuePrompt ()
DECLARE FUNCTION Exists% (FIL$)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE FUNCTION VideoMode% ()

DEFINT A-Z
' $INCLUDE: 'include.bas'

REM $DYNAMIC
SUB BoxSelections (actyl, XPOINT, NumOfResp, internalwidth)
 '      This routine draws a box around the selections for the different
 '      option pages.

      'GLOBAL - frametype

      ulr - actyl - 1
      ulc - XPOINT
      CALL SetFrameColor
      framtyp - frametype
      CALL frame(ulr, ulc, NumOfResp, internalwidth, framtyp)
      CALL SetNormalColor

END SUB

REM $STATIC
SUB ComputeFinalProbs (NUMDISEASES%, MAXNUMBER%, MAXPROBABILITY%,
       PROB#(), FINALPROB#())
'            Calculate final probabilitiies here (FINALPROB).  Determine
' the disease (MAXNUMBER) with the greatest probability
       (MAXPROBABILITY).
' Remember that PROB#() and the other vars are local here.
'NUMDISEASES is the number of diseases to consider.

      SUMPROB# - 0: MAXPROBABILITY - 0
      FOR i - 1 TO NUMDISEASES%: SUMPROB# - SUMPROB# + PROB#(i): NEXT i
      FOR i - 1 TO NUMDISEASES%
        FINALPROB#(i) - PROB#(i) / SUMPROB# * 100
        IF MAXPROBABILITY < FINALPROB#(i) THEN
          MAXPROBABILITY - FINALPROB#(i)
          MAXNUMBER = i
        END IF
      NEXT i

END SUB

REM $DYNAMIC
FUNCTION Exists% (FIL$)
'       This function checks for the existance of the file fil$.  It
'       returns TRUE (non-zero) if present and false (zero) if not
       found.

  CONST FALSE = 0
  filenum = FREEFILE
  OPEN "R", filenum, FIL$, 1
  N% = LOF(filenum)
  CLOSE filenum
  IF N% = 0 THEN
    booltest% = FALSE
  ELSE
    booltest% = NOT FALSE
  END IF
  Exists% = booltest%

END FUNCTION

SUB headingPRINT (a$)
'       This routine prints the heading A$, using the current color
'  headingcolor as the foreground color.  Checks to see if screen mode
       is ok.

   'GLOBAL - headingcolor, forecolor

   dummyc = headingcolor
   SetColor (dummyc)
   PRINT a$;
   dummyc = forecolor
   SetColor (dummyc)

END SUB

SUB HelpDataEntry (HLPFIL$, quest)
'       This routine prints the help text for the questions in the
'       data entry section.

40000 REM  Help routine
      'Check for existence of file
      IF NOT Exists%(HLPFIL$) THEN
        CLS
        SCREEN 0, 1, 0
        CLS
        LOCATE 10, 10
        PRINT "Help file not found."
        SOUND 200, 5
        CALL TextPause
        EXIT SUB
      END IF

'            Open file containing definitions.
      OPEN "I", #2, HLPFIL$

'            Put instructions at the bottom of the screen.
40215 FLAG = 0: CLS
      SCREEN 0, 1, 0
      CLS
      CALL SetColor(hpframecolor)
      CALL frame(1, 1, 22, 78, 1)
      CALL TextContinuePrompt
      CALL SetNormalColor

'            Check for the existence of a definition.  If there is
' No definition, tell the user, close the file and
' return to the symptom page.
       LOCATE 2, 2, 0
       DO WHILE ((NOT EOF(2)) AND (HLP <= quest))
        INPUT #2, HLP
        IF NOT EOF(2) THEN
          LINE INPUT #2, HLPline$
'            Definition exists.  Display it
          IF HLP = quest THEN
            IF FLAG = 0 THEN
              CALL SetColor(questioncolor)
              CALL CenterString(LTRIM$(RTRIM$(HLPline$)))
              LOCATE CSRLIN + 1
              CALL SetNormalColor
              FLAG = 1
            ELSE
              LOCATE , 3: PRINT HLPline$
            END IF
          END IF
        END IF
      LOOP
      CLOSE #2
'            No Definition Exists for the Symptom.
40100 IF FLAG = 0 THEN
        LOCATE 10, 10
        PRINT "Sorry, no extra information exists for this question."
      END IF
      CALL GetKey(a$)
      EXIT SUB


END SUB

SUB hpresponsePRINT (a$)
'       This routine prints the response A$, using the current color
'  hpresponsecolor as the foreground color.
'  Checks to see if screen mode is ok. This routine is used on the H&P
'  pages.

   'GLOBAL - hpresponsecolor, forecolor

   dummyc = hpresponsecolor
   SetColor (dummyc)
   PRINT a$;
   dummyc = forecolor
   SetColor (dummyc)

END SUB

SUB questionPRINT (a$)
'       This routine prints the question A$, using the current color
'  questioncolor as the foreground color.  Checks to see if screen mode
       is ok.

   'GLOBAL - questioncolor, forecolor

   dummyc = questioncolor
   SetColor (dummyc)
   PRINT a$;
   dummyc = forecolor
   SetColor (dummyc)

END SUB

REM $STATIC
SUB ResetVariables (VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
'       This routine resets the above variables to their respective
       defaults.

      FOR i = 1 TO 200
        VARIABLE%(i) = 0
      NEXT i
      sex$ = "_"
      SSN$ = "___-__-____"
      AGE$ = "__"
      STARTIME$ = ""
      STARTDATE$ = ""

END SUB

REM $DYNAMIC
SUB responsePRINT (a$)
'       This routine prints the response A$, using the current color
'  responsecolor as the foreground color.  Checks to see if screen mode
       is ok.

   'GLOBAL - responsecolor, forecolor

   dummyc = responsecolor
   SetColor (dummyc)
   PRINT a$;
   dummyc = forecolor
   SetColor (dummyc)

END SUB

SUB SetColor (thecolor)
'This routine sets the color THECOLOR if in appropriate mode.
   smode% = VideoMode%
   SELECT CASE smode%
     CASE 16     'screen 9
      COLOR thecolor
     CASE 6      'screen 2
       'Do nothing.  COLOR gives error in mode 2.
     CASE 3      'screen 0
      COLOR thecolor
     CASE ELSE
     'Do nothing
   END SELECT

END SUB

SUB SetFrameColor
'This routine sets the frame color if in appropriate mode.

      'GLOBAL - framecolor

      CALL SetColor(framecolor)

END SUB

      SUB SetNormalColor
'       This routine returns the color attributes to the normal color
'       if color is allowed in the current screen mode.
```

```
      'GLOBAL - forecolor

      CALL SetColor(forecolor)

END SUB

SUB SetScreenMode (smode)
' Routine sets the proper screen mode and clears screen.

   SELECT CASE smode
     CASE 9
       SCREEN 9, 0, 0, 0
     CASE 2
       SCREEN 2, 0, 0, 0
       OUT &H3D9, 7
     CASE ELSE
       SCREEN 0
       WIDTH 80
   END SELECT
   CLS
END SUB

SUB TextDxPause
'       This routine calls TextContinuePrompt and then waits for a
       keypress.
'       It is used by HM DX page, also to erase info at bottom of screen
'       before writing prompt.
  FOR i = 23 TO 25
    LOCATE i, 1
    PRINT SPACE$(75);
  NEXT i
  CALL TextPause

END SUB

SUB TextPause
'       This routine calls TextContinuePrompt and then waits for a
       keypress.
  CALL TextContinuePrompt
  CALL GetKey(a$)

END SUB

   SUB textPRINT (a$)
'       This routine prints the text string A$, using the current color
'  textcolor as the foreground color.  Checks to see if screen mode is
       ok.
    'GLOBAL - textcolor, forecolor

    SetColor (textcolor)
    PRINT a$;
    SetColor (forecolor)

END SUB
```

ABDXSUB5.BAS

```
DECLARE SUB returnadapter (adapt%)
DEFINT A-Z

'CALL returnadapter(a)
'PRINT a
'STOP




1 ScrnERR = 1
  RESUME NEXT

SUB returnadapter (adapt)
'       This routine returns the display adapter type as:
'       0 = no graphics adapter
'       1 = CGA adapter
'       2 = EGA/VGA adapter

  SHARED ScrnERR
  'start error checking
  ON ERROR GOTO 1

  'check for EGA/VGA.
  ScrnERR = 0
  SCREEN 9
  IF ScrnERR = 0 THEN
    adapt = 2
  ELSE
    ScrnERR = 0
    'check for CGA
    SCREEN 2
    IF ScrnERR = 0 THEN
      adapt = 1
    ELSE
      adapt = 0
    END IF
  END IF
  'reset screen
  SCREEN 0

  'end error checking
  ON ERROR GOTO 0
END SUB
```

ABDXSUB6.BAS

```
DECLARE SUB UnPackDatabase (filename$, VARIABLE!(), APRIORI#(),
       arraywidth%, arraylength%)
DEFINT A-Z
' $INCLUDE: 'include.bas'

SUB UnPackDatabase (filename$, VARIABLE!(), APRIORI#(), arraywidth,
       arraylength)
'       This routine reads in a database in file filename$, and places
' the data in VARIABLE().  arraywidth refers to the number diseases and
' arraylength refers to the actual number of response items.
'  The first two records are used to store the apriori probabilites for
' each disease.  Record 1 contains the whole integer mantissa; record 2
' has the exponent (stored in data file as positive, but converted to
       neg).
'  The database is packed to a single byte per VARIABLE element.
'  If the byte value is less than 128 then a straight conversion is
       used.
'  If the value is > 128, then 128 is subtracted brom the byte and the
' result is dividied by 10.  ex  25 -> 25; 129 -> 0.1

  filenum = FREEFILE
  OPEN "R", filenum, filename$, arraywidth
  FIELD #filenum, arraywidth AS datarow$
  N% = LOF(filenum) \ arraywidth
  IF N% = 0 THEN
    BEEP
    PRINT "Database file not found.  Unable to continue."
    STOP
  END IF

  '  Get apriori values from the first two records.
  GET #filenum, 1
  '  Get mantissa as integer; implies a maximum of two digits precision
  FOR j = 1 TO arraywidth
    APRIORI#(j) = ASC(MID$(datarow$, j, 1))
  NEXT j

  GET #filenum, 2
  '  Get exponent as integer; always converted to negative.
  FOR j = 1 TO arraywidth
    APRIORI#(j) = APRIORI#(j) * 10 ^ (-1 * (ASC(MID$(datarow$, j, 1))))
  NEXT j

  FOR i = 1 TO arraylength
    GET #filenum, i + 2
    FOR j = 1 TO arraywidth
```

ABDXSUB6.BAS (cont'd)

```
      value = ASC(MID$(datarow$, j, 1))
      IF value > 128 THEN
        VARIABLE!(i, j) = (value MOD 128) / 10
      ELSE
        VARIABLE!(i, j) = value
      END IF
    NEXT j

  NEXT i
  CLOSE #filenum
END SUB
```

DATES.BAS

```
DECLARE FUNCTION juliantodate$ (julian&)
DECLARE FUNCTION mdytodate$ (month%, day%, year%)
DECLARE FUNCTION ValidDate% (dat$)
DECLARE FUNCTION DateToJulian& (dat$)
DEFINT A-Z

DEFSNG A-Z
FUNCTION DateToJulian& (dat$)
'       This function returns the numerical Julian date for dat$ (MM-DD-
       YYYY)

'       Program modified from one found in:
'              "The Microsoft QuickBASIC Programmer's Toolbox"
'               by John Clark Craig, Microsoft press, 1988

  month% = VAL(LEFT$(dat$, 2))
  day% = VAL(MID$(dat$, 4, 2))
  year% = VAL(RIGHT$(dat$, 4))
  IF year% < 1583 THEN
  ' year is <1583. aborting
    DateToJulian& = -1
    EXIT FUNCTION
  END IF
  IF month% > 2 THEN
    month% = month% - 3
  ELSE
    month% = month% + 9
    year% = year% - 1
  END IF
  tempa& = 146097 * (year% \ 100) \ 4
  tempb& = 1461& * (year% MOD 100) \ 4
  tempc& = (153 * month% + 2) \ 5 + day% + 1721119
  DateToJulian& = tempa& + tempb& + tempc&

END FUNCTION

FUNCTION juliantodate$ (julian&)
'       Returns a date in MM-DD-YYYY format from Julian day number
       julian&.

'       Program modified from one found in:
'              "The Microsoft QuickBASIC Programmer's Toolbox"
'               by John Clark Craig, Microsoft press, 1988

  x& = 4 * julian& - 6884477
  y& = (x& \ 146097) * 100
  d& = (x& MOD 146097) \ 4

  x& = 4 * d& + 3
  y& = (x& \ 1461) + y&
  d& = (x& MOD 1461) \ 4 + 1

  x& = 5 * d& - 3
  m& = x& \ 153 + 1
  d& = (x& MOD 153) \ 5 + 1
  IF m& < 11 THEN
    month% = m& + 2
  ELSE
    month% = m& - 10
  END IF
  day% = d&
  year% = y& + m& \ 11
  dat$ = mdytodate$(month%, day%, year%)
  juliantodate$ = dat$

END FUNCTION

FUNCTION mdytodate$ (month%, day%, year%)
'       Converts m,d,y to date string in format MM-DD-YYYY.

'       Program modified from one found in:
'              "The Microsoft QuickBASIC Programmer's Toolbox"
'               by John Clark Craig, Microsoft press, 1988

  yr$ = RIGHT$("000" + MID$(STR$(year%), 2), 4)
  mon$ = RIGHT$("0" + MID$(STR$(month%), 2), 2)
  da$ = RIGHT$("0" + MID$(STR$(day%), 2), 2)
  mdy$ = mon$ + "-" + da$ + "-" + yr$
  mdytodate$ = mdy$
END FUNCTION

DEFINT A-Z
FUNCTION ValidDate% (dat$)
'       Returns true if valid date dat$, otherwise returns false.

'       Program modified from one found in:
'              "The Microsoft QuickBASIC Programmer's Toolbox"
'               by John Clark Craig, Microsoft press, 1988
  CONST FALSE = 0
  CONST TRUE = NOT FALSE
  julian& = DateToJulian&(dat$)
  convert$ = juliantodate$(julian&)
  IF dat$ = convert$ THEN
    ValidDate% = TRUE
  ELSE
```

DATES.BAS (cont'd)

```
    ValidDate% - FALSE
  END IF

END FUNCTION
```

FSF600.BAS

```
DECLARE SUB GetUCResponse (ch$, filter$)
DECLARE SUB GetYNResponse (N$)
DECLARE SUB PrintDumpBuffer (a$, B$)
DECLARE SUB splitem (a$, B$, APRINT$, txtwidth%)
DECLARE SUB BottomOfPageCheck ()
DECLARE SUB ActualPrint (ASTRING$, TabFlag%)
DECLARE SUB PrintWordWrap (a$, B$)
DECLARE SUB TopMarginPrint (TOPMARGIN%)
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB UnPackArray (PackString$, thearray%())
DECLARE SUB DrawWindow (ulr%, ulc%, numlines%, length%, frametyp%,
       thecolor%)
DECLARE FUNCTION Exists% (FIL$)
DECLARE SUB ClearBuffer ()
DECLARE SUB InitPHRASE (PHRASE$())
DECLARE SUB MenuEntryPage (NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB inversetext (row%, col%, theresponse$)
DECLARE SUB versetext (row%, col%, theresponse$)
DECLARE SUB TextPause ()
DECLARE SUB SetColor (thecolor%)
DECLARE SUB CenterPrint (row%, CenterString$)
DECLARE SUB GetSF600Case (RealFileName$, heading$, FORGND%, BACGND%,
       RECORD%)
DECLARE SUB GetSetupStuff (LEFTMAR1%, LEFTMAR2%, TOP1%, TOP2%, BOP1%,
       BOP2%, LINWIDTH%)
DECLARE SUB ReplaceIt (mainstring$, oldstring$, newstring$)
DECLARE SUB SF600Help ()
DECLARE SUB KarenWindow ()
DEFINT A-Z
DECLARE SUB DrawSFBox (heading$, PAGE%, maxpages%)
DECLARE SUB NewInverseRoutine (NEWROW%, NEWCOL%, RCASE$())
DECLARE SUB OldInverseRoutine (OLDROW%, OLDCOL%, RCASE$())
DECLARE SUB computeStop (PAGE%, maxpages%, N%, NSTOP%)
DECLARE SUB GetKey (N$)
REM $DYNAMIC
'________________________________________________________________
' $INCLUDE: 'include.bas'

REM $STATIC
SUB ActualPrint (ASTRING$, TabFlag%)
'  This routine prints ASTRING$ using tabbing as defined by TabFlag.

'  Shared variables
```

```
  SHARED TABNUM, LMARGIN, OPUT, CLICK, MAXCLICK

  ' check for bottom of page and act upon it.
    CALL BottomOfPageCheck
  'divvy up job by output device
  SELECT CASE OPUT
    CASE 1   'screen
      SELECT CASE TabFlag
CASE 0         'CR with tabs
  LOCATE CLICK, TABNUM + LMARGIN
  CLICK = CLICK + 1

CASE 1         'CR no tabs
  LOCATE CLICK, LMARGIN + 1
  CLICK = CLICK + 1

CASE ELSE      ' no CR, no tabs - (TabFlag = 2)
  LOCATE CLICK, 2
  ' Ensure that ASTRING$ fits on screen OK, since no wordwrapping.
  ' This is mainly for the read in stored plan routine.  The string
  ' width may be OK for a printer, but too wide here.
  IF LEN(ASTRING$) > 78 THEN
    ASTRING$ = LEFT$(ASTRING$, 77) + CHR$(16)
  END IF

      END SELECT


      ' Use the ; for all screen printing so that the CR won't
      ' mess up the frame.
      PRINT ASTRING$;

    CASE 2   'printer
      SELECT CASE TabFlag
CASE 0         ' CR with tabs
  PRINT #2, TAB(TABNUM + LMARGIN); ASTRING$
  PRINT #2,
  CLICK = CLICK + 2

CASE 1         ' CR no tabs
  PRINT #2, ASTRING$
  PRINT #2,
  CLICK = CLICK + 2

CASE ELSE      ' no CR, no tabs - (TabFlag = 2)
  PRINT #2, ASTRING$;

      END SELECT
```

```
     CASE 3   'file
       SELECT CASE TabFlag
 CASE 0         ' CR with tabs
   PRINT #2, TAB(TABNUM + LMARGIN); ASTRING$
   CLICK = CLICK + 1

 CASE 1         ' CR no tabs
   PRINT #2, ASTRING$
   CLICK = CLICK + 1

 CASE ELSE      ' no CR, no tabs - (TabFlag = 2)
   PRINT #2, ASTRING$;

       END SELECT

     CASE ELSE
       'should never get here.
       BEEP

   END SELECT

END SUB

SUB BottomOfPageCheck
'  This routine checks for the bottom of the page.  If there,
       appropriate
' action is taken.

' Shared variables
  SHARED CLICK, MAXCLICK, OPUT, PAGE
  SHARED LMARGIN, TOPMARGIN, THEDATE$, THEBOAT$, TIM$
  SHARED LEFTMAR1, TOP1, BOP1, LEFTMAR2, TOP2, BOP2

  ' Check for end of page.  If not there, exit
  IF CLICK < MAXCLICK THEN
    EXIT SUB
  END IF

  SELECT CASE OPUT
    CASE 1      ' screen
      'Bottom of screen directions.
      CALL ClearBuffer
      CALL TextPause
      CLS
      'Draw frame if scrn output
      CALL SetColor(framecolor)
      CALL frame(1, 1, 23, 78, 3)
      CALL SetColor(ForeColor)
      CLICK = 2
      LOCATE CLICK, 2, 0

    CASE 2      ' printer
      ' How about framing this?
      CLS
      BEEP
      PRINT "This page is full."
      PRINT "Change the SF-600 in the printer."
      PRINT
      LOCATE , , 0
      COLOR backcolor, ForeColor
      PRINT "Press the ENTER/RETURN key when ready. "
      COLOR ForeColor, backcolor
      LINE INPUT N$
      IF PAGE = 1 THEN PAGE = 2 ELSE PAGE = 1
      IF PAGE = 1 THEN
LMARGIN = LEFTMAR1
TOPMARGIN = TOP1
MAXCLICK = BOP1
      ELSE
LMARGIN = LEFTMAR2
TOPMARGIN = TOP2
MAXCLICK = BOP2
      END IF
      ' Print top margin
      CALL TopMarginPrint(TOPMARGIN)
      ' Print SF600 heading.
      PRINT #2, SPACE$(LMARGIN) + THEDATE$ + SPACE$(3) + "(CONT'D)" +
       SPACE$(3) + THEBOAT$
      PRINT #2,
      PRINT #2, SPACE$(LMARGIN) + "   " + TIM$;
      CLICK = 3

    CASE 3      ' file
      CLICK = 1

    CASE ELSE
      ' Should never get here.
      BEEP
  END SELECT
END SUB

SUB ClearBuffer
'       This routine clears the text input buffer
DO
a$ = INKEY$
LOOP UNTIL a$ = ""
END SUB
```

```
REM $DYNAMIC
SUB computeStop (PAGE, maxpages, N%, NSTOP%)
  IF PAGE < maxpages THEN
    NSTOP% = 60
  ELSE
    NSTOP% = N% - ((maxpages - 1) * 60)
  END IF
END SUB

SUB DrawSFBox (heading$, PAGE, maxpages)

'       This routine draws the box used for the SF600 generator.

     CLS
     a$ = heading$
     a$ = a$ + "  Page " + STR$(PAGE) + " of " + STR$(maxpages)
     CALL SetColor(headingcolor)
     CALL CenterPrint(1, a$)
     CALL SetColor(ForeColor)
     CALL SetColor(framecolor)
     LOCATE 2, 1, 0
     PRINT CHR$(201);
     REM  LUC
     LOCATE 23, 1, 0
     PRINT CHR$(200);
     REM  LLC
     FOR I = 2 TO 78
FOR J = 0 TO 1
 LOCATE 2 + (J * 21), I, 0
 IF I <> 27 AND I <> 53 THEN
   PRINT CHR$(205);
   GOTO 1100
  END IF
 IF J = 0 THEN PRINT CHR$(209);  ELSE PRINT CHR$(207);
1100   NEXT J
     NEXT I
     LOCATE 2, 79, 0
     PRINT CHR$(187);
     REM  RUC
     LOCATE 23, 79, 0
     PRINT CHR$(188);
     REM  RLC
     FOR I = 3 TO 22
       FOR J = 0 TO 1
 LOCATE I, 1 + (J * 78), 0
 PRINT CHR$(186);
 LOCATE I, 27 + (J * 26), 0
 PRINT CHR$(179);
       NEXT J
     NEXT I
     CALL SetColor(ForeColor)

     REM      Print Options
     CALL SetColor(infocolor)
     LOCATE 24, 1, 0
     PRINT "Arrow keys move the cursor, PgUp, PgDn to change the page.
       Push Enter";
     LOCATE 25, 1, 0
     PRINT "to select the desired response or '?' for more information.
       ESC to exit.";
     CALL SetColor(ForeColor)
END SUB

REM $STATIC
SUB DrawWindow (ulr, ulc, numlines, length, frametyp, thecolor)
'       Draws frame about coordinates given which form the corner of
'     the frame.  Numlines and length do not include frame itself.
'       Also, can have several types of frames.
'     1 = single frame, 2 = double frame; 3-5 = block frames.

  CALL SetColor(thecolor)
  CALL frame(ulr, ulc, numlines, length, frametyp)
  CALL SetColor(ForeColor)
END SUB

REM $DYNAMIC
SUB GetSetupStuff (LEFTMAR1%, LEFTMAR2%, TOP1%, TOP2%, BOP1%, BOP2%,
       LINWIDTH%)
'       This routine checks for existence of SETUP.DAT.  If found, it
'       retrieves the margins and page lengths.  If not found, it uses
'       default values.
3000 OPEN "R", #1, "SETUP.DAT", 1
     N% = LOF(1)
     CLOSE #1
     IF N% = 0 THEN
       KILL "SETUP.DAT"
 '  LEFTMAR1 = left margin of the front of the SF600.
 '  LEFTMAR2 = left margin of the back of the SF600.
 '  TOP1 = Top margin of the front of the SF600.
 '  TOP2 = Top margin of the back of the SF600.
 '  BOP1 = Bottom margin of the front of the SF600.
 '  BOP2 = Bottom margin of the back of the SF600.
       LEFTMAR1 = 0
       LEFTMAR2 = 0
       TOP1 = 0
       TOP2 = 0
       BOP1 = 44
       BOP2 = 56
       LINWIDTH - 65
       CLS
       LOCATE 10, 10
       PRINT "SETUP.DAT file is not found.  Program will use system"
       LOCATE 11, 10
       PRINT " default values."
       CALL TextPause
     ELSE
       OPEN "SETUP.DAT" FOR INPUT AS #1
       IF NOT EOF(1) THEN
 INPUT #1, LEFTMAR1
 INPUT #1, LEFTMAR2
 INPUT #1, TOP1
 INPUT #1, TOP2
 INPUT #1, BOP1
 INPUT #1, BOP2
 INPUT #1, LINWIDTH
       END IF
3080   CLOSE #1
     END IF


END SUB

SUB GetSF600Case (RealFileName$, heading$, FORGND, BACGND, RECORD)
'This routine displays all cases in the SF600 window and returns the
' selected case as RECORD.

    REM $DYNAMIC
    REDIM RCASE$(60)

     REM READ REAL.DAT ROUTINE
     OPEN "R", #1, RealFileName$, 128
     FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$
     N% - LOF(1) / 128
     IF N% - 0 THEN
       CLS
       LOCATE 10, 10
       PRINT "No real cases stored."
       CALL TextPause
       RECORD - 0
       CLOSE #1
       ERASE RCASE$
       EXIT SUB
     END IF

     PAGE - 1
     maxpages = 1 + INT(N% / 60)
     CALL computeStop(PAGE, maxpages, N%, NSTOP%)

1220 REM box routine
     CALL DrawSFBox(heading$, PAGE, maxpages)
     row = 0
     COLNUM = 0
     COLFLAG = 0
      REM show a page of cases ( <= 60 cases per page)
     FOR I = 1 TO NSTOP%
       GET #1, I + ((PAGE - 1) * 60)
       row = row + 1
       IF row > 20 THEN
  row = 1
  COLNUM = COLNUM + 1
END IF
       NEWSSN$ = MID$(LSSN$, 1, 3) + MID$(LSSN$, 5, 2) + MID$(LSSN$, 8,
       4)
REM REMOVE -'S
       NEWDATE$ = LEFT$(LDAT$, 6) + RIGHT$(LDAT$, 2)
REM GET RID OF 19 IN 1985
       RCASE$(I) = NEWSSN$ + " " + NEWDATE$ + " " + LTIM$ + " "
       LOCATE 2 + row, 2 + (COLNUM * 26), 0
PRINT RCASE$(I);
     NEXT I
     LASTROW = row
     LASTCOL = COLNUM
     NUMRESP = N%
     OLDROW = 1
      NEWROW = 1
      OLDCOL = 0
      NEWCOL = 0
     CALL NewInverseRoutine(NEWROW, NEWCOL, RCASE$())

     REM        Answer Routine
1410 DO
       CALL GetKey(N$)
     LOOP UNTIL (LEN(N$) = 1 OR LEN(N$) = 2)

     IF LEN(N$) = 1 THEN
       SELECT CASE N$

 CASE "?"
   CALL SF600Help

 CASE CHR$(13)
   RECORD = NEWROW + 20 * NEWCOL
   RECORD = RECORD + ((PAGE - 1) * 60)
   CLOSE #1
  ERASE RCASE$
  EXIT SUB

CASE CHR$(27)
  CLS
  RECORD = 0
  CLOSE #1
  ERASE RCASE$
  EXIT SUB

CASE ELSE
  BEEP
  GOTO 1410

      END SELECT
    ELSE
      N$ = MID$(N$, 2, 1)
      SELECT CASE N$
CASE CHR$(80)
  GOSUB 1590

CASE CHR$(72)
  GOSUB 1640

CASE CHR$(77)
  GOSUB 1730

CASE CHR$(75)
  GOSUB 1690

CASE CHR$(81)    '       pagedown
  PAGE = PAGE + 1
  IF PAGE > maxpages THEN
    BEEP
    PAGE = maxpages
  END IF
  CALL computeStop(PAGE, maxpages, N%, NSTOP%)
  GOTO 1220

CASE CHR$(73)    '       pageup
  PAGE = PAGE - 1
  IF PAGE < 1 THEN
    BEEP
    PAGE = 1
  END IF
  CALL computeStop(PAGE, maxpages, N%, NSTOP%)
  GOTO 1220

CASE ELSE
```

```
   SOUND 100, 4

       END SELECT
     END IF
     GOTO 1410

     REM  DOWN ARROW
1590 NEWROW = NEWROW + 1
     IF NEWROW > 20 THEN
NEWCOL = NEWCOL + 1
NEWROW = 1
IF NEWCOL > LASTCOL THEN NEWCOL = 0
     END IF
     IF NEWROW > LASTROW AND NEWCOL = LASTCOL THEN
NEWROW = 1
NEWCOL = 0
      END IF
     GOSUB 1760
     RETURN

     REM  UP ARROW
1640 NEWROW = NEWROW - 1
     IF NEWROW < 1 THEN
NEWCOL = NEWCOL - 1
NEWROW = 20
IF NEWCOL < 0 THEN NEWCOL = LASTCOL
      END IF
     IF NEWROW > LASTROW AND NEWCOL = LASTCOL THEN NEWROW = LASTROW
     GOSUB 1760
     RETURN

     REM  LEFT ARROW
1690 NEWCOL = NEWCOL - 1
     IF NEWCOL < 0 THEN
NEWCOL = LASTCOL
IF NEWROW > LASTROW THEN NEWROW = LASTROW
      END IF
     GOSUB 1760
     RETURN

     REM RIGHT ARROW
1730 NEWCOL = NEWCOL + 1
     IF NEWCOL > LASTCOL THEN NEWCOL = 0
     IF NEWROW > LASTROW AND NEWCOL = LASTCOL THEN NEWROW = LASTROW
     GOSUB 1760
     RETURN


1760 CALL OldInverseRoutine(OLDROW, OLDCOL, RCASE$())
```

```
     CALL NewInverseRoutine(NEWROW, NEWCOL, RCASE$())
     OLDROW = NEWROW
     OLDCOL = NEWCOL
     RETURN

END SUB

REM $STATIC
SUB GetYNResponse (N$)
' returns either Y or N for sf600 program

11900 COLOR questionColor + 16, backcolor
      PRINT CHR$(177);
      CALL GetUCResponse(N$, "YN")
      LOCATE CSRLIN, POS(0) - 1, 0
      COLOR ForeColor, backcolor
      PRINT " ";

END SUB

REM $DYNAMIC
SUB KarenWindow
'This routine draws the lines and boxes that Karen designed for the CPDX
' program.

      CLS
      CALL SetColor(framecolor)
      CALL frame(1, 2, 9, 73, 2)
      CALL SetColor(ForeColor)

END SUB

SUB NewInverseRoutine (NEWROW, NEWCOL, RCASE$())
'       This routine inverses the cursor on the case selection page.

'50010 REM NEW INVERSE ROUTINE
      NROW = 2 + NEWROW
      NCOL = 26 * NEWCOL + 2
      rcaseptr = NEWROW + 20 * NEWCOL
      CALL inversetext(NROW, NCOL, RCASE$(rcaseptr))

END SUB

SUB OldInverseRoutine (OLDROW, OLDCOL, RCASE$())
'       This routine normalizes the inversed cursor on the case page.
'50070 REM OLD IN-INVERSE ROUTINE
      NROW = 2 + OLDROW
      NCOL = 26 * OLDCOL + 2
      rcaseptr = OLDROW + 20 * OLDCOL
      CALL versetext(NROW, NCOL, RCASE$(rcaseptr))

END SUB

REM $STATIC
SUB PrintDumpBuffer (a$, B$)
' This routine prints all of B$, then resets a$ and B$ to NULL
' originally  a$="":gosub 35090

' Shared variables
  SHARED TEXTWIDTH

  a$ = ""
  ' This IF statement ensures that B$ would not be printed twice if it
  '  was the exact length of TextWidth.
  IF B$ <> "" THEN
    CALL PrintWordWrap(a$, B$)
  END IF
  CALL ActualPrint(B$, 0)
  B$ = ""
END SUB

SUB PrintWordWrap (a$, B$)
' This routine word wraps lines and prints them by calling ActualPrint.

' Shared variables
  SHARED TEXTWIDTH

  DO
    CALL splitem(a$, B$, APRINT$, TEXTWIDTH)
    IF APRINT$ <> "" THEN
      CALL ActualPrint(APRINT$, 0)
    END IF
  LOOP UNTIL APRINT$ = ""
END SUB

REM $DYNAMIC
SUB ReplaceIt (mainstring$, oldstring$, newstring$) STATIC
  tempstring$ = mainstring$
  ptr% = INSTR(tempstring$, oldstring$)
  IF ptr% <> 0 THEN
    firstpart$ = LEFT$(tempstring$, ptr% - 1)
    lastpos = LEN(tempstring$) - ptr% - LEN(oldstring$) + 1
    lastpart$ = RIGHT$(tempstring$, lastpos)
    tempstring$ = firstpart$ + newstring$ + lastpart$
    mainstring$ = tempstring$
  END IF
END SUB
```

```
'--------------------------------------------------------------------
SUB SF600 (BOAT1$, BOAT2$, HMNAM$, HMSSN$)
  REM    PROG  ver 1.03
  REM  SF-600 GENERATION FOR ABDX PROGRAM
  DEFINT A-Z
  REM  Copywrite (C) 1985,1986,1987 Navy Submarine Medical Research
       Laboratory
      REM    author: David G. Southerland, MD
  REM 1.01- Corrected bug in canned PLAN routine.
  REM 1.02- 3/12/86 -Added capability to change LINE WIDTH from
       SETUP.DAT.
  REM 1.03- 2/2/87 - Combined diagnostic, training,
  REM and this program into a single program.  Removed refs to LOOP% and
       changed CASE% to THECASE%.
      REM 2.00 - addition of female pronouns and female questions to
       medical record entry form.

'            Dynamic array allocation.
      REM $DYNAMIC
'            THECASE%() - array containing response data.
'            PHRASE$() - array of strings associated with corresponding
       elements of THECASE%().
'            MONTH$() - array containing the months.
'            RCASE$() - array of displayed patients with SSN, date and
       time.
'            CASEPTR() - array containing all chosen responses in a
       question that allows multiple responses.

  ' Shared Variables
  SHARED TABNUM, CLICK, MAXCLICK, OPUT, PAGE, TEXTWIDTH
  SHARED LMARGIN, TOPMARGIN, THEDATE$, THEBOAT$, TIM$
  SHARED LEFTMAR1, TOP1, BOP1, LEFTMAR2, TOP2, BOP2


    REDIM THECASE%(200), PHRASE$(200)
    REDIM MONTH(12) AS STRING * 3
    REDIM RCASE$(60)
    REDIM CASEPTR(15)
    REDIM Choices$(10, 14)

    heading$ = "ABDX Program SF-600 Generator Version 2.0"
    SCREEN 0, 0, 0
    KEY OFF
    SCREEN 0, 1, 0

'         This routine reads the text phrases for PHRASE$(), for
'         use in FSF600.BAS.
'            Phrase ( PHRASE$() ) data.
```

```
      FILNAM$ = "phrase.dat"
      IF NOT Exists%(FILNAM$) THEN
   CLS
LOCATE 10, 10
PRINT "File "; FILNAM$; " not found.  Cannot continue without it."
SCREEN 0, 1, 0, 0
   END
      END IF
      filenum = FREEFILE
      OPEN FILNAM$ FOR INPUT AS filenum
      FOR I = 11 TO 174
LINE INPUT #filenum, a$
'decrypt
CALL decipher(a$)
PHRASE$(I) = RIGHT$(a$, LEN(a$) - 3)
      NEXT I
      CLOSE filenum

    BP$ = ""
    malesex = 1
    femalesex = 0
    TABNUM = 13
    CLICK = 1

    'show some color
    sfquestcolor = questionColor

    '  Maximum number of lines printed on the display screen.
    MAXCLICK = 23
    DUM = 0
    SKIP = 1
    RealFileName$ = "REAL.DAT"
    CALL GetSF600Case(RealFileName$, heading$, headingcolor, backcolor,
       RECORD)
    IF RECORD = 0 THEN
      ERASE THECASE%, PHRASE$, MONTH, RCASE$, CASEPTR, Choices$
      EXIT SUB
    END IF

    OPEN "R", #1, RealFileName$, 128
    FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$
    GET #1, RECORD
    SSN$ = LSSN$
    AGE$ = LAGE$
    OTHER$ = LOTH$
    TIM$ = LTIM$
    DAT$ = LDAT$
```

```
SIM$ = LSIM$
LLVAR$ = LVAR$
HMDX = CVI(LHMD$)
CLS
CLOSE

'  Unpack data and place it into THECASE%().
CALL UnPackArray(LLVAR$, THECASE%())

 IF THECASE%(2) = 1 THEN
   sex = femalesex
   UpPronoun$ = "She"
   LowPronoun$ = "she"
   sex$ = "female"
   UpPosPronoun$ = "Her"
 ELSE
   sex = malesex
   UpPronoun$ = "He"
   LowPronoun$ = "he"
   sex$ = "male"
   UpPosPronoun$ = "His"
 END IF
 IF sex = femalesex THEN
   CALL ReplaceIt(PHRASE$(60), "he", "she")
   CALL ReplaceIt(PHRASE$(68), "his", "her")
   CALL ReplaceIt(PHRASE$(69), "his", "her")
   CALL ReplaceIt(PHRASE$(81), "he", "she")
   CALL ReplaceIt(PHRASE$(82), "he", "she")
 END IF

 CALL GetSetupStuff(LEFTMAR1%, LEFTMAR2%, TOP1%, TOP2%, BOP1%,
   BOP2%, LINWIDTH%)
 CLS


 ' Select output
 Choices$(1, 1) = "CONSOLE - SF600 printed on display screen"
 Choices$(1, 2) = "PRINTER - SF600 printed on printer       "
 Choices$(1, 3) = "FILE    - SF600 printed to a file        "
 NR% = 1
 resplength% = 3
 DATAHEADING$ = heading$
 menuheading$ = "Select Output"
 HELPFILE$ = "HSF00.DAT"
 CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
   menuheading$, Choices$(), HELPFILE$)
 OPUT = NR%

 SELECT CASE OPUT
```

```
      CASE 1          '  Output goes to the display screen.
a$ = "SCRN:"
LMARGIN = 1

      CASE 2          '  Output goes to the printer.
SKIP = 2
a$ = "LPT1:"
GOSUB 3090

      CASE 3         ' Output goes to a file.
  CLS
LOCATE 9, 5
PRINT "Enter the file name for output : ";
LINE INPUT a$

      CASE ELSE
  CLS
ERASE THECASE%, PHRASE$, MONTH, RCASE$, CASEPTR, Choices$
EXIT SUB

    END SELECT

2210 IF OPUT <> 1 THEN
       OPEN "O", #2, a$
       IF OPUT = 2 THEN WIDTH #2, 255
     END IF
     GOTO 10000

3090 CLS
     PRINT "Are you starting on the first page of the SF-600? (Y/N)";
     CALL GetYNResponse(N$)
     IF N$ = "N" THEN PAGE = 2 ELSE PAGE = 1
3150 IF PAGE = 1 THEN
       LMARGIN = LEFTMAR1
       TOPMARGIN = TOP1
       MAXCLICK = BOP1
     ELSE
3200   LMARGIN = LEFTMAR2
       TOPMARGIN = TOP2
       MAXCLICK = BOP2
     END IF
3230 RETURN

10000 REM DATA CHECK
      CLS
      CALL DrawWindow(1, 2, 8, 73, 2, framecolor)
      LOCATE 3, 9: PRINT "Currently, the system will print range values
       for Vital Signs"
      LOCATE 4, 6: PRINT "and WBC results.  To replace the range values
       with specific"
      LOCATE 5, 6: PRINT "findings, enter the new value at the prompt.
       To leave the values"
      LOCATE 6, 6: PRINT "as shown, press the Enter/Return key at each
       prompt.  If you would"
      LOCATE 7, 6: PRINT "like the category not to be listed on the SF-
       600, then enter the"
      LOCATE 8, 6: PRINT "letter 'X' followed by the Enter/Return key."
      BLANK$ = SPACE$(18)
      CHKSTART = 89
      CHKSTOP = 92
      row = 12
      CATEGORY$ = "Temp  - "
      REM TEMP
      GOSUB 34000
      GOSUB 10100
      GOTO 10250
10100 LOCATE row, 1
      PRINT CATEGORY$;
      IF NUMCOUNT = 0 THEN 10130
      LOCATE row, 8
      PRINT PHRASE$(CASEPTR(1));
10130 LOCATE row, 40
      PRINT "New " + CATEGORY$;
      COLOR backcolor, ForeColor
      PRINT SPACE$(28)
      LOCATE row, 52, 1
      LINE INPUT I$
      COLOR ForeColor, backcolor

'             Clears question of any positive response.
      IF I$ = "X" OR I$ = "x" THEN
FOR I = CHKSTART TO CHKSTOP
  THECASE%(I) = 0
NEXT I
I$ = ""
GOTO 10210
      END IF
      IF I$ = "" THEN 10220
      IF I$ <> "" THEN
I$ = " " + I$
IF NUMCOUNT = 0 THEN
  PHRASE$(CHKSTART) = I$
  THECASE%(CHKSTART) = 1
  ELSE
  PHRASE$(CASEPTR(1)) = I$
END IF
      END IF
```

```
10210 LOCATE row, 8, 0
      PRINT MID$(I$ + SPACE$(30), 1, 29);
10220 LOCATE , , 0
      RETURN
10250 CHKSTART = 93
      CHKSTOP = 95
      row = 14
      CATEGORY$ = "Pulse - "
      REM PULSE
      GOSUB 34000
      GOSUB 10100
      CHKSTART = 96
      CHKSTOP = 101
      row = 16
      CATEGORY$ = "BP    - "
      REM BP(SYS/DIAS)
      GOSUB 34000
      LOCATE row, 1, 0
      PRINT CATEGORY$;
      IF BP$ = "" THEN IF NUMCOUNT = 0 THEN 10330
      LOCATE row, 8, 0
      IF BP$ <> "" THEN
PRINT BP$;
      ELSE
PRINT PHRASE$(CASEPTR(1)); PHRASE$(CASEPTR(2));
      END IF
10330 LOCATE row, 40
      PRINT "New " + CATEGORY$;
      COLOR backcolor, ForeColor
      PRINT SPACE$(28)
      LOCATE row, 52, 1
      LINE INPUT I$
      COLOR ForeColor, backcolor
      IF I$ = "X" OR I$ = "x" THEN
FOR I = CHKSTART TO CHKSTOP
  THECASE%(I) = 0
NEXT I
I$ = ""
BP$ = ""
GOTO 10405
      END IF
      IF I$ = "" THEN 10410
      BP$ = " " + I$
10405 LOCATE row, 8
      PRINT MID$(BP$ + SPACE$(30), 1, 29);
10410 CHKSTART = 110
      CHKSTOP = 114
      row = 18
      CATEGORY$ = "WBC   - "
      REM WBC
      GOSUB 34000
      GOSUB 10100
10440 LOCATE 21, 1, 0
      PRINT "Are all these OK ? (Y/N) ";
      COLOR ForeColor, backcolor
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN 10000

      CHKSTART = 42
      CHKSTOP = 47
      ITEM = 46
      REM AGG FACTORS
      IF THECASE%(ITEM) = 0 THEN 11100
      CATEGORY$ = "OTHER for Aggravating Factors"
      B$ = "  The OTHER aggravating factor is "
      a$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
      PHRASE$(ITEM) = a$
11100 CHKSTART = 55
      CHKSTOP = 60
      ITEM = 59
      REM REL FACTORS
      IF THECASE%(ITEM) = 0 THEN 11200
      CATEGORY$ = "OTHER for Relieving Factors"
      B$ = "  The OTHER Relieving factor is "
      a$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
      PHRASE$(ITEM) = a$
11200 CHKSTART = 83
      CHKSTOP = 84
      ITEM = 83
      REM PREV ABDOMINAL SURGERY
      IF THECASE%(ITEM) = 0 THEN 11300
      CATEGORY$ = "positive Previous Abdominal Surgery"
      B$ = "  " + UpPronoun$ + " has had "
      a$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) <> "." THEN a$ = a$ + "."
      PHRASE$(ITEM) = a$
11300 CHKSTART = 85
```

```
      CHKSTOP = 86
      ITEM = 85
      REM PREV ILLNESS
      IF THECASE%(ITEM) = 0 THEN 11400
      CATEGORY$ = "positive Previous Illness"
      B$ = "  " + UpPronoun$ + " has had "
      a$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) <> "." THEN a$ = a$ + "."
      PHRASE$(ITEM) = a$
11400 CHKSTART = 87
      CHKSTOP = 88
      ITEM = 87
      REM MEDICATION
      IF THECASE%(ITEM) = 1 THEN
CATEGORY$ = "positive Medication"
B$ = "  " + UpPronoun$ + " is currently taking "
a$ = ""
GOSUB 34000
GOSUB 12000
a$ = " " + a$
IF RIGHT$(a$, 1) <> "." THEN a$ = a$ + "."
PHRASE$(ITEM) = a$
      END IF

      ' Beginning of Female specific question, more info gathering
       routines
      'LMP
      CHKSTART = 164
      CHKSTOP = 165
      ITEM = 165
      IF THECASE%(ITEM) = 1 THEN
CATEGORY$ = "late/overdue"
a$ = " Her period "
B$ = ""
GOSUB 34000
GOSUB 12000
a$ = " " + a$
IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
PHRASE$(ITEM) = a$
      END IF

      'Vaginal Discharge
      CHKSTART = 166
      CHKSTOP = 167
      ITEM = 166
      IF THECASE%(ITEM) = 1 THEN
```

```
CATEGORY$ = "recent vaginal discharge"
a$ = " She has had a recent vaginal discharge which "
B$ = ""
GOSUB 34000
GOSUB 12000
a$ = " " + a$
IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
PHRASE$(ITEM) = a$
      END IF

11500 CHKSTART = 118
      CHKSTOP = 119
      ITEM = 118
      REM SCARS
      IF THECASE%(ITEM) = 0 THEN 11600
      CATEGORY$ = "positive Abdominal Surgical Scars"
      a$ = " The patient has "
      B$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) <> "." THEN a$ = a$ + "."
      PHRASE$(ITEM) = a$
11600 CHKSTART = 129
      CHKSTOP = 130
      ITEM = 129
      REM MASSES
      IF THECASE%(ITEM) = 0 THEN 11700
      CATEGORY$ = "positive Abdominal Masses"
      a$ = " There is an abdominal mass "
      B$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
      PHRASE$(ITEM) = a$
11700 CHKSTART = 148
      CHKSTOP = 152
      ITEM = 149
      REM RECTAL
      FOR I = CHKSTART TO CHKSTOP
IF THECASE%(I) = 1 THEN 11705
      NEXT I
      GOTO 11950
11705 IF THECASE%(ITEM) = 0 THEN 11800
      CATEGORY$ = "positive Rectal Mass"
      B$ = "  The rectal examination reveals "
      a$ = ""
      GOSUB 34000
      GOSUB 12000
      a$ = " " + a$
      IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
      PHRASE$(ITEM) = a$
11800 CLS
      PRINT "Did you check the stool for occult blood. ? (Y/N) ";
      COLOR ForeColor, backcolor
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN 11950
      PRINT
      PRINT "Is the stool positive for occult blood ? (Y/N) ";
      CALL GetYNResponse(N$)
      IF N$ = "Y" THEN
HEME$ = "  The stool is positive for occult blood."
      ELSE
HEME$ = "  The stool is negative for occult blood."
      END IF


      ' Female specific physical question, more info gathering routines
      ' Vaginal exam - similar to rectal exam (the programming , that
       is)
11950 CHKSTART = 153
      CHKSTOP = 159
      ITEM = 158
      REM PELVIC MASSES
      IF THECASE%(ITEM) = 1 THEN
CATEGORY$ = "positive Pelvic Masses"
B$ = "  The pelvic examination reveals "
a$ = ""
GOSUB 34000
GOSUB 12000
a$ = " " + a$
IF RIGHT$(a$, 1) = "." THEN a$ = MID$(a$, 1, LEN(a$) - 1)
PHRASE$(ITEM) = a$
      END IF
      GOTO 13000


12000 AA$ = a$
      BB$ = B$
12005 a$ = AA$
      B$ = BB$
      CLS
      CALL KarenWindow
      LOCATE 2, 5
      PRINT "  You have selected ";
      COLOR backcolor, ForeColor
      PRINT CATEGORY$;
      COLOR ForeColor, backcolor
      PRINT ".";
      LOCATE 4, 5: PRINT "Complete the following sentence.  Be as brief
       as possible.  If"
      LOCATE 5, 5: PRINT "you desire the programmed default statement,
       then enter the ENTER/"
      LOCATE 6, 5: PRINT "RETURN key on the first line.  If you desire
       to delete the item, enter"
      LOCATE 7, 5: PRINT "the letter 'X' followed by the ENTER/RETURN
       key on the first line."
      LOCATE 8, 5: PRINT "After entering all the statements you desire,
       enter the ENTER/RETURN"
      LOCATE 9, 5: PRINT "key on a line by itself.  Note - Use only one
       sentence."
      CONTER = 1
      LOCATE 12, 1, 1
      COLOR sfquestcolor, backcolor
      PRINT CONTER; ">>";
      COLOR ForeColor, backcolor
      IF a$ = "" THEN PRINT B$;  ELSE PRINT a$;
12110 LINE INPUT I$
      IF I$ = "" THEN
IF CONTER > 1 THEN
  GOTO 12160
  ELSE
  a$ = MID$(PHRASE$(ITEM), 2, (LEN(PHRASE$(ITEM)) - 1))
  GOTO 12160
END IF
      END IF
      IF I$ = "X" OR I$ = "x" THEN
THECASE%(ITEM) = 0
GOTO 12160
      ELSE
THECASE%(ITEM) = 1
      END IF
      IF INSTR(" .?!", RIGHT$(I$, 1)) = 0 THEN I$ = I$ + " "
      a$ = a$ + I$
      CONTER = CONTER + 1
      COLOR sfquestcolor, backcolor
      PRINT CONTER; ">>";
      COLOR ForeColor, backcolor
      GOTO 12110
12160 LOCATE 22, 1, 0
      PRINT " Is the above correct ? (Y/N) ";
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN GOTO 12005

12220 IF RIGHT$(a$, 1) = " " THEN a$ = MID$(a$, 1, (LEN(a$) - 1))
      RETURN
```

```
13000 a$ = ""
      TITLENAM$ = "HISTORY"
      GOSUB 41000
      HXTXT$ = a$
      a$ = ""
      TITLENAM$ = "PHYSICAL EXAM"
      GOSUB 41000
      PETXT$ = a$

      SELECT CASE HMDX
CASE 1
  DX$ = "APPENDICITIS"
CASE 2
  DX$ = "NON-SPECIFIC ABDOMINAL PAIN"
CASE 3
  DX$ = "RENAL COLIC"
CASE 4
  DX$ = "PERFORATED DUODENUM"
CASE 5
  DX$ = "CHOLECYSTITIS"
CASE 6
  DX$ = "SMALL BOWEL OBSTRUCTION"
CASE 7
  DX$ = OTHER$
CASE 8
  DX$ = "APPENDICITIS"
CASE 9
  DX$ = "NON-SPECIFIC ABDOMINAL PAIN"
CASE 10
  DX$ = "PELVIC INFLAMMATORY DISEASE"
CASE 11
  DX$ = "URINARY TRACT INFECTION"
CASE 12
  DX$ = "OVARIAN CYST"
CASE 13
  DX$ = "ECTOPIC PREGNANCY"
CASE 14
  DX$ = "INCOMPLETE ABORTION"
CASE ELSE
  DX$ = OTHER$
      END SELECT

14080 CLS
      PRINT "Your original diagnosis was ";
      COLOR backcolor, ForeColor
      PRINT DX$;
      COLOR ForeColor, backcolor
      PRINT "."
      PRINT
      PRINT "Do you desire to change it ? (Y/N) ";
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN 14130
14090 LOCATE 15, 1
      PRINT "Enter your new diagnosis - ";
      COLOR backcolor, ForeColor
      PRINT SPACE$(51)
      LOCATE 15, 28, 1
      LINE INPUT D$
      COLOR ForeColor, backcolor
      IF D$ = "" THEN BEEP: GOTO 14080
      DX$ = D$
14130 LOCATE , , 0
15000 FILNAM$ = ""
      PLAN$ = ""
15010 CLS
      PRINT " You may type in the plan or if you have a routine plan"
      PRINT "stored as a file, you may use it."
      PRINT
      PRINT "Do you have a routine plan already on disk ? (Y/N) ";
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN 15080
      PRINT " Enter the name of the file containing the plan - ";
      LINE INPUT FILNAM$
      IF FILNAM$ = "" THEN 15010
      IF NOT Exists%(FILNAM$) THEN
   CLS
LOCATE 10, 10
PRINT "File "; FILNAM$; " is not found."
CALL TextPause
GOTO 15000
      END IF
      GOTO 20000
15080 CLS
      CALL KarenWindow
      C$ = ""
      a$ = "Plan: 1.  "
      LOCATE 3, 5: PRINT "Enter as many lines as you desire for each
       plan number.  When you are"
      LOCATE 4, 5: PRINT "finished with a plan number, press the
       ENTER/RETURN key on a line by"
      LOCATE 5, 5: PRINT "itself and you will proceed to the next plan
       number.  When you are"
      LOCATE 6, 5: PRINT "finished with the plan, press the ENTER/RETURN
       key on the first line"
      LOCATE 7, 5: PRINT "of the next new plan number.  You can exit the
       plan only by pressing"
      LOCATE 8, 5: PRINT "the ENTER/RETURN key on the first line of a
       new plan number."
      CONTER = 1
      LOCATE 12, 1, 1
      COLOR sfquestcolor, backcolor
      PRINT CONTER; ">";
      COLOR ForeColor, backcolor
      PRINT STR$(CONTER); ".  ";
      FIRSTLIN = 0
15210 LINE INPUT I$
      IF I$ = "" THEN 15260
      IF RIGHT$(I$, 1) = " " THEN I$ = MID$(I$, 1, (LEN(I$) - 1))
      IF INSTR(".?!", RIGHT$(I$, 1)) = 0 THEN I$ = I$ + " " ELSE I$ = I$
       + "  "
      a$ = a$ + C$ + I$
      FIRSTLIN = 1
      B$ = ""
      C$ = ""
15240 COLOR sfquestcolor, backcolor
      PRINT CONTER; ">";
      COLOR ForeColor, backcolor
      PRINT B$;
      GOTO 15210
15260 IF FIRSTLIN = 0 THEN 15269
      CONTER = CONTER + 1
      a$ = a$ + CHR$(13)
      B$ = STR$(CONTER) + ".  "
      C$ = "      " + B$
      FIRSTLIN = 0
      GOTO 15240
15269 LOCATE 25, 1, 0
      PRINT " Is the above correct ? (Y/N) ";
      CALL GetYNResponse(N$)
      IF N$ = "N" THEN 15080

15320 IF a$ = "" THEN 15330
      IF RIGHT$(a$, 1) = " " THEN a$ = MID$(a$, 1, (LEN(a$) - 1))
      IF RIGHT$(a$, 1) = " " THEN a$ = MID$(a$, 1, (LEN(a$) - 1))
15330 PLAN$ = a$

      IF OPUT = 2 THEN
   CLS
PRINT "Make sure the printer is on and the SF-600 aligned."
PRINT " If not, then do so before answering the question below."
 PRINT
 PRINT
 PRINT
PRINT "  Is the printer now on and the paper in place? (Y/N)";
CALL GetYNResponse(N$)
```

```
' exit subprogram
IF N$ = "N" THEN 31010
      END IF

20000 CLS
      ' Print top margin (@ 42400)
      CALL TopMarginPrint(TOPMARGIN)

      MONTH(1) = "JAN": MONTH(2) = "FEB"
      MONTH(3) = "MAR": MONTH(4) = "APR"
      MONTH(5) = "MAY": MONTH(6) = "JUN"
      MONTH(7) = "JUL": MONTH(8) = "AUG"
      MONTH(9) = "SEP": MONTH(10) = "OCT"
      MONTH(11) = "NOV": MONTH(12) = "DEC"

      THEDATE$ = MID$(DAT$, 4, 2) + " " + MONTH(VAL(MID$(DAT$, 1, 2))) +
       " " + MID$(DAT$, 7, 4)
      THEBOAT$ = BOAT1$ + "  (" + BOAT2$ + ")"
      TBOAT = INT((LINWIDTH - LEN(THEBOAT$)) / 2)
      ' actual column width for text
      TEXTWIDTH = LINWIDTH - LMARGIN

      'Draw frame if scrn output
      IF OPUT = 1 THEN
CALL SetColor(framecolor)
CALL frame(1, 1, 23, 78, 3)
CALL SetColor(ForeColor)
CLICK = 2
      END IF

      ' Prints APRINT$ without added tabbing and without a CR.
      APRINT$ = SPACE$(LMARGIN) + THEDATE$
      CALL ActualPrint(APRINT$, 2)

      APRINT$ = SPACE$(TBOAT) + THEBOAT$
      ' Prints APRINT$ with added tabing and with a CR.
      CALL ActualPrint(APRINT$, 0)

      APRINT$ = SPACE$(LMARGIN) + "   " + TIM$
      CALL ActualPrint(APRINT$, 2)

      REM History

      a$ = "    This " + AGE$ + " year old " + sex$ + " presents with
       abdominal pain which began in"

      ' Print as much of A$ as you can on a per line basis.
      B$ = ""
      CALL PrintWordWrap(a$, B$)
      ' Pain at onset.
      CHKSTART = 11
      CHKSTOP = 23

      ' Check for multiple responses and load into CASEPTR().
      GOSUB 34000

      ' Join multiple responses into one phrase.
      GOSUB 34050

      ' Pain at present.
      CHKSTART = 24
      CHKSTOP = 36
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20150
      a$ = " and is now located in"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Type of pain.
20150 CHKSTART = 37
      CHKSTOP = 39
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20220
      a$ = " and is described as being"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = " in nature"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
20220 a$ = "."
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      '  Duration of pain.
      CHKSTART = 51
      CHKSTOP = 54
      PRIOR = 0
      GOSUB 34000
      a$ = "  The pain"
      IF NUMCOUNT = 0 THEN 20270
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      PRIOR = 1
      GOSUB 34050
      '  Severity of pain.
20270 CHKSTART = 40
      CHKSTOP = 41
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20305
      IF PRIOR = 0 THEN a$ = "  The pain is"
      IF PRIOR = 1 THEN a$ = " and is"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      PRIOR = PRIOR + 2
      GOSUB 34050

      '  Progress of pain.
20305 CHKSTART = 48
      CHKSTOP = 50
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20335
      IF PRIOR = 0 THEN
a$ = "  The pain"
GOTO 20330
      END IF
      a$ = " and it"
20330 ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      GOSUB 34050
20335 IF PRIOR > 0 THEN
a$ = "."
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF

      '  Aggravating factors.
      CHKSTART = 42
      CHKSTOP = 47
      GOSUB 34000
      FLG = 1
      a$ = "  By history,"
      IF NUMCOUNT = 0 THEN
FLG = 0
GOTO 20410
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      IF NUMCOUNT > 1 THEN a$ = " seem" ELSE a$ = " seems"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = " to make the pain worse"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = " and"

      '  Relieving factors.
20410 CHKSTART = 55
      CHKSTOP = 60
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
IF FLG = 0 THEN
  GOTO 20480
  ELSE
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
  GOTO 20470
END IF
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      IF NUMCOUNT > 1 THEN a$ = " seem" ELSE a$ = " seems"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      a$ = " to relieve the pain."
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      '  Nausea.
20470 CHKSTART = 61
      CHKSTOP = 62
20480 a$ = "  " + UpPronoun$ + " has experienced"
      GOSUB 34000
      FLG = 1
      IF NUMCOUNT = 0 THEN
FLG = 0
GOTO 20510
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = " and"

      '  Vomiting.
20510 CHKSTART = 63
      CHKSTOP = 64
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
IF FLG = 0 THEN
  GOTO 20540
  ELSE
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
  GOTO 20540
END IF
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Appetite
20540 CHKSTART = 70
      CHKSTOP = 71
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20580
      a$ = "  " + UpPosPronoun$ + " appetite has been"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Jaundice
20580 CHKSTART = 72
      CHKSTOP = 73
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20620
      GOSUB 34050

      '  Bowels.
20620 CHKSTART = 65
      CHKSTOP = 69
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20670
      a$ = "  The patient has had"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = "."
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      '  Urination
20670 CHKSTART = 74
      CHKSTOP = 78
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20710
      a$ = "  " + UpPronoun$ + " has also had"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = "."
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      '  Previous indigestion.
20710 CHKSTART = 79
      CHKSTOP = 80
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20740
      GOSUB 34050

      '  Similar pain.
20740 CHKSTART = 81
      CHKSTOP = 82
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20770
      GOSUB 34050

      '  Surgery.
20770 CHKSTART = 83
      CHKSTOP = 84
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20810
      a$ = "  " + UpPronoun$ + " has had"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Previous illness.
20810 CHKSTART = 85
      CHKSTOP = 86
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 20850
      a$ = "  Also, the patient has had"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Current medication.
20850 CHKSTART = 87
      CHKSTOP = 88
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  " + UpPronoun$ + " is currently taking"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
```

```
GOSUB 34050
      END IF


      ' Female Specific History Questions
      ' Periods
      CHKSTART = 160
      CHKSTOP = 163
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  Her periods"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
GOSUB 34050
      END IF

      'LMP
      CHKSTART = 164
      CHKSTOP = 165
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
GOSUB 34050
a$ = "."
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF

      'Vaginal Discharge
      CHKSTART = 166
      CHKSTOP = 167
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
GOSUB 34050
a$ = "."
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF

      'Pregnancy
      CHKSTART = 168
      CHKSTOP = 170
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  The patient states that pregnancy"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
GOSUB 34050
      END IF
      'Faint/Dizzy
      CHKSTART = 171
      CHKSTOP = 172
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  She"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
GOSUB 34050
a$ = " recently felt faint/dizzy."
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF

      'PID/VD history
      CHKSTART = 173
      CHKSTOP = 174
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  She has"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
GOSUB 34050
a$ = " prior history of pelvic inflammatory disease or venereal disease.
       "
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF



20900 IF HXTXT$ <> "" THEN
a$ = "  " + HXTXT$
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)

      '  Temp.
      CHKSTART = 89
      CHKSTOP = 92
      GOSUB 34000
      a$ = "Vital Signs:"
      IF NUMCOUNT = 0 THEN 20980
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = "  Temp  -"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = ""

      '  Pulse.
20980 CHKSTART = 93
      CHKSTOP = 95
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21050
      IF a$ = "" THEN
' GOSUB 35090
CALL PrintDumpBuffer(a$, B$)
a$ = "            "
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = "  Pulse -"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = ""

      '  BP (systolic).
21050 CHKSTART = 96
      CHKSTOP = 98
      IF BP$ = "" THEN 21070
      IF a$ = "" THEN
' GOSUB 35090
CALL PrintDumpBuffer(a$, B$)
a$ = "            "
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = "  BP    -" + BP$
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOTO 21110
21070 GOSUB 34000
      IF NUMCOUNT = 0 THEN 21110
      IF a$ = "" THEN
' GOSUB 35090
CALL PrintDumpBuffer(a$, B$)
a$ = "            "
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      a$ = "  BP    -"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  BP (diastolic).
      CHKSTART = 99
      CHKSTOP = 101
      GOSUB 34000
      GOSUB 34050

      '  WBC.
21110 CHKSTART = 110
      CHKSTOP = 114
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21170
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      CALL PrintDumpBuffer(a$, B$)
      a$ = "        Lab: WBC   -"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
21170 a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)

      '  Mood.
      CHKSTART = 102
      CHKSTOP = 104
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21230
      a$ = "    Physical examination reveals a patient whose mood is"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Color.
21230 CHKSTART = 105
      CHKSTOP = 109
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21290
      a$ = "  The patient's color is"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      a$ = "."
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)

      '  Inspection.
21290 CHKSTART = 115
      CHKSTOP = 117
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21330
      GOSUB 34050

      '  Scars.
21330 CHKSTART = 118
      CHKSTOP = 119
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21380
      GOSUB 34050

      '  Bowel Sounds.
21380 CHKSTART = 124
      CHKSTOP = 126
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21420
      GOSUB 34050

      '  Distention.
21420 CHKSTART = 127
      CHKSTOP = 128
      GOSUB 34000
      IF NUMCOUNT = 0 THEN 21460
      GOSUB 34050

      '  Vol guard.
21460 CHKSTART = 120
      CHKSTOP = 121
      PRIOR = 0
      GOSUB 34000
      IF NUMCOUNT = 0 THEN GOTO 21510
      GOSUB 34050
      PRIOR = 1

      '  Rigidity.
21510 CHKSTART = 122
      CHKSTOP = 123
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
IF PRIOR = 1 THEN
```

```
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
  GOTO 21560
  ELSE
  GOTO 21560
END IF
      END IF
      IF PRIOR = 0 THEN a$ = "  Involuntary" ELSE a$ = " and
       involuntary"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Tenderness.
21560 CHKSTART = 131
      CHKSTOP = 143
      PRIOR = 0
      GOSUB 34000
      IF NUMCOUNT = 0 THEN GOTO 21620
      a$ = "  Tenderness is noted in"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050
      PRIOR = 1

      '  Rebound.
21620 CHKSTART = 146
      CHKSTOP = 147
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
IF PRIOR = 1 THEN
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
  GOTO 21670
  ELSE
  GOTO 21670
END IF
      END IF
      IF PRIOR = 0 THEN a$ = "  Rebound" ELSE a$ = " and rebound"
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Abd mass.
21670 CHKSTART = 129
      CHKSTOP = 130
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
a$ = " "
      ELSE
GOSUB 34050
a$ = " and"
      END IF

      '  Murphy's sign.
21730 CHKSTART = 144
      CHKSTOP = 145
      GOSUB 34000
      IF NUMCOUNT = 0 THEN
IF a$ = " and" THEN
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
END IF
GOTO 21780
      END IF
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      GOSUB 34050

      '  Rectal.
21780 CHKSTART = 148
      CHKSTOP = 152
      GOSUB 34000
      IF NUMCOUNT > 0 THEN
a$ = "  The rectal exam reveals"
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
GOSUB 34050
a$ = "."
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
IF HEME$ <> "" THEN
  a$ = HEME$
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
END IF
      END IF

      ' Female Specific Physical Questions
      'Vaginal Examination
      ' A little messy since mass in in the middle of the responses.

      'First check for normal.  If so, use complete sentence.
      IF THECASE%(153) = 1 THEN
a$ = PHRASE$(153)
```

```
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      ELSE
' pelvic multiple responses - L, R, cx, and general tenderness, and
'   pelvic mass.
CHKSTART = 154
CHKSTOP = 158
GOSUB 34000
IF NUMCOUNT > 0 THEN
  a$ = "  The pelvic exam reveals"
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
  GOSUB 34050
  a$ = "."
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
END IF
IF THECASE%(159) = 1 THEN
  'Clots - last response of question
  a$ = PHRASE$(159)
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
END IF
      END IF




21840 IF PETXT$ <> "" THEN
a$ = "   " + PETXT$
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      END IF
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      a$ = "Impression: " + DX$
      ' GOSUB 35050
      B$ = ""
      CALL PrintWordWrap(a$, B$)

      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      IF PLAN$ <> "" THEN
```

```
a$ = ""
' GOSUB 35090
CALL PrintDumpBuffer(a$, B$)
a$ = ""
FOR I = 1 TO LEN(PLAN$)
  AA$ = MID$(PLAN$, I, 1)
  IF AA$ = CHR$(13) THEN
    'end of line,so print CR
    ' GOSUB 35090
            CALL PrintWordWrap(a$, B$)
            ' CALL PrintDumpBuffer(a$, B$)
    a$ = ""
    ' GOSUB 35090
    CALL PrintDumpBuffer(a$, B$)
  ELSE
    a$ = a$ + AA$
  END IF
NEXT I
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
a$ = ""
' GOSUB 35090
CALL PrintDumpBuffer(a$, B$)
GOTO 23000
      END IF
22110 IF FILNAM$ = "" THEN 23000

      '  Incorporate previously stored plan.
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      OPEN "I", #1, FILNAM$
      IF EOF(1) THEN
CLOSE #1
GOTO 23000
      END IF
      LINE INPUT #1, I$
      ' APRINT$ = SPACE$(TABNUM + LMARGIN) + "Plan: " + I$
      ' DUM = 0

      APRINT$ = "Plan: " + I$
      CALL ActualPrint(APRINT$, 0)

22150 IF EOF(1) THEN
CLOSE #1
GOTO 23000
      END IF
      LINE INPUT #1, I$
      IF I$ = "" THEN
```

```
' DUM = 0
APRINT$ = I$

CALL ActualPrint(APRINT$, 1)
GOTO 22150
      END IF
      APRINT$ = SPACE$(7) + I$
      ' GOSUB 42000
      CALL ActualPrint(APRINT$, 0)
      GOTO 22150
23000 REM FINISH UP
      FOR I = 1 TO 2
APRINT$ = ""
CALL ActualPrint(APRINT$, 1)
      NEXT I

      '  Print name, SSN for signature.
      NUM = TABNUM + INT((LINWIDTH - LEN(HMNAM$)) / 2)
      OLDTAB = TABNUM
      TABNUM = NUM
      APRINT$ = HMNAM$
      ' GOSUB 42000
      CALL ActualPrint(APRINT$, 0)
      APRINT$ = HMSSN$
      ' GOSUB 42000
      CALL ActualPrint(APRINT$, 0)
      TABNUM = OLDTAB
      a$ = ""
      ' GOSUB 35090
      CALL PrintDumpBuffer(a$, B$)
      IF OPUT = 1 THEN
CALL TextPause
      END IF
31010 CLOSE
      SCREEN 0, 1, 0, 0
      ERASE THECASE%, PHRASE$, MONTH, RCASE$, CASEPTR, Choices$
      EXIT SUB

      '  Check for multiple responses.
34000 NUMCOUNT = 0
      FOR I = 1 TO 15
CASEPTR(I) = 0
      NEXT I
      FOR I = CHKSTART TO CHKSTOP
IF THECASE%(I) = 1 THEN
  NUMCOUNT = NUMCOUNT + 1
  CASEPTR(NUMCOUNT) = I
END IF
      NEXT I
      RETURN

      '  Joins multiple responses into one string.
34050 IF NUMCOUNT <= 2 THEN
a$ = PHRASE$(CASEPTR(1))
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
IF NUMCOUNT = 2 THEN
  a$ = " and" + PHRASE$(CASEPTR(2))
  ' GOSUB 35090
  CALL PrintWordWrap(a$, B$)
END IF
RETURN
      END IF
      FOR I = 1 TO NUMCOUNT - 1
a$ = PHRASE$(CASEPTR(I)) + ","
' GOSUB 35090
CALL PrintWordWrap(a$, B$)
      NEXT I
      a$ = " and" + PHRASE$(CASEPTR(NUMCOUNT))
      ' GOSUB 35090
      CALL PrintWordWrap(a$, B$)
      RETURN


41000 AA$ = a$
41005 a$ = AA$
      CLS
      CALL KarenWindow
      LOCATE 2, 5: PRINT "  Here you may enter anything you feel is
       important to the "; TITLENAM$; "."
      LOCATE 3, 5: PRINT "Type the ENTER/RETURN key on the first line if
       you have nothing to add."
      LOCATE 4, 5: PRINT "Otherwise, enter as many lines as you wish.
       When you have finished,type"
      LOCATE 5, 5: PRINT "the ENTER/RETURN key on a separate line by
       itself."
      CONTER = 1
      LOCATE 12, 1, 1
      COLOR sfquestcolor, backcolor
      PRINT CONTER; ">>";
      COLOR ForeColor, backcolor
      'David, this was IF a$="" then print a$.  I think it should be
       <>"".
      IF a$ <> "" THEN PRINT a$;
41110 LINE INPUT I$
      IF I$ = "" THEN 41160
      IF RIGHT$(I$, 1) = " " THEN I$ = MID$(I$, 1, (LEN(I$) - 1))
      IF INSTR(".?!", RIGHT$(I$, 1)) = 0 THEN I$ = I$ + " " ELSE I$ = I$
       + "  "
      a$ = a$ + I$
      CONTER = CONTER + 1
      COLOR sfquestcolor, backcolor
      PRINT CONTER; ">>";
      COLOR ForeColor, backcolor
      GOTO 41110
41160 LOCATE 25, 1, 0
      PRINT " Is the above correct ? (Y/N) ";
      COLOR backcolor + 16, ForeColor
      PRINT CHR$(177);
      COLOR ForeColor, backcolor
      CALL GetKey(N$)
      IF N$ = "Y" THEN 41220
      IF N$ <> "N" THEN
  BEEP
GOTO 41160
      END IF
      GOTO 41005

      '  More of the print routine.
41220 IF a$ <> "" THEN
IF RIGHT$(a$, 1) = " " THEN a$ = MID$(a$, 1, (LEN(a$) - 1))
IF RIGHT$(a$, 1) = " " THEN a$ = MID$(a$, 1, (LEN(a$) - 1))
IF RIGHT$(a$, 1) <> "." THEN a$ = a$ + "."
      END IF
      RETURN

END SUB

SUB SF600Help
'       This routine displays a simple help text for the case selection
'       display pages for the SF600 module.
      SCREEN 0, 1, 3
      CLS
      CALL DrawWindow(1, 2, 8, 73, 2, framecolor)
      LOCATE 3, 5: PRINT "These are the real cases which you have saved.
       The dashes have been"
      LOCATE 4, 5: PRINT "eliminated from the SSN to save space.  To
       select a case printing, use"
      LOCATE 5, 5: PRINT "the arrow keys to highlight the appropriate
       case and then press the"
      LOCATE 6, 5: PRINT "ENTER/RETURN key to continue.  You may press
       the ESC key from the SSN"
      LOCATE 7, 5: PRINT "listing page to exit the program.  The PgUp
       and PgDn keys will move"
      LOCATE 8, 5: PRINT "you through the display pages of real cases."
      CALL TextPause
      SCREEN 0, 1, 0

END SUB

REM $STATIC
SUB splitem (a$, B$, APRINT$, txtwidth%)
' This combines a$ and b$ and if the resultant width is > txtwidth,
'splits the string (wordwraps) into APRINT$ with the rest in b$.
'   a$ - input string
'   b$ - buffer string  grows until > txtwidth
'   APRINT$ - output string - returns null if not enough string in b$
'   txtwidth% - width of formatted string (does not include indents).

'intitialize
  CONST punch$ = ".!?"


  ' allow a tab analog (4 spaces) here; otherwise strip spaces.
  IF LEFT$(a$, 4) <> "    " THEN
    a$ = LTRIM$(a$)
  END IF

  IF LEN(B$) > 0 THEN
    ' if a$ begins with punction, then don't insert space.
    ' also, spaces not added if a$ is NULL.
    IF INSTR(punch$, LEFT$(a$, 1)) = 0 THEN
      IF INSTR(punch$, RIGHT$(B$, 1)) > 0 THEN
B$ = B$ + "  " + a$
      ELSE
B$ = B$ + " " + a$
      END IF
    ELSE
      B$ = B$ + a$
    END IF
  ELSE
    B$ = a$
  END IF

  ' set up variables for exit without changes
  a$ = ""
  APRINT$ = ""

    ' check size
    lenb = LEN(B$)
    IF lenb < txtwidth THEN
      ' Too small - no changes necessary.
    ELSEIF lenb = txtwidth THEN
      ' just the right size
      APRINT$ = B$
      B$ = ""
    ELSE
      ' Too big; break into two
      ' look for rightmost space
      ptr = txtwidth + 1
      DO WHILE MID$(B$, ptr, 1) <> " " AND ptr > 1
ptr = ptr - 1
      LOOP

      ' Split 'em
      IF ptr = 1 THEN
' What! No spaces?
APRINT$ = LEFT$(B$, txtwidth)
B$ = MID$(B$, txtwidth + 1)
      ELSE
APRINT$ = LEFT$(B$, ptr - 1)
B$ = MID$(B$, ptr + 1)
B$ = LTRIM$(RTRIM$(B$))
      END IF

    END IF

END SUB

SUB TopMarginPrint (TOPMARGIN)
' This routine prints the proper number of CR's to make the top margin.

42400 IF TOPMARGIN <> 0 THEN
FOR I = 1 TO TOPMARGIN
  PRINT #2,
NEXT I
      END IF

END SUB
```

TEMPLATE.BAS

```
DECLARE SUB UpdateAge (agevar%, VARIABLE%())
DECLARE SUB GetUCResponse (ch$, filter$)
DECLARE FUNCTION ValidDate% (dat$)
DECLARE FUNCTION validtime% (ttime$)
DECLARE SUB templatehelp (helpstring$, a$, templatestring$, blankchar$,
       returncode%, errorstring$, errorflag%)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB parseline (x$, sep$, a$())
DECLARE SUB parseword (a$, sep$, word$)
DECLARE SUB headingPRINT (a$)
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB questionPRINT (a$)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DEFINT A-Z
DECLARE FUNCTION checktemplate% (ke$, checkchar$, blankchar$)
DECLARE SUB SkipHardForward (hardstring$, templatestring$, ptr%,
       length%)
DECLARE SUB SkipHardBackward (hardstring$, templatestring$, ptr%)
DECLARE SUB template (a$, templatestring$, blankchar$, returncode%)
DECLARE FUNCTION getkeycode% ()

' $INCLUDE: 'include.bas'

    FUNCTION checktemplate (ke$, checkchar$, blankchar$) STATIC
'       This function compares the character ke$ with the corresponding
'       template char, checkchar$ and returns 1 if OK, 0 if not.

'                 Numeric data required, and obligatory
        IF checkchar$ = "#" THEN
          IF ke$ >= "0" AND ke$ <= "9" THEN
          tempfunc = 1
          ELSE
            tempfunc = 0
          END IF
'                 Numeric data required, not obligatory
        ELSEIF checkchar$ = "%" THEN
          IF (ke$ >= "0" AND ke$ <= "9") OR ke$ = blankchar$ THEN
          tempfunc = 1
          ELSE
            tempfunc = 0
          END IF
'                 UC alphanumeric data required and obligatory.
        ELSEIF checkchar$ = "A" THEN
          ke$ = UCASE$(ke$)
          IF ke$ >= "A" AND ke$ <= "Z" THEN
            tempfunc = 1
          ELSE
            tempfunc = 0
          END IF
'                 UC alphanumeric data required, but not obligatory.
        ELSEIF checkchar$ = "Z" THEN
          ke$ = UCASE$(ke$)
          IF (ke$ >= "A" AND ke$ <= "Z") OR ke$ = blankchar$ THEN
            tempfunc = 1
          ELSE
            tempfunc = 0
          END IF
'               Male/Female [M/F] required.
        ELSEIF checkchar$ = "M" THEN
          ke$ = UCASE$(ke$)
          IF ke$ = "M" OR ke$ = "F" THEN
            tempfunc = 1
          ELSE
            tempfunc = 0
          END IF
'               Anything else passes OK
        ELSE
          tempfunc = 1
        END IF
        checktemplate = tempfunc

END FUNCTION

FUNCTION getkeycode STATIC
'       This function waits for a key to be pressed and returns its
'       unique key-code integer.
  DO
    k$ = INKEY$
  LOOP UNTIL k$ <> ""
  getkeycode = CVI(k$ + CHR$(0))
END FUNCTION

SUB parseline (x$, sep$, a$()) STATIC
'        Parses a line (x$) using sep$ as the separator descriptor, and
'       returns the parsed words in an array (a$() ).
  t$ = x$
  FOR i = LBOUND(a$) TO UBOUND(a$)
    CALL parseword(t$, sep$, a$(i))
    IF a$(i) = "" THEN EXIT FOR
  NEXT i
  t$ = ""

END SUB
```

```
SUB parseword (a$, sep$, word$)
'       Parses a word (word$) from a string (a$) using sep$ as the
'       string containing the separators.
  word$ = ""
  lena = LEN(a$)
  IF a$ = "" THEN EXIT SUB
  FOR i = 1 TO lena
    IF INSTR(sep$, MID$(a$, i, 1)) = 0 THEN EXIT FOR
  NEXT i
  FOR j = i TO lena
    IF INSTR(sep$, MID$(a$, j, 1)) THEN
      EXIT FOR
    END IF
  NEXT j
  FOR k = j TO lena
    IF INSTR(sep$, MID$(a$, k, 1)) = 0 THEN
      EXIT FOR
    END IF
  NEXT k
  IF i > lena THEN
    a$ = ""
    EXIT SUB
  END IF
  IF j > lena THEN
    word$ = MID$(a$, i)
    a$ = ""
    EXIT SUB
  END IF
  word$ = MID$(a$, i, j - i)
  IF k > lena THEN
    a$ = ""
  ELSE
    a$ = MID$(a$, k)
  END IF
END SUB

SUB SexSSNAgeDate (STFLAG, TRAINING, SIMULATE, sex$, SSN$, AGE$,
       STARTDATE$, STARTIME$, VARIABLE%())
'       This routine allows input for sex, ssn, age, date,and time.

'init sex
  IF VARIABLE(1) = 1 THEN
    sex$ = "M"
  ELSEIF VARIABLE(2) = 1 THEN
    sex$ = "F"
  ELSE
    sex$ = "_"
  END IF
```

```
'init time and date
 IF STARTIME$ = "" THEN STARTIME$ = LEFT$(TIME$, 5)
 IF STARTDATE$ = "" THEN STARTDATE$ = DATE$

  tempsex$ = sex$
  tempssn$ = SSN$
  tempage$ = AGE$
  tempdate$ = STARTDATE$
  temptime$ = STARTIME$

  SCREEN 0, 1, 0, 0
  CLS

  blankstring$ = "_"
  pointtoinput = 0
  finishpage = 0
  mainrow = 6
  maincol = 26
  errorcode = 0

  SexPageHeading1$ = "Sex / SSN / Age / Date / Time"
  SexPageHeading2$ = "Data Entry Page"
  SexHeading$ = "Patient's Sex [ ]"
  SSNHeading$ = "Patient's SSN [           ]"
  AgeHeading$ = "Patient's age [  ]"
  DateHeading$ = "Date of exam  [          ]"
  TimeHeading$ = "Time of exam  [     ]"
  Sexhelp$ = "Enter 'M' for male or 'F' for female.|This must be
       answered."
  Agehelp$ = "Enter the age of the patient ( between" + STR$(AGEMINIMUM)
       + " and" + STR$(AGEMAXIMUM) + ").|A valid age must be present to
       continue."
  sexerror$ = " Only M or F accepted."
  ssnerror$ = "Only numbers accepted for the SSN."
  ageerror$ = "Invalid age.  Must be between" + STR$(AGEMINIMUM) + "
       and" + STR$(AGEMAXIMUM) + ", inclusive."
  dateerror$ = "Invalid date.  Use format MM-DD-YYYY"
  timeerror$ = "Invalid time.  Use 24 hour format."

  IF TRAINING = 1 OR SIMULATE = 0 THEN
'        Modify helpstrings if either in training mode or simulated
       mode.
'        I know that SIMULATE should equal 1 for consistency, but it
       doesn't.
     SSNhelp$ = "A Social Security number has been chosen for you.|There
       is no need to change it.|With a REAL case, you would enter the
       patient's SSN here."
```

```
    Datehelp$ = "Today's date has been chosen for you.|There is no need
      to change it.|With a real CASE, you would enter the date of the
      exam here."
    Timehelp$ = "The current time has been chosen for you.|There is no
      need to change it.|With a real CASE, you would enter the time of
      the exam here."

  ELSE
    SSNhelp$ = "Enter the Social Security Number.|The hyphens will be
      added automatically.|A numeric SSN must be present to continue."
    Datehelp$ = "The current date has been chosen.|If this is
      incorrect, change it."
    Timehelp$ = "The current time has been chosen.|If this is
      incorrect, change it."
  END IF

'page heading
      CALL LocateCenter(2, SexPageHeading1$)
      headingPRINT (SexPageHeading1$)
      CALL LocateCenter(3, SexPageHeading2$)
      headingPRINT (SexPageHeading2$)
'frame data
      CALL SetFrameColor
      framtyp = frametype
      CALL frame(mainrow - 1, maincol - 2, 9, 28, framtyp)
      CALL SetNormalColor

'initially show question
        LOCATE mainrow, maincol
        questionPRINT (SexHeading$)
        LOCATE mainrow + 2, maincol
        questionPRINT (SSNHeading$)
        LOCATE mainrow + 4, maincol
        questionPRINT (AgeHeading$)
        LOCATE mainrow + 6, maincol
        questionPRINT (DateHeading$)
        LOCATE mainrow + 8, maincol
        questionPRINT (TimeHeading$)

        LOCATE mainrow, maincol + 15
        PRINT sex$;
        LOCATE mainrow + 2, maincol + 15
        PRINT SSN$;
        LOCATE mainrow + 4, maincol + 15
        PRINT AGE$;
        LOCATE mainrow + 6, maincol + 15
        PRINT STARTDATE$;
        LOCATE mainrow + 8, maincol + 15
        PRINT STARTIME$;
```

```
'       finishpage = 0    cycle thru for more input
'       finishpage = 1    exit sex/ssn/etc page

 DO
  SELECT CASE pointtoinput
   CASE 0
    'male/female stuff
    LOCATE mainrow, maincol + 15
    CALL templatehelp(Sexhelp$, tempsex$, "M", blankstring$, rc,
   sexerror$, errorcode)
'       rc = 0   CR
             1   Esc
'            2   up arrow
'            3   down arrow
    IF rc = 1 THEN
     finishpage = 1
    ELSEIF rc = 0 OR rc = 3 THEN
     pointtoinput = 1
    END IF

   CASE 1
    'ssn stuff
    LOCATE mainrow + 2, maincol + 15
    CALL templatehelp(SSNhelp$, tempssn$, "###-##-####",
   blankstring$, rc, ssnerror$, errorcode)
    IF rc = 1 THEN
     finishpage = 1
    ELSEIF rc = 0 OR rc = 3 THEN
     pointtoinput = 2
    ELSEIF rc = 2 THEN
     pointtoinput = 0
    END IF

   CASE 2
    'age stuff
    LOCATE mainrow + 4, maincol + 15
    CALL templatehelp(Agehelp$, tempage$, "##", blankstring$, rc,
   ageerror$, errorcode)
    IF rc = 1 THEN
     finishpage = 1
    ELSEIF rc = 0 OR rc = 3 THEN
     pointtoinput = 3
    ELSEIF rc = 2 THEN
     pointtoinput = 1
    END IF
    'if did not escape...
    IF rc <> 1 THEN
     'Check for valid age.  For ABDX,  age  >=17 and <=79.
    valage = VAL(tempage$)
    IF valage < AGEMINIMUM OR valage > AGEMAXIMUM THEN
      SOUND 900, 1
      finishpage = 0
      pointtoinput = 2
      errorcode = 1
    END IF
  END IF

CASE 3
  'date stuff
  LOCATE mainrow + 6, maincol + 15
  CALL templatehelp(Datehelp$, tempdate$, "##-##-####",
 blankstring$, rc, dateerror$, errorcode)
  IF rc = 1 THEN
    finishpage = 1
  ELSEIF rc = 0 OR rc = 3 THEN
    pointtoinput = 4
  ELSEIF rc = 2 THEN
    pointtoinput = 2
  END IF
  'if did not escape...
  IF rc <> 1 THEN
    'Check for valid date
    IF NOT (ValidDate%(tempdate$)) THEN
      SOUND 900, 1
      finishpage = 0
      pointtoinput = 3
      errorcode = 1
    END IF
  END IF

CASE 4
  'time stuff
  LOCATE mainrow + 8, maincol + 15
  CALL templatehelp(Timehelp$, temptime$, "##:##", blankstring$,
 rc, timeerror$, errorcode)
  IF rc = 1 THEN
    finishpage = 1
  ELSEIF rc = 0 OR rc = 3 THEN
    pointtoinput = 5
  ELSEIF rc = 2 THEN
    pointtoinput = 3
  END IF
  'if did not escape...
  IF rc <> 1 THEN
    'Check for valid time
    IF NOT validtime(temptime$) THEN
      SOUND 900, 1
```

```
            finishpage = 0
            pointtoinput = 4
            errorcode = 1
          END IF
        END IF

      CASE 5
        ' Routine to check if all responses entered are OK
        ' get confirmation
        CALL scrollup(17, 2, 23, 79, 0, backcolor)
        LOCATE 20, 26
        PRINT "Are These correct? (Y/N) [ ]";
        LOCATE 20, 52
        YNEsc$ = "YN" + CHR$(27)
        CALL GetUCResponse(ch$, YNEsc$)
        ' if not OK then cycle back
        IF ch$ = "N" THEN
          finishpage = 0
          pointtoinput = 0
        ELSE
          IF ch$ = CHR$(27) THEN rc = 1
          finishpage = 1
        END IF

      CASE ELSE
        'shouldn't get here.
        finishpage = 1

      END SELECT
    LOOP UNTIL finishpage = 1

'       Since using temp values, don't need to change anything if Escape
'       was pressed.  Check for other than Escape
    IF rc <> 1 THEN
'       Check if any variables are different from  original.  If so,
       then
'       do update STFLAG
      IF tempsex$ <> sex$ THEN
        STFLAG = 1
        sex$ = tempsex$
        IF tempsex$ = "M" THEN
          VARIABLE(1) = 1
          VARIABLE(2) = 0
        ELSE                            'if not male, then by def,
       female.
          VARIABLE(1) = 0
          VARIABLE(2) = 1
        END IF
      END IF
      IF tempssn$ <> SSN$ THEN
        STFLAG = 1
        SSN$ = tempssn$
      END IF
      IF tempage$ <> AGE$ THEN
        STFLAG = 1
        AGE$ = tempage$
        agevar = VAL(AGE$)
        CALL UpdateAge(agevar%, VARIABLE%())
      END IF
      IF tempdate$ <> STARTDATE$ THEN
        STFLAG = 1
        STARTDATE$ = tempdate$
      END IF
      IF temptime$ <> STARTIME$ THEN
        STFLAG = 1
        STARTIME$ = temptime$
      END IF
    END IF


END SUB

  SUB SkipHardBackward (hardstring$, templatestring$, ptr)
'       This routine will decrement ptr to the appropriate location
'       skipping any hard chars in the template

  DO WHILE INSTR(hardstring$, MID$(templatestring$, ptr + 1, 1)) <> 0
       AND ptr > 0
    ptr = ptr - 1
  LOOP
END SUB

  SUB SkipHardForward (hardstring$, templatestring$, ptr, length)
'       This routine will increment ptr to the appropriate location
'       skipping any hard chars in the template

  DO WHILE INSTR(hardstring$, MID$(templatestring$, ptr + 1, 1)) <> 0
       AND ptr < length
    ptr = ptr + 1
  LOOP

END SUB

SUB template (a$, templatestring$, blankchar$, returncode%)
'       This routine will display a$ at the current location
'       and replace any deletions with blankchar$.  It will allow
'       modification of the string using keys and the template.
'       Returncode will return 0,1,2,or 3 if CR, Esc, up arrow, or down
       arrow.
'       Template now is: # - 0-9
'                        A -alphanumeric
'                        - - a dash , a hard character.  You can't
       modify it.
'                        : - a colon, another hard character.

'initialize stuff
  CONST CR = 13
  CONST esc = 27
  CONST uparrow = 18432
  CONST downarrow = 20480
  CONST leftarrow = 19200
  CONST rightarrow = 19712
  CONST home = 18176
  CONST endkey = 20224
  CONST backspace = 8
  CONST delete = 21248

  hardstring$ = "-:"
  quitsub = 0
  startcol = POS(0)
  startrow = CSRLIN
  length = LEN(templatestring$)
  oldptr = 0
  ptr = 0

'       Main loop
 DO
  DO
    LOCATE startrow, startcol, 0
    PRINT a$;
'       Position blinking cursor
    IF ptr < 0 THEN ptr = 0
    IF ptr > length - 1 THEN ptr = length - 1

    IF ptr < oldptr THEN
      CALL SkipHardBackward(hardstring$, templatestring$, ptr)
    ELSE
      CALL SkipHardForward(hardstring$, templatestring$, ptr, length)
    END IF
    LOCATE startrow, startcol + ptr, 1, 6, 7
    oldptr = ptr

    keynum = getkeycode

    SELECT CASE keynum
    CASE backspace
        MID$(a$, ptr + 1, 1) = blankchar$
        ptr = ptr - 1

    CASE delete
        MID$(a$, ptr + 1, 1) = blankchar$

    CASE uparrow
      returncode = 2
      quitsub = 1

    CASE downarrow
      returncode = 3
      quitsub = 1


    CASE leftarrow
      IF ptr > 0 THEN
        ptr = ptr - 1
      END IF

    CASE rightarrow
    ' should be 2?
      IF ptr < length - 1 THEN
        ptr = ptr + 1
      END IF

    CASE home
      ptr = 0

    CASE endkey
      ptr = length - 1

    CASE esc
      returncode = 1
      quitsub = 1

    CASE CR
      returncode = 0
      quitsub = 1

    CASE IS < 32
      SOUND 999, 1

    CASE IS > 255
      SOUND 999, 1

    CASE ELSE
'       all writable characters, I hope.
```

```
      ke$ = CHR$(keynum)

      IF ptr < length THEN
'               Check template.
        checkchar$ = MID$(templatestring$, ptr + 1, 1)
        IF checktemplate(ke$, checkchar$, blankchar$) = 1 THEN
          IF ptr < LEN(a$) THEN
            MID$(a$, ptr + 1, 1) = ke$
          ELSE
            a$ = a$ + ke$
          END IF
            ptr = ptr + 1
        ELSE
          SOUND 54, 5
        END IF

      ELSE
        SOUND 999, 1
      END IF

    END SELECT
  LOOP UNTIL quitsub > 0

'       Finally, check that obligatory fields are filled in, unless
'       escaping
  obligflag = 1
  IF returncode <> 1 THEN
    FOR i = 1 TO length
      checkchar$ = MID$(templatestring$, i, 1)
      mainchar$ = MID$(a$, i, 1)
      IF checktemplate(mainchar$, checkchar$, blankchar$) = 0 THEN
        obligflag = 0
      END IF

    NEXT i
    quitsub = 0
  END IF
 LOOP UNTIL obligflag > 0


END SUB

SUB templatehelp (helpstring$, a$, templatestring$, blankchar$,
       returncode%, errorstring$, errorflag%)
'       This routine displays a help string at the bottom of the page
'       and then calls subroutine template.
'       Also displays errorstring if errorflag =1
'       Save cursor position while typing help text.
 DIM hlp$(2)
 origrow = CSRLIN
 origcol = POS(0)
 helprow = 19

 CALL scrollup(helprow - 2, 2, helprow - 2, 79, 0, backcolor)
 IF errorflag = 1 THEN
   CALL LocateCenter(helprow - 2, errorstring$)
   PRINT errorstring$;
   SOUND 999, 1
   errorflag = 0
 END IF

 CALL parseline(helpstring$, "|", hlp$())

 CALL scrollup(helprow, 2, helprow + 4, 79, 0, backcolor)
 SetColor (infocolor)
 FOR i = 0 TO 2
   IF hlp$(i) <> "" THEN
     CALL LocateCenter(helprow + i, hlp$(i))
     PRINT hlp$(i);
   END IF
 NEXT i
 SetColor (forecolor)

 LOCATE origrow, origcol
 CALL template(a$, templatestring$, blankchar$, returncode%)

 ERASE hlp$

END SUB

FUNCTION validtime (ttime$)
'       This routine returns true (non-0) if the ttime$ represents
'       a valid time in 24 hour format.
 CONST FALSE = 0
 CONST TRUE = NOT FALSE

 hr$ = LEFT$(ttime$, 2)
 min$ = RIGHT$(ttime$, 2)
 IF hr$ < "00" OR hr$ > "24" OR min$ < "00" OR min$ > "59" THEN
   validtime = FALSE
   EXIT FUNCTION
 END IF
 IF hr$ = "24" AND min$ <> "00" THEN
   validtime = FALSE
   EXIT FUNCTION
 END IF
```

```
    validtime = TRUE

END FUNCTION
```

FPRINT.ASM

```
;This routine will print a string of text faster than QuickBASIC does.
; It will check for a monochrome monitor or CGA/EGA.  It should only be
;used in text modes, ie where the video RAM begins at either B000h
       (mono)
;or B800h (color).  It will write to the current page.  You send the
;text string and attribute in the form:

;  CALL fprint(text$,attr%)


DATASEGMENT WORD PUBLIC 'DATA'

dispage   db  ?
attrib  db  ?
vidseg  dw  ?

DATAENDS

DGROUP  GROUP   DATA

CODESEGMENT WORD PUBLIC 'CODE'

ASSUME  CS:CODE,DS:DGROUP,ES:DGROUP,SS:DGROUP

Public  FPRINT

FPRINT  Proc  Far
  push  BP
   mov  BP,SP
   mov  AH,0Fh
   int  10h     ; get screen mode, return it in AL
   mov  dispage,BH  ; current display page returned in BH
   mov  BX,0B000h   ; add video ram base to it.
   cmp  AL,7    ; is it monochrome display?
    jz  mono    ; yes then skip add
   add  BX,0800h
mono:   mov   vidseg,BX  ;Now BX contains the segment address for
            ; the video ram
   mov  AH,3
   mov  BH,dispage
   int  10h     ; read cursor position, DH=row, DL=col
   mov  AL,DH
   mov  CL,160    ;multiply row by 80 chars and 80 attrib
   mul  CL    ; per line.
   mov  DH,0
   add  AX,DX     ;add col, correcting for chars
   add  AX,DX     ;add again, correcting for attributes

   mov  BH,dispage  ; put display page in BH
   mov  BL,0    ;  then multiply by 1000h using shifts
   shl  BX,1    ;  to compute page offset from begining
   shl  BX,1    ;  of video ram.
   shl  BX,1
   shl  BX,1
   add  ax,bx     ;add display page offset to cursor offset
   mov  DI,AX     ;save video ram offset address in DI

   mov  ES,vidseg
   mov  SI,[BP+06]
   mov  AH,[SI]
   mov  SI,[BP+08]
   mov  CX,[SI]
  jcxz  exit
   mov  SI,[SI+02]
   cld
riteit:   lodsb
 stosw
  loop  riteit


exit:   pop   BP
   ret  4

FPRINT  ENDP



;-----------------------------------------------------
;
;  routine to scroll active window up  SCROLINE% lines at a time
;
;  QuickBASIC declaration:
;    CALL SCROLLUP(Lrow%,Lcol%,Rrow%,Rcol%,scroline%,attribute%)
;      note: 1,1 = upper left hand corner
;
;NOTE - This routine was modified from  the one called by TICK,
; a MS Pascal program.  That original one passed values by value,
; whereas QB passes by reference.
;
public  SCROLLUP
SCROLLUP proc far
        push bp
        mov bp,sp
        mov si,[bp+14]  ;get lcol in cl
mov cx,[si]
```

```
        dec cl          ;adjust frame of reference
        mov si,[bp+16]
mov ax,[si]
        mov ch,al       ;get lrow in ch
        dec ch          ;adjust
        mov si,[bp+10]  ;get rcol in dl
mov dx,[si]
        dec dl          ;adjust
        mov si,[bp+12]
mov ax,[si]
        mov dh,al       ;get rrow in dh
        dec dh          ;adjust
        mov si,[bp+6]
mov bx,[si]
        mov bh,bl       ;get attribute in bh
        mov si,[bp+8]   ;get number of lines to scroll in al
mov ax,[si]
        mov ah,6
        int 10h
        pop bp
        ret 12
SCROLLUP endp
;
;
;
;----------------------------------------------------
;
;  routine to scroll active window down  SCROLINE% lines at a time
;
;  QuickBASIC declaration:
;    CALL SCROLLDN(Lrow%,Lcol%,Rrow%,Rcol%,scroline%,attribute%)
;
;      note: 1,1 = upper left hand corner
;
;NOTE - This routine was modified from  the one called by TICK,
; a MS Pascal program.  That original one passed values by value,
; whereas QB passes by reference.
;
public  SCROLLDN
SCROLLDN proc far
        push bp
        mov bp,sp
        mov si,[bp+14]  ;get lcol in cl
mov cx,[si]
        dec cl          ;adjust frame of reference
        mov si,[bp+16]
mov ax,[si]
        mov ch,al       ;get lrow in ch
        dec ch          ;adjust
        mov si,[bp+10]  ;get rcol in dl
mov dx,[si]
        dec dl          ;adjust
        mov si,[bp+12]
mov ax,[si]
        mov dh,al       ;get rrow in dh
        dec dh          ;adjust
        mov si,[bp+6]
mov bx,[si]
        mov bh,bl       ;get attribute in bh
        mov si,[bp+8]   ;get number of lines to scroll in al
mov ax,[si]
        mov ah,7
        int 10h
        pop bp
        ret 12
SCROLLDN endp
;
;
CODEENDS
   END
```

INTRPT.ASM

This program is included with Microsoft QuickBASIC 4.0 and higher. It is therefore not listed here.

CIPHER.C

```
/* Routines used in encryption and decryption of .DAT files in ABDX and
   CPDX.
   These two routines are called by QuickBASIC.

*/

/* define QB string descriptor structure */
struct bas_str
  {
  int s_len;
  char *s_addr;
  };

  void encipher(QB_string)
  struct bas_str *QB_string;
    {
    int len=QB_string->s_len;
    unsigned char *str=QB_string->s_addr;
    register i;
    unsigned a_byte;
    if (len>0)
      {
      for (i=0;i<len;i++)
     {
a_byte=*(str+i);
/* take a char, add 1, and XOR the result with 0 thru 31, then
   XOR with 3 */
*(str+i)=(a_byte^(i%31)^3) +1;
     }
      }
    }

  void decipher(QB_string)
  struct bas_str *QB_string;
    {
    int len=QB_string->s_len;
    unsigned char *str=QB_string->s_addr;
    register i;
    unsigned a_byte;
    if (len>0)
      {
      for (i=0;i<len;i++)
     {
a_byte=*(str+i);
/* take a char, subtract 1, and XOR the result with 0 thru 32 */
*(str+i)=(--a_byte)^(i%31)^3;
```

```
     }
      }
    }

/*
   main()
   {
   }
*/
```

Appendix B
Utility Program Listings

ABDXSTAT.BAS

```
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB STATSexSSNAgeDate (STFLAG%, TRAINING%, SIMULATE%, sex$,
       SSN$, AGE$, STARTDATE$, STARTIME$, VARIABLE%())
DECLARE SUB GetUCResponse (ch$, filter$)
DECLARE FUNCTION ValidDate% (dat$)
DECLARE FUNCTION validtime% (ttime$)
DECLARE SUB templatehelp (helpstring$, a$, templatestring$, blankchar$,
       returncode%, errorstring$, errorflag%)
DECLARE SUB GetSF600Case (RealFileName$, heading$, FORGND%, BACGND%,
       RECORD%)
DECLARE SUB CompareFemDXes (HMDX%, MAXNUM%)
DECLARE SUB TextPause ()
DECLARE SUB SetVideoMode (vm%)
DECLARE SUB experimental ()
DECLARE FUNCTION Translate% (HMDX%)
DECLARE SUB SF600 (BOAT1$, BOAT2$, HMNAM$, HMSSN$)
DECLARE SUB CompareAbdDXes (COMPAR%(), VARIABLE%(), MAXNUM%, HMDX%,
       QUESTPTR%(), QUESTIONS$())
DECLARE FUNCTION Centered% (s$)
DECLARE SUB SelectDatabase (lookatfemale%, VARIABLE%())
DECLARE SUB ComputeFinalProbs (MAXNUMBER%, MAXPROBABILITY%, PROB#(),
       FINALPROB#())
DECLARE SUB TXMenu (MAXNUM%, sex$)
DECLARE SUB MaleGraph (FINPROB#())
DECLARE SUB DisplayEncryptedFile (TheFile$, ReturnPage%)
DECLARE SUB FemaleGraph (FINPROB#())
DECLARE SUB TextDxPause ()
DECLARE SUB PutCase (whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
DECLARE SUB ResetVariables (VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
DECLARE SUB MenuSummaryPage (menurow%, menucol%, NR%, resplength%,
       exitchar$, menuheading$, Choices$(), HELPFILE$)
DECLARE SUB MenuEntryPage (NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
DECLARE SUB GetCase (filnam$, whichcase%, VARIABLE%(), SSN$, AGE$,
       OTHER$, STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
DECLARE SUB PackArray (PackString$, thearray%())
DECLARE SUB ModifyNarrative (CASENUM%, escflag%)
```

ABDXSTAT.BAS (cont'd)

```
DECLARE SUB UnPackArray (PackString$, thearray%())
DECLARE SUB DisplayHPgetstatments (SXloc%(), SXresp$(), abortHP%)
DECLARE SUB DisplayMissedHP (SSN$, STARTIME$, STARTDATE$, VARIABLE%(),
       THECASE%())
DECLARE SUB DisplayHProwcol (row%, col%, sxstrng$)
DECLARE SUB CenterString (infostring$)
DECLARE SUB DisplayHPFrame (TRAINING%, SIMULATE%, SSN$, STARTIME$,
       STARTDATE$)
DECLARE SUB DisplayHP (TRAINING%, SIMULATE%, SSN$, STARTIME$,
       STARTDATE$, VARIABLE%())
DECLARE SUB DisplayHPprint (HP%, SXloc%(), SXresp$(), VARIABLE%())
DECLARE SUB TextContinuePrompt ()
DECLARE SUB DisplayHPhelp ()
DECLARE SUB DisplayHPTitle (HP%)
DECLARE FUNCTION Exists% (FIL$)
DECLARE SUB PaintGraph (VAR%, WhichOne%)
DECLARE SUB DrawGraph (WhichOne%)
DECLARE FUNCTION VideoMode% ()
DECLARE SUB DataEntryPage (exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
DECLARE SUB PutCursor (quest%, resp%, Choices$())
DECLARE SUB UpdateAsterisk (FirstRow%, FirstCol%, NonePtr%,
       VariablePtr%, GraphFlag%, Numresp%, OffSet%, VARIABLE%())
DECLARE SUB LocateCenter (crow%, infostring$)
DECLARE SUB HPframe ()
DEFINT A-Z
DECLARE SUB SexSSNAgeDate (STFLAG%, TRAINING%, SIMULATE%, sex$, SSN$,
       AGE$, STARTDATE$, STARTIME$, VARIABLE%())
DECLARE SUB InitializeColors (graphmode$, monmode$)
DECLARE SUB GetGraphMode (graphmode$, monmode$)
DECLARE SUB SetColor (thecolor%)
DECLARE SUB hpresponsePRINT (a$)
DECLARE SUB textPRINT (a$)
DECLARE SUB questionPRINT (a$)
DECLARE SUB headingPRINT (a$)
DECLARE SUB responsePRINT (a$)
DECLARE SUB SetTrainingColors (TRAINING%)
DECLARE SUB SetFrameColor ()
DECLARE SUB SetNormalColor ()
DECLARE SUB SetScreenMode (ScrnMode%)
DECLARE SUB GetKey (a$)
DECLARE SUB narrative (NR%, CASENUM%, THECASE%())
DECLARE SUB LoadTrainingCase (CASENUM%, THECASE%())
DECLARE SUB frame (ulr%, ulc%, numlines%, length%, frametyp%)
DECLARE SUB GetBoatStuff (BOAT1$, BOAT2$, HMNAM$, HMSSN$)
DECLARE SUB Disclaimer (VERSION$)
DECLARE SUB GraphContinuePrompt ()
DECLARE SUB BoxSelections (actyl%, XPOINT%, NumOfResp%, Xwidth%)

COMMON SHARED /normalcolor/ forecolor AS INTEGER, backcolor AS INTEGER
COMMON SHARED /ColorName1/ black%, blue%, green%, cyan%, red%, magenta%
COMMON SHARED /ColorName2/ brown%, white%, gray%, ltblue%, ltgreen%,
       ltcyan%
COMMON SHARED /ColorName3/ ltred%, ltmagenta%, yellow%, hiwhite%
COMMON SHARED /DefaultStuff/ headingcolor%, textcolor%, framecolor%,
       questioncolor%, responsecolor%, hpresponsecolor%, graphcolor%,
       helpcolor%, frametype%
COMMON SHARED /DefaultStuff2/ infocolor%
COMMON SHARED /ScreenStuff/ GRAPHICS%, MONITOR%, ScrnMode AS INTEGER,
       Vertbits AS INTEGER, YOffsetpict AS INTEGER
COMMON SHARED /GraphStuff/ hpframecolor%, bargraph%()

' $INCLUDE: 'include.bas'
black% = 0
blue% = 1
green% = 2
cyan% = 3
red% = 4
magenta% = 5
brown% = 6
white% = 7
gray% = 8
ltblue% = 9
ltgreen% = 10
ltcyan% = 11
ltred% = 12
ltmagenta% = 13
yellow% = 14
hiwhite% = 15

'      Dummy values so that variables are declared in main module.  They
       are
'       modified shortly by SetTrainingColors.
headingcolor% = 1
framecolor% = 1
frametype% = 1

  REM Copyright (C) 1985,1986,1987,1988 Navy Submarine Medical Research
       Laboratory
  KEY OFF
      RANDOMIZE TIMER
  DEFINT A-Z
REM $DYNAMIC
   DIM VARIABLE%(200)
   DIM question$(10), MULTIP%(10), Numresp%(10)
   DIM None(10), GraphFlag(10), SKIPBLANK(10), VariablePtr(10)
   DIM Choices$(10, 14)
   DIM QUESTPTR%(44), QUESTIONS$(44), DIFFER%(15)
   DIM BAYES!(154, 7), PROB#(7), FINPROB#(7)
   DIM FEMBAYES!(174, 7), FEMPROB#(7), FEMFINPROB#(7)

      DIM COMPAR%(6, 6, 15)

   DIM THECASE%(200), INARRAY%(7), OUTARRAY%(7), bargraph%(7)

'   male = 0
'   female = 1

' the following is added for testing purposes. but allows the program
' to be used normally, if "test" is not used at the command prompt.
' ie, ABDX test <CR>.
IF COMMAND$ = "TEST" THEN
  davidflag% = 1
ELSE
  davidflag% = 0
END IF


   VERSION$ = " STATS (ver 3.00)"
      TRAINING = 0             ' 0 = main  ;  1 = training
      STFLAG = 0               ' 0 = no changes; 1 = changes have been
       made
      SIMULATE = 1             ' 1 = main  ;  0 = simulated (I know, I
       know!)

'      CALL SetTrainingColors(TRAINING)
      REM main prog > TRAINING=0; training prog > TRAINING=1

      CALL GetGraphMode(graphmode$, monmode$)

'       Set up graphics mode default (checked for CGA or EGA in
       graphmode$)
      CALL InitializeColors(graphmode$, monmode$)
'       Initialize display page colors
      CALL SetTrainingColors(TRAINING)

      ' BIOS to appropriate 80 col text mode
      IF monmode$ = "C" THEN
      '  CALL SetVideoMode(3)    'screen 0, 80 col, color
      ELSE
      '  CALL SetVideoMode(2)    'screen 0, 80 col, B&W
      END IF


      ' experimental program
      CALL experimental

'       Get name of vessel, user's name, and display submarine if
       present.
      CALL GetBoatStuff(BOAT1$, BOAT2$, HMNAM$, HMSSN$)
'            Go to subroutine to enter Bayesian probabilities,
' response and category names,and clear array variable.
      GOSUB 60000
      CALL GetKey(a$)
'       Subprogram which displays warning/disclaimer.
      CALL Disclaimer(VERSION$)

   SELECT CASE GRAPHICS
     CASE 9
       Vertbits - 14
       YOffsetpict - 4
     CASE 2
       Vertbits - 8
       YOffsetpict - 6
     CASE ELSE
       Vertbits - 8
       YOffsetpict - 6
   END SELECT

   GOTO 31000

'   REM   Enter SS#, Age.
30  'stat change
    CALL STATSexSSNAgeDate(STFLAG, TRAINING, SIMULATE, sex$, SSN$, AGE$,
       STARTDATE$, STARTIME$, VARIABLE%())
    'CALL SexSSNAgeDate(STFLAG, TRAINING, SIMULATE, sex$, SSN$, AGE$,
       STARTDATE$, STARTIME$, VARIABLE%())

100 REM Data Entry Option Page
    ' Choices for Main Option Page.
    Choices$(1, 1) - "GO TO HISTORY PAGES       "
    Choices$(1, 2) - "GO TO PHYSICAL EXAM PAGES "
    Choices$(1, 3) - "MAKE DIAGNOSIS            "
    Choices$(1, 4) - "GO TO SSN/AGE/TIME PAGE   "
    Choices$(1, 5) - "RETURN TO MAIN OPTION PAGE"
    IF TRAINING - 1 THEN
      Choices$(1, 5) - "GO TO TRAINING OPTION PAGE"
      TYP$ - "Training "
    ELSE
      TYP$ - ""
    END IF

    ' New method for menu
    NR% - 1
    resplength% = 5
    DATAHEADING$ = "Abdominal Pain Diagnosis " + TYP$ + "Program" +
       VERSION$
    menuheading$ = "Data Entry Options:"
    HELPFILE$ = "HP13.DAT"

    CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
    ON NR GOTO 1000, 5000, 50000, 30, 31000
    GOTO 100


1000 REM PAGE 1 of Hx.  Similar to above page.
     IF sex$ = FEMALE$ THEN
       MaxHxPages$ = "5"
     ELSE
       MaxHxPages$ = "4"
     END IF

1010 NUMQUEST = 2
     NUMCOL1QUESTS = 1
     TOPROW = 4
     TOPCOL = 19
     HELPFILE$ = "H14.TXT"
     DATAHEADING$ = "History"
     PAGEOF$ = "1"
     OFPAGE$ = MaxHxPages$

     FOR i = 1 TO 2
       question$(i) = QUESTIONS$(i)
       MULTIP%(i) = 1
       Numresp%(i) = QUESTPTR%(i)
       None(i) = 13
       GraphFlag(i) = i
       SKIPBLANK(i) = 2
Choices$(i, 1) = "   RUQ     ": Choices$(i, 2) = "   LUQ      "
Choices$(i, 3) = "   RLQ     ": Choices$(i, 4) = "   LLQ      "
Choices$(i, 5) = "UPPER HALF ": Choices$(i, 6) = "LOWER HALF  "
Choices$(i, 7) = "RIGHT HALF ": Choices$(i, 8) = "LEFT HALF   "
Choices$(i, 9) = "  CENTRAL  ": Choices$(i, 10) = " GENERAL    "
       Choices$(i, 11) = "RIGHT FLANK": Choices$(i, 12) = "LEFT FLANK "
       Choices$(i, 13) = " NO PAIN    "
     NEXT i

       VariablePtr(1) = 11
       VariablePtr(2) = 24
       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)

     IF exitchar$ = "P" THEN 100
     IF exitchar$ = "X" THEN 100

2000 REM Page 2 of Hx
' Multip (Whether Multiple Responses are Possible (1)
' or Not (0)); and Numresp (Number of Symptoms in Each Category).

     NUMQUEST = 6
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H24.TXT"
     DATAHEADING$ = "History"
     PAGEOF$ = "2"
     OFPAGE$ = MaxHxPages$
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 2)
     NEXT i

     question$(1) = QUESTIONS$(3)
       Choices$(1, 1) = "Intermittent"
       Choices$(1, 2) = "Steady      "
       Choices$(1, 3) = "Colicky     "
       VariablePtr(1) = 37
       MULTIP(1) = 0
       None(1) = 0
       GraphFlag(1) = 0
       SKIPBLANK(1) = 0

     question$(2) = QUESTIONS$(4)
       Choices$(2, 1) = "Moderate    "
       Choices$(2, 2) = "Severe      "
       VariablePtr(2) = 40
       MULTIP(2) = 0
       None(2) = 0
       GraphFlag(2) = 0
       SKIPBLANK(2) = 1

     question$(3) = QUESTIONS$(5)
       Choices$(3, 1) = "Movement    "
       Choices$(3, 2) = "Cough       "
       Choices$(3, 3) = "Breathing   "
       Choices$(3, 4) = "Food        "
       Choices$(3, 5) = "Other       "
```

```
   Choices$(3, 6) = "None        "
   VariablePtr(3) = 42
   MULTIP(3) = 1
   None(3) = 6
   GraphFlag(3) = 0
   SKIPBLANK(3) = 3

  question$(4) = QUESTIONS$(6)
   Choices$(4, 1) = "Better      "
   Choices$(4, 2) = "Same        "
   Choices$(4, 3) = "Worse       "
   VariablePtr(4) = 48
   MULTIP(4) = 0
   None(4) = 0
   GraphFlag(4) = 0
   SKIPBLANK(4) = 0

  question$(5) = QUESTIONS$(7)
   Choices$(5, 1) = "12h or less "
   Choices$(5, 2) = "12-24h      "
   Choices$(5, 3) = "24-48h      "
   Choices$(5, 4) = "48+h        "
   VariablePtr(5) = 51
   MULTIP(5) = 0
   None(5) = 0
   GraphFlag(5) = 0
   SKIPBLANK(5) = 1

  question$(6) = QUESTIONS$(8)
   Choices$(6, 1) = "Lying Still "
   Choices$(6, 2) = "Vomiting    "
   Choices$(6, 3) = "Antacids    "
   Choices$(6, 4) = "Food        "
   Choices$(6, 5) = "Other       "
   Choices$(6, 6) = "None        "
   VariablePtr(6) = 55
   MULTIP(6) = 1
   None(6) = 6
   GraphFlag(6) = 0
   SKIPBLANK(6) = 1

   CALL DataEntryPage(exitchar$, question$(), Choices$(),
   VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
   GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
   NUMQUEST%, DATAHEADINGS$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
   IF exitchar$ = "P" THEN 1010
   IF exitchar$ = "X" THEN 100

3000 REM Page 3 of Hx
```

```
NUMQUEST = 6
NUMCOL1QUESTS = 3
TOPROW = 4
TOPCOL = 7
HELPFILE$ = "H34.TXT"
DATAHEADING$ = "History -  Other Symptoms"
PAGEOF$ = "3"
OFPAGE$ = MaxHxPages$
FOR i = 1 TO NUMQUEST
  Numresp%(i) = QUESTPTR%(i + 8)
  GraphFlag(i) = 0
  question$(i) = QUESTIONS$(8 + i)
NEXT i

Choices$(1, 1) = "Present       ": Choices$(1, 2) = "Absent
  "
VariablePtr(1) = 61
MULTIP(1) = 0
None(1) = 0
SKIPBLANK(1) = 1

Choices$(2, 1) = "Present       ": Choices$(2, 2) = "Absent
  "
VariablePtr(2) = 63
MULTIP(2) = 0
None(2) = 0
SKIPBLANK(2) = 1

Choices$(3, 1) = "Normal        ": Choices$(3, 2) = "Constipated
  "
Choices$(3, 3) = "Diarrhea      ": Choices$(3, 4) = "Blood in
  Stool"
Choices$(3, 5) = "Mucus in Stool"
VariablePtr(3) = 65
MULTIP(3) = 1
None(3) = 1
SKIPBLANK(3) = 1

Choices$(4, 1) = "Decreased     ": Choices$(4, 2) = "Normal
  "
VariablePtr(4) = 70
MULTIP(4) = 0
None(4) = 0
SKIPBLANK(4) = 1

Choices$(5, 1) = "Present       ": Choices$(5, 2) = "Absent
  "
VariablePtr(5) = 72
```

```
     MULTIP(5) = 0
     None(5) = 0
     SKIPBLANK(5) = 1

     Choices$(6, 1) = "Normal        ": Choices$(6, 2) = "Frequency
       "
     Choices$(6, 3) = "Painful       ": Choices$(6, 4) = "Dark Urine
       "
     Choices$(6, 5) = "Blood in Urine"
     VariablePtr(6) = 74
     MULTIP(6) = 1
     None(6) = 1
     SKIPBLANK(6) = 1

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 2000
       IF exitchar$ = "X" THEN 100

4000 REM PAGE 4 of Hx
     NUMQUEST = 5
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H44.TXT"
     DATAHEADING$ = "History -  Past History"
     PAGEOF$ = "4"
     OFPAGE$ = MaxHxPages$

     FOR i = 1 TO NUMQUEST
       MULTIP%(i) = 0
       Numresp%(i) = QUESTPTR%(i + 14)
       question$(i) = QUESTIONS$(14 + i)
       Choices$(i, 1) = "Yes": Choices$(i, 2) = "No"
       None(i) = 0
       GraphFlag(i) = 0
       VariablePtr(i) = 79 + (i - 1) * 2
       SKIPBLANK(i) = 1
     NEXT i


     CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
     IF exitchar$ = "P" THEN 3000
     IF exitchar$ = "X" THEN 100


4500 IF sex$ = FEMALE$ THEN
       REM PAGE 5 of Hx  Only shown if patient is female.
       NUMQUEST = 6
       NUMCOL1QUESTS = 3
       TOPROW = 4
       TOPCOL = 7
       HELPFILES$ = "H54.TXT"
       DATAHEADING$ = "History -  OB/GYN"
       PAGEOF$ = "5"
       OFPAGE$ = MaxHxPages$
       FOR i = 1 TO NUMQUEST
 Numresp%(i) = QUESTPTR%(i + 38)
 GraphFlag(i) = 0
 question$(i) = QUESTIONS$(i + 38)
 MULTIP(i) = 0
 SKIPBLANK(i) = 1
 None(i) = 0
       NEXT i

       Choices$(1, 1) = "Not Started": Choices$(1, 2) = "Ceased"
       Choices$(1, 3) = "Regular": Choices$(1, 4) = "Irregular"
       VariablePtr(1) = 160

       Choices$(2, 1) = "Normal": Choices$(2, 2) = "Late/Overdue"
  '     CHOICES$(2, 3) = "Overdue"  -- combined with LATE in the
       database.
       VariablePtr(2) = 164

       Choices$(3, 1) = "Yes": Choices$(3, 2) = "No"
       VariablePtr(3) = 166

       Choices$(4, 1) = "Impossible": Choices$(4, 2) = "Possible"
       Choices$(4, 3) = "Confirmed"
       VariablePtr(4) = 168

       Choices$(5, 1) = "Yes": Choices$(5, 2) = "No"
       VariablePtr(5) = 171

       Choices$(6, 1) = "Yes": Choices$(6, 2) = "No"
       VariablePtr(6) = 173

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
```

```
    IF exitchar$ = "P" THEN 4000
    ' "X" and "N" will fall through the END IF and will goto 100
    ' automatically.
  END IF
  GOTO 100

5000 REM Page 5; Page 1 of PE.
  NUMQUEST = 4
  NUMCOL1QUESTS = 2
  TOPROW = 4
  TOPCOL = 7
  HELPFILE$ = "H15.TXT"
  DATAHEADING$ = "Physical-  Vital Signs"
  PAGEOF$ = "1"
  OFPAGE$ = "5"

  FOR i = 1 TO NUMQUEST
    MULTIP%(i) = 0
    Numresp%(i) = QUESTPTR%(i + 19)
    None(i) = 0
    GraphFlag(i) = 0
    question$(i) = QUESTIONS$(i + 19)
  NEXT i

  Choices$(1, 1) = "< 98.6      ": Choices$(1, 2) = "98.6 - 100.2"
  Choices$(1, 3) = "100.3 - 102 ": Choices$(1, 4) = "> 102       "
  VariablePtr(1) = 89
  SKIPBLANK(1) = 1

  Choices$(2, 1) = "< 80        ": Choices$(2, 2) = "80 - 99     "
  Choices$(2, 3) = "> 99        "
  VariablePtr(2) = 93
  SKIPBLANK(2) = 1

  Choices$(3, 1) = "< 90        ": Choices$(3, 2) = "90 - 129    "
  Choices$(3, 3) = "> 129       "
  VariablePtr(3) = 96
  SKIPBLANK(3) = 1

  Choices$(4, 1) = "< 70        ": Choices$(4, 2) = "70 - 89     "
  Choices$(4, 3) = "> 89        "
  VariablePtr(4) = 99
  SKIPBLANK(4) = 2

  CALL DataEntryPage(exitchar$, question$(), Choices$(),
    VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
    GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
    NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
  IF exitchar$ = "P" THEN 100
```

```
     IF exitchar$ = "X" THEN 100

5500 REM Page 6; Page 2 of PE.

     NUMQUEST = 3
     NUMCOL1QUESTS = 2
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H25.TXT"
     DATAHEADING$ = "Physical-  General/Lab"
     PAGEOF$ = "2"
     OFPAGE$ = "5"
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 23)
       question$(i) = QUESTIONS$(i + 23)
       GraphFlag(i) = 0
       SKIPBLANK(i) = 1
     NEXT i


     Choices$(1, 1) = "Normal         ": Choices$(1, 2) = "Distressed
       "
     Choices$(1, 3) = "Anxious        "
     VariablePtr(1) = 102
     MULTIP(1) = 0
     None(1) = 0

     Choices$(2, 1) = "Normal         ": Choices$(2, 2) = "Pale
       "
     Choices$(2, 3) = "Flushed        ": Choices$(2, 4) = "Jaundiced
       "
     Choices$(2, 5) = "Cyanotic       "
     VariablePtr(2) = 105
     MULTIP%(2) = 1
     None(2) = 1

     Choices$(3, 1) = "< 8,000        ": Choices$(3, 2) = " 8,000 -
       10,000"
     Choices$(3, 3) = "10,100 - 12,000": Choices$(3, 4) = "12,100 -
       15,000"
     Choices$(3, 5) = "> 15,000       "
     VariablePtr(3) = 110
     MULTIP%(3) = 0
     None(3) = 0
     CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
     IF exitchar$ = "P" THEN 5000
     IF exitchar$ = "X" THEN 100


6000 REM Page 7; Page 3 of PE.

     NUMQUEST = 7
     NUMCOL1QUESTS = 4
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H35.TXT"
     DATAHEADING$ = "Physical -  Abdominal Exam"
     PAGEOF$ = "3"
     OFPAGE$ = "5"
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 26)
       question$(i) = QUESTIONS$(i + 26)
       GraphFlag(i) = 0
       MULTIP(i) = 0
       SKIPBLANK(i) = 1
     NEXT i

     Choices$(1, 1) = "Normal                 "
     Choices$(1, 2) = "Visible Perstalsis     "
     Choices$(1, 3) = "Decreased Abd. Movement"
     VariablePtr(1) = 115
     None(1) = 0

     Choices$(2, 1) = "Present                "
     Choices$(2, 2) = "Absent                 "
     VariablePtr(2) = 118
     None(2) = 0

     Choices$(3, 1) = "Present                "
     Choices$(3, 2) = "Absent                 "
     VariablePtr(3) = 120
     None(3) = 0

     Choices$(4, 1) = "Present                "
     Choices$(4, 2) = "Absent                 "
     VariablePtr(4) = 122
     None(4) = 0

     Choices$(5, 1) = "Normal                 "
     Choices$(5, 2) = "Absent                 "
     Choices$(5, 3) = "Hyperactive            "
     VariablePtr(5) = 124
     None(5) = 0

     Choices$(6, 1) = "Present                "
     Choices$(6, 2) = "Absent                 "
     VariablePtr(6) = 127
     None(6) = 0

     Choices$(7, 1) = "Present                "
     Choices$(7, 2) = "Absent                 "
     VariablePtr(7) = 129
     None(7) = 0

     CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
     IF exitchar$ = "P" THEN 5500
     IF exitchar$ = "X" THEN 100

7000 REM PAGE 8; Page 4 of PE.


     NUMQUEST = 1
     NUMCOL1QUESTS = 1
     TOPROW = 4
     TOPCOL = 19
     HELPFILE$ = "H45.TXT"
     DATAHEADING$ = "Physical Exam"
     PAGEOF$ = "4"
     OFPAGE$ = "5"

       question$(1) = QUESTIONS$(34)
       MULTIP%(1) = 1
       Numresp%(1) = QUESTPTR%(34)
       None(1) = 13
       GraphFlag(1) = 3
       SKIPBLANK(1) = 2
       Choices$(1, 1) = "   RUQ      ": Choices$(1, 2) = "   LUQ       "
       Choices$(1, 3) = "   RLQ      ": Choices$(1, 4) = "   LLQ       "
       Choices$(1, 5) = "UPPER HALF  ": Choices$(1, 6) = "LOWER HALF   "
       Choices$(1, 7) = "RIGHT HALF  ": Choices$(1, 8) = "LEFT HALF    "
       Choices$(1, 9) = "  CENTRAL   ": Choices$(1, 10) = " GENERAL    "
       Choices$(1, 11) = "RIGHT FLANK": Choices$(1, 12) = "LEFT FLANK "
       Choices$(1, 13) = " NO PAIN   "

       VariablePtr(1) = 131
       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 6000
       IF exitchar$ = "X" THEN 100


8000 REM Page 9; Last page of PE.

     NUMQUEST = 3
     NUMCOL1QUESTS = 3
     TOPROW = 4
     TOPCOL = 7
     HELPFILE$ = "H55.TXT"
     DATAHEADING$ = "Physical-  Abdominal Exam"
     PAGEOF$ = "5"
     OFPAGE$ = "5"
     FOR i = 1 TO NUMQUEST
       Numresp%(i) = QUESTPTR%(i + 34)
       question$(i) = QUESTIONS$(i + 34)
       GraphFlag(i) = 0
       SKIPBLANK(i) = 1
     NEXT i

     Choices$(1, 1) = "Present            "
     Choices$(1, 2) = "Absent             "
     VariablePtr(1) = 144
     MULTIP(1) = 0
     None(1) = 0

     Choices$(2, 1) = "Present            "
     Choices$(2, 2) = "Absent             "
     VariablePtr(2) = 146
     MULTIP(2) = 0
     None(2) = 0

     Choices$(3, 1) = "Normal              "
     Choices$(3, 2) = "Mass Felt           "
     Choices$(3, 3) = "Left Tender         "
     Choices$(3, 4) = "Right Tender        "
     Choices$(3, 5) = "General Tenderness "
     VariablePtr(3) = 148
     MULTIP(3) = 1
     None(3) = 1

     IF sex$ = FEMALE$ THEN
       NUMQUEST = 4
       VariablePtr(4) = 153
       Numresp%(4) = QUESTPTR%(4 + 34)
       question$(4) = QUESTIONS$(38)
       GraphFlag(4) = 0
       SKIPBLANK(4) = 1
       MULTIP%(4) = 1
       None(4) = 1
       Choices$(4, 1) = "Normal             "
       Choices$(4, 2) = "Right Tenderness   "
       Choices$(4, 3) = "Left  Tenderness   "
       Choices$(4, 4) = "Cervical Tenderness"
       Choices$(4, 5) = "General Tenderness "
       Choices$(4, 6) = "Mass               "
       Choices$(4, 7) = "Blood (Clots)      "
     END IF

       CALL DataEntryPage(exitchar$, question$(), Choices$(),
       VariablePtr%(), MULTIP%(), SKIPBLANK%(), Numresp%(), None%(),
       GraphFlag%(), VARIABLE%(), NUMCOL1QUESTS%, TOPROW%, TOPCOL%,
       NUMQUEST%, DATAHEADING$, PAGEOF$, OFPAGE$, HELPFILE$, STFLAG%)
       IF exitchar$ = "P" THEN 7000
       IF exitchar$ = "X" THEN 100
       IF exitchar$ = "N" THEN 100



31000 REM  Primary Selection routine
      CALL SetTrainingColors(TRAINING)
      IF TRAINING = 0 THEN GOTO 32000

31004 REM Training Option Page

      'check for training case data file
      IF NOT Exists%("ABDTRN.DAT") THEN
SCREEN 0
   CLS
LOCATE 10, 10
PRINT "The Training Module is not available."
LOCATE 11, 10
PRINT "The data file for the training cases is not present."
TRAINING = 0
CALL TextPause
GOTO 32000
      END IF
      Choices$(1, 1) = "Easy Case Narrative  "
      Choices$(1, 2) = "Harder Case Narrative"
      Choices$(1, 3) = "Enter DATA           "
      Choices$(1, 4) = "Exit Training Module "

      ' New method for menu
      NR% = 1
      resplength% = 4
      DATAHEADING$ = "Abdominal Pain Diagnosis Training Module" +
       VERSION$
      menuheading$ = "Training Options"
      HELPFILE$ = "HPT12.DAT"

      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

      '  Branch Depending on User's Selection (Case Narrative,
      '   Enter Data, Exit).
      SELECT CASE NR%
CASE 1, 2
  CALL narrative(NR, CASENUM, THECASE%())
CASE 3
  CALL ModifyNarrative(CASENUM%, escflag%)
  IF escflag <> 1 THEN
    CALL LoadTrainingCase(CASENUM, THECASE%())
    GOTO 30
  END IF
CASE 4
  TRAINING = 0
  CALL SetTrainingColors(TRAINING)
CASE ELSE
      END SELECT
      GOTO 31000


32000 REM  Primary Selection routine for main program.
32004 REM Main Option Page
      Choices$(1, 1) = "Real Case          "

      Choices$(1, 2) = "Modify Real Case   "
      Choices$(1, 3) = "Delete Real Case   "
      Choices$(1, 4) = "Exit Program       "
      resplength% = 4

      'Choices$(1, 2) = "Simulated Case     "
      'Choices$(1, 3) = "Training Module    "
      'Choices$(1, 4) = "Last Real Case     "
      'Choices$(1, 5) = "Last Simulated Case"
      'Choices$(1, 6) = "Instructions - HELP"
      'Choices$(1, 7) = "Generate SF600     "
      'Choices$(1, 8) = "Exit Program       "
      'resplength% = 8

      ' New method for menu
      NR% = 1
      DATAHEADING$ = "Abdominal Pain Diagnosis Program" + VERSION$
      menuheading$ = "Main Options"
      HELPFILE$ = "HP12.DAT"
      DO
'initialize StatWhichCase% for stat program.
StatWhichCase% = 0
CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

'             Branch to Main Option Selected.
'      ON NR GOTO 32320, 32330, 32340, 32350, 32350, 32600, 32800, 32900
SELECT CASE NR%
CASE 1                        ' Real Case
  StatWhichCase% = 0
32320     SIMULATE = 1
  GOSUB 60135
  GOTO 30
CASE 2                        ' Modify Real Case
  STFLAG = 0
  filnam$ = "REAL.DAT"
  SIMULATE = 1

  ' open case, get data
          modheading$ = "Modify Real Case"
          RealFileName$="REAL.DAT"
          CALL GetSF600Case(RealFileName$, modheading$, normalcolor%,
       backcolor%, whichcase%)

  IF whichcase% <> 0 THEN
    CALL GetCase(filnam$, whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
    StatWhichCase% = whichcase%
    '  set lookatfemale variable
    IF HMDX% > 7 THEN
      lookatfemale = 1
    END IF
    GOTO 100
  END IF


CASE 3                        ' Delete Real Case
  filnam$ = "REAL.DAT"
  SIMULATE = 1
  ' select case
  modheading$ = "Delete Real Case"
          RealFileName$="REAL.DAT"
          CALL GetSF600Case(RealFileName$, modheading$, normalcolor%,
       backcolor%, whichcase%)
```

```
IF whichcase% <> 0 THEN
  CLS
  LOCATE 10, 10, 1
  PRINT "Are your sure you want to delete case # "; whichcase%;
     "[Y/N];";
  DO
    CALL GetKey(a$)
  LOOP UNTIL INSTR("YN", a$) <> 0
  IF a$ = "Y" THEN
    'get num of cases
    delfilenum = FREEFILE
    realfile$ = "REAL.DAT"
    OPEN "R", #delfilenum, realfile$, 128
    FIELD #delfilenum, 128 AS Lstring$
    N% = LOF(1) / 128
    IF N% > 1 THEN
      'open new file tem p.$$$
      temfilenum = FREEFILE
      tempfile$ = "TEM P.$$$"
      OPEN "R", #temfilenum, tempfile$, 128
      FIELD #temfilenum, 128 AS temstring$

    'copy up to delete case
      FOR i = 1 TO whichcase% - 1
        GET #delfilenum, i
        LSET temstring$ = Lstring$
        PUT #temfilenum, i
      NEXT i
    'copy cases after delete case
      FOR i = whichcase% + 1 TO N%
        GET #delfilenum, i
        LSET temstring$ = Lstring$
        PUT #temfilenum, i - 1
      NEXT i
    'close both files
    CLOSE #temfilenum
    CLOSE #delfilenum
    'copy org real.dat to real.bak
    bakfile$ = "REALABD.BAK"
    IF Exists%(bakfile$) THEN
      ' For some bizzare reason, a file with length 0 exists.
      ' So, calling exists% will always create a file if not present.
    END IF
    KILL bakfile$
    NAME realfile$ AS bakfile$
    'ren tem p.$$$ to real.dat
    NAME tempfile$ AS realfile$
    ELSE
      'since only one case, kill it.
```

```
        CLOSE #delfilenum
        KILL realfile$
      END IF

    END IF
    GOTO 31000
  END IF

CASE 4                      'Exit Program.
32900     CLS
  SCREEN 0, 1, 0, 0
   CLS
   END
CASE ELSE

END SELECT
' loop forever.  Will break out of loop when needed.
      LOOP UNTIL 1 = 2


50000 IF STFLAG <> 0 AND TRAINING = 0 THEN
CALL SelectDatabase(lookatfemale, VARIABLE())
      END IF



'            This portion of the program calculates diagnostic
       probabilities.
      REM Initial probabilities -  DIV 10^25 to keep from getting an
       OVERFLOW ERROR.
50020 'DATA 1.8D-25,5.4D-25,0.3D-25,0.01D-25,0.5D-25,0.3D-25,1.6D-25
     PROB#(1) = 1.8D-25
     PROB#(2) = 5.4D-25
     PROB#(3) = 3D-26
     PROB#(4) = 1D-27
     PROB#(5) = 5D-26
     PROB#(6) = 3D-26
     PROB#(7) = 1.6D-25


'need: PROB#(),TFLAG,TRAINING,VARIABLE(),THECASE%(),BAYES!(),exitcode
'  have an exitcode and can remove several vars
       (TFLAGIF,TFLAGCASE,NUMCOUNT

      SCREEN 0, 1, 0, 0: CLS
      NUMCOUNT = 0
      REM All the checks for enough DATA entered can go here.
      REM Probabilities computed here
      TFLAG = 0: TFLAGCASE = 0: TFLAGIF = 0
      FOR i = 1 TO 152
IF TRAINING = 1 THEN
  IF THECASE%(i) = 1 THEN
    TFLAGCASE = TFLAGCASE + 1
    IF VARIABLE%(i) = THECASE%(i) THEN
      TFLAGIF = TFLAGIF + 1
    END IF
  END IF
  IF VARIABLE%(i) = THECASE%(i) THEN
    TFLAG = TFLAG + 1
  END IF
END IF
IF VARIABLE%(i) = 1 THEN
  NUMCOUNT = NUMCOUNT + 1
  FOR j = 1 TO 7
    PROB#(j) = PROB#(j) * BAYES!(i, j)
  NEXT j
END IF
      NEXT i

51025 IF TRAINING = 1 THEN
 IF TFLAGIF > .75 * TFLAGCASE THEN
   GOTO 51040
  ELSE
   GOTO 51036
 END IF
       END IF
'NOTE NOTE !!!!!!!!!!!!! the next line is just for Ellen's version for
'                        entering data.
'      IF davidflag% = 1 THEN GOTO 51040
       GOTO 51040

      IF NUMCOUNT > 32 THEN 51040
      PRINT "Insufficient DATA has been entered for accurate diagnosis."
      PRINT "Please enter more DATA."
      GOTO 51038
51036 PRINT "You have missed too many items.  Are you sure you have the
       right case?"
51038 LOCATE 25, 15
      CALL SetColor(infocolor)
      PRINT " To return to main menu, press any key";
      CALL SetColor(infocolor)
      CALL GetKey(a$)
      GOTO 100


'            Calculate final probabilitiies here (FINPROB).  Determine
' the disease (MAXNUM) with the greatest probability (MAXPROB).
```

```
51040 CALL ComputeFinalProbs(MAXNUM%, MAXPROB%, PROB#(), FINPROB#())

      FINPROB#(2) = FINPROB#(2) + FINPROB#(7)
      IF MAXNUM = 7 THEN MAXNUM = 2
      IF FINPROB#(2) > MAXPROB THEN MAXNUM = 2
      IF MAXNUM = 2 THEN MAXPROB = FINPROB#(2)
      MALEMAXNUM% = MAXNUM

      'compute GYN probabilities.
      'skip all of the stuff about training checks and number of
       questions
      ' answered.
     REM Initial probabilities -  DIV 10^25 to keep from getting an
       OVERFLOW ERROR.
     '  fOR NOW, BOGUS PROBABILITIES.
     FEMPROB#(1) = 1.8D-25
     FEMPROB#(2) = 5.4D-25
     FEMPROB#(3) = 3D-26
     FEMPROB#(4) = 1D-27
     FEMPROB#(5) = 5D-26
     FEMPROB#(6) = 3D-26
     FEMPROB#(7) = 1.6D-25


      IF sex$ = FEMALE$ THEN
FOR i = 1 TO 174
  IF VARIABLE(i) = 1 THEN
    FOR j = 1 TO 7
      IF FEMBAYES!(i, j) <> 0 THEN
        FEMPROB#(j) = FEMPROB#(j) * FEMBAYES!(i, j)
      END IF
    NEXT j
  END IF
NEXT i

        CALL ComputeFinalProbs(FEMMAXNUM%, FEMMAXPROB%, FEMPROB#(),
       FEMFINPROB#())
FEMMAXNUM% = FEMMAXNUM% + 7
      END IF

      REM PAGE 0 -- HM EVALUATION
'            Skip this page if the case is not new of if no response
' changes have been made.
      'stat change.  Always come thru HM dx page.
      'IF STFLAG = 0 THEN 52000


      'How about here is where the HM is asked re which database to use.
      'if female and appropriate responses marked, then
      ' ask which database to use.
      'On this page, can list all of the disease considered.
      'Will then need to move agree/disagree part to next screen.

      'Can have toggle on diagnostic summary page to switch back and
       forth
      ' between abd and gyn databases.



      IF lookatfemale = 0 THEN
Choices$(1, 1) = "APPENDICITIS               "
Choices$(1, 2) = "NON-SPECIFIC ABDOMINAL PAIN"
Choices$(1, 3) = "RENAL COLIC                "
Choices$(1, 4) = "PERFORATED DUODENAL ULCER  "
Choices$(1, 5) = "CHOLECYSTITIS              "
Choices$(1, 6) = "SMALL BOWEL OBSTRUCTION    "
Choices$(1, 7) = "OTHER                      "
resplength% = 7
menuheading$ = "Your Diagnosis"
      ELSE
Choices$(1, 1) = "APPENDICITIS               "  'dx #  8
Choices$(1, 2) = "NON-SPECIFIC ABDOMINAL PAIN"  'dx #  9
Choices$(1, 3) = "PELVIC INFLAMMATORY DISEASE"  'dx # 10
Choices$(1, 4) = "URINARY TRACT INFECTION    "  'dx # 11
Choices$(1, 5) = "OVARIAN CYST               "  'dx # 12
Choices$(1, 6) = "ECTOPIC PREGNANCY          "  'dx # 13
Choices$(1, 7) = "INCOMPLETE ABORTION        "  'dx # 14
Choices$(1, 8) = "OTHER                      "  'dx # 15
resplength% = 8
menuheading$ = "Your Gynecological Diagnosis"
      END IF

      ' New method for menu
      '-------------stat routine begins---------------------------

      IF HMDX > 7 THEN
NR% = HMDX - 7
      ELSEIF HMDX = 0 THEN
NR% = 1
      ELSE
NR% = HMDX
      END IF

      '-------------stat routine ends-----------------------------

      DATAHEADING$ = "Corpsman's Diagnosis Entry Page"
      HELPFILE$ = "HP00.DAT"

      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)

'            Store Corpsman's Diagnosis variable HMDX (# of the
' Diagnosis) and HMDX$ (name of the diagnosis).
      IF lookatfemale = 0 THEN
HMDX = NR
      ELSE
HMDX = NR + 7
      END IF

      IF HMDX = 7 OR HMDX = 15 THEN
       'stat change
51250  ' DO
  LOCATE 16, 1
  PRINT "HM's OTHER diagnosis: "; OTHER$
  LOCATE 17, 1: PRINT SPACE$(75);
  LOCATE 17, 1
  PRINT "Enter name of other diagnosis (CR if no change): ";
  LINE INPUT tempOTHER$
  IF tempOTHER$ <> "" THEN
    OTHER$ = tempOTHER$
  END IF
' LOOP UNTIL OTHER$ <> ""
LOCATE 19, 10
PRINT "This database does not consider "; OTHER$;
LOCATE 20, 10
PRINT "in the differential diagnosis of acute abdominal pain."
CALL TextDxPause
GOTO 52000
      END IF

      ' stat line
      GOTO 52000

51350 IF Translate%(HMDX) = MAXNUM THEN
LOCATE 16, 1
PRINT "   The program-generated probabilities AGREE with your
       provisional";
LOCATE 17, 1: PRINT "diagnosis.";
CALL TextDxPause
GOTO 52000
      END IF

      ' Put all of this on another page.

      ' Check for questions to recheck if original database selected.
      IF lookatfemale = 0 THEN
CALL CompareAbdDXes(COMPAR%(), VARIABLE%(), MAXNUM%, HMDX%, QUESTPTR%(),
       QUESTIONS$())
      ELSE
CALL CompareFemDXes(HMDX%, MAXNUM%)
      END IF

52000 REM PAGE 14 -- Diagnostic Summary Page
52080 CALL SetScreenMode(ScrnMode)
      LOCATE 1, 18: headingPRINT ("Diagnostic Summary Page")
      LOCATE 1, 59: PRINT "SSN:  "; SSN$;
      LOCATE 2, 59: PRINT "Time: "; STARTIME$;
      LOCATE 3, 59: PRINT "Date: "; STARTDATE$;
      IF TRAINING = 1 THEN
LOCATE 5, 59: PRINT "Score:"; INT((TFLAG / 152) * 100); "%";
IF TFLAG = 152 THEN PRINT "!!!";
SELECT CASE HMDX
  CASE 1
   HMChosenDX$ = "APPEND"
  CASE 2
   HMChosenDX$ = "NONSAP"
  CASE 3
   HMChosenDX$ = "RENCOL"
  CASE 4
   HMChosenDX$ = "PERFDU"
  CASE 5
   HMChosenDX$ = "CHOLE"
  CASE 6
   HMChosenDX$ = "SMBOBS"
  CASE ELSE
   HMChosenDX$ = OTHER$
END SELECT
LOCATE 6, 59: PRINT "HM dx: "; HMChosenDX$;
LOCATE 17, 59: PRINT "TRAINING CASE";
LOCATE 18, 64: PRINT "#"; CASENUM;
      ELSE
LOCATE 17, 59
IF SIMULATE = 0 THEN
  PRINT "SIMULATED CASE";
  ELSE
  PRINT "REAL CASE";
END IF
      END IF
      LOCATE 20, 59: PRINT BOAT1$;
      LOCATE 21, 59: PRINT BOAT2$;

      IF Vertbits = 14 THEN
WINDOW SCREEN (0, 0)-(639, 199)
      END IF
      IF sex$ = FEMALE$ AND lookatfemale = 1 THEN
MAXNUM = FEMMAXNUM
CALL FemaleGraph(FEMFINPROB#())
      ELSE
MAXNUM = MALEMAXNUM
CALL MaleGraph(FINPROB#())
      END IF



'      LINE (0, 150)-(449, 150), hpframecolor
'      LINE (0, 28)-(449, 28), hpframecolor, , &HF00F
'      LINE (0, 82)-(449, 82), hpframecolor, , &HF00F


      IF TRAINING = 0 THEN
resplength% = 5
HELPFILE$ = "HP10.DAT"
Choices$(1, 1) = "CHANGE INPUT DATA"
Choices$(1, 2) = "ANOTHER DIAGNOSIS"
Choices$(1, 3) = "DISPLAY TREATMENT"
Choices$(1, 4) = "DISPLAY H & P    "
Choices$(1, 5) = "END INTERACTION  "
IF sex$ = FEMALE$ THEN
  resplength% = resplength% + 1
  IF lookatfemale = 1 THEN
    Choices$(1, resplength%) = "DISPLAY ABD DX'ES"
  ELSE
    Choices$(1, resplength%) = "DISPLAY GYN DX'ES"
  END IF
END IF
      ELSE
resplength% = 6
HELPFILE$ = "HPT10.DAT"
Choices$(1, 1) = "CHANGE INPUT DATA"
Choices$(1, 2) = "ANOTHER CASE     "
Choices$(1, 3) = "DISPLAY TREATMENT"
Choices$(1, 4) = "DISPLAY H & P    "
Choices$(1, 5) = "END INTERACTION  "
Choices$(1, 6) = "SHOW MISSED ITEMS"

      END IF


      NR% = 1
      menuheading$ = "Options"

    DO
```

```
      CALL MenuSummaryPage(10, 59, NR%, resplength%, exitchar$,
       menuheading$, Choices$(), HELPFILE$)
      ' Help text is written in text mode.  Have to redraw whole screen.
      IF exitchar$ = "?" THEN EXIT DO

' for testing only
IF davidflag% = 1 THEN
  LOCATE 1, 1
  PRINT USING "###### ###### ######"; FRE(0), FRE(""), FRE(-1)
END IF


'52440 ON NR GOTO 52450, 52501, 54000, 53000, 52500, 55000
      SELECT CASE NR%
CASE 1                   ' Change Input Data
  ' save it only if real case and changes have been made
  IF (TRAINING = 0 AND STFLAG = 1 AND SIMULATE = 1) THEN
    filnam$ = "REAL.DAT"
    StatWhichCase% = whichcase%
    CALL PutCase(whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
    STFLAG = 0
  END IF
    GOTO 100

CASE 2                   ' Another Dx/Case
  IF TRAINING = 0 AND STFLAG = 1 THEN
    ' Save both real and simulated cases,
    '  if not training, and changes were made.
    StatWhichCase% = whichcase%
    CALL PutCase(whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
  END IF
  CALL ResetVariables(VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
  STFLAG = 0
  lookatfemale = 0       'may not be necessary
  GOTO 31000

CASE 3                   ' Display Treatment
  '  User selects option to Display Treatment Protocol.
  CALL TXMenu(MAXNUM, sex$)
   CLS
  GOTO 52000

CASE 4                   ' Display H&P
  CALL DisplayHP(TRAINING, SIMULATE, SSN$, STARTIME$, STARTDATE$,
       VARIABLE%())
  GOTO 52000
```

```
CASE 5                  ' End
  IF TRAINING = 0 AND STFLAG = 1 THEN
    'save main real or simulated cases if changes made.
    CALL PutCase(StatWhichCase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
  END IF
  SCREEN 0, 1, 0, 0
   CLS
   END

CASE 6                  ' Show Missed Items
  IF TRAINING = 1 THEN
    CALL DisplayMissedHP(SSN$, STARTIME$, STARTDATE$, VARIABLE%(),
       THECASE%())
    GOTO 52000
  ELSE                  ' Should only get here with real case if female
    IF lookatfemale = 1 THEN
      CALL MaleGraph(FINPROB#())
      Choices$(1, resplength%) = "DISPLAY GYN DX'ES"
      MAXNUM = MALEMAXNUM
      lookatfemale = 0
    ELSE
      CALL FemaleGraph(FEMFINPROB#())
      Choices$(1, resplength%) = "DISPLAY ABD DX'ES"
      MAXNUM = FEMMAXNUM
      lookatfemale = 1
    END IF
  END IF
CASE ELSE
      END SELECT
    LOOP UNTIL NR% <> 6 AND sex$ <> FEMALE$
    GOTO 52000

60000 REM READ IN BAYESIAN DATA
      CALL UnPackDatabase("REGABD.DAT", BAYES!(), 7, 152)

      CALL UnPackDatabase("FEMABD.DAT", FEMBAYES!(), 7, 174)

      RESTORE 60300
      FOR i = 1 TO 44: READ QUESTIONS$(i): NEXT i
      RESTORE 60410
      FOR i = 1 TO 44: READ QUESTPTR%(i): NEXT i
      RESTORE 60500
      FOR i = 1 TO 6
FOR j = 1 TO 6
  IF i <> j THEN
    FOR N = 1 TO 15
      READ COMPAR%(i, j, N)
      IF COMPAR%(i, j, N) < 1 THEN EXIT FOR
    NEXT N
  END IF
NEXT j
      NEXT i

'  VARIABLE(I) = 0 when the response has not been entered
' and VARIABLE(I) = 1 when the response has been entered.
60135 CALL ResetVariables(VARIABLE%(), sex$, SSN$, AGE$, STARTIME$,
       STARTDATE$)
      RETURN

      'QUESTIONS$()
60300 DATA SITE OF PAIN(onset),SITE OF PAIN(now)
      DATA TYPE OF PAIN,SEVERITY OF PAIN,AGGRAVATING FACTORS
      DATA PROGRESS OF PAIN,DURATION OF PAIN,RELIEVING FACTORS
      DATA NAUSEA,VOMITING,BOWELS,APPETITE,JAUNDICE,URINATION
      DATA PREVIOUS INDIGESTION,PREVIOUS SIMILAR PAIN,PREVIOUS SURGERY
      DATA PREVIOUS ILLNESS,TAKING MEDICATIONS
      DATA "TEMPERATURE (F):","PULSE RATE:","BP (Systolic):","BP
       (Diastolic):"
      DATA "MOOD:","COLOR:","WBC COUNT:"
      DATA "INSPECTION:","SCARS:","GUARDING:","RIGIDITY:","BOWEL
       SOUNDS:"
      DATA "DISTENTION:","MASSES:"
      DATA TENDERNESS,"MURPHY'S SIGN:",REBOUND TENDERNESS,RECTAL EXAM
      DATA "VAGINAL EXAMINATION"
      DATA PERIODS,"LAST PERIOD","VAGINAL DISCHARGE",PREGNANCY
      DATA "FAINT/DIZZY","PREV HISTORY"
      'QUESTPTR%()
60410 DATA 13,13,3,2,6,3,4,6,2,2,5,2,2,5,2,2,2,2,2,4,3,3,3
      DATA 3,5,5,3,2,2,2,3,2,2,13,2,2,5,7,4,2,2,3,2,2
      'COMPAR%()
60500 DATA 13,19,26,32,38,42,43,50,70,107,120,133,146,151,0
      DATA 26,38,40,42,43,74,107,120,133,146,151,0
      DATA 19,26,40,102,107,115,133,151,0
      DATA 26,40,42,43,107,133,145,146,151,0
      DATA 26,38,40,42,43,65,102,107,120,133,146,151,0
      DATA 37,47,71,105,121,147,148,0
      DATA 40,74,0
      DATA 37,40,48,102,105,115,148,0
      DATA 37,40,65,145,133,0
      DATA 48,65,102,0
      DATA 21,22,34,35,39,41,47,75,76,105,106,107,141,142,147
      DATA 15,19,20,28,41,79,100,121,135,147,148,0
      DATA 21,22,34,35,39,75,76,102,105,115,128,0
      DATA 21,22,34,35,75,76,141,142,145,0
      DATA 21,22,34,35,41,65,75,76,128,140,0
      DATA 11,15,24,28,41,82,106,131,144,148,0
```

```
   DATA 32,33,39,41,47,66,83,103,106,121,127,140,147,148,0
   DATA 21,22,34,35,41,75,76,141,142,0
   DATA 15,20,28,33,42,55,103,106,117,140,145,0
   DATA 15,20,28,33,38,41,65,117,0
   DATA 15,20,28,33,38,0
   DATA 11,24,38,41,63,66,79,120,131,144,0
   DATA 19,32,39,49,66,83,103,106,127,139,0
   DATA 15,20,28,39,74,104,135,0
   DATA 11,24,38,107,131,144,0
   DATA 11,15,24,28,74,81,120,131,144,0
   DATA 19,32,105,115,135,148,0
   DATA 11,24,102,105,115,131,144,0
   DATA 19,32,39,66,80,83,103,127,140,0
   DATA 19,32,39,83,103,127,140,145,0


REM $STATIC
SUB CompareFemDXes (HMDX%, MAXNUM%)
'   This routine displays the Best Questions to Ask if the HM's
'   diagnosis differs from the computer's (GYN database only).
'   See CompareAbdDXes for abd database.

  CLS
  TheHMDX = HMDX - 7
  row = 10
  col = 10

  filnam$ = "bestques.dat"
  IF NOT Exists%(filnam$) THEN
    EXIT SUB
  END IF

  ' draw window here.

  IF TheHMDX = MAXNUM THEN
    'the dx'es agree.  already commented on in ABDX.BAS
  ELSEIF TheHMDX = 8 THEN
    'other dx not considered.  already commented on in ABDX.BAS
  ELSE
    'create a number such that the smaller of TheHMDX or MAXNUM is in
    'the tens' position and the other is in the unit's position.
    IF TheHMDX < MAXNUM THEN
combinednumber = TheHMDX * 10 + MAXNUM
    ELSE
combinednumber = MAXNUM * 10 + TheHMDX
    END IF


    filenum = FREEFILE
```

```
      OPEN filnam$ FOR INPUT AS filenum


      foundflag = 0
      stringnumber = 0
      DO WHILE stringnumber < combinednumber
LINE INPUT #filenum, quest$
'decrypt string
CALL decipher(quest$)
stringnumber = VAL(LEFT$(quest$, 2))
      LOOP
      IF stringnumber = combinednumber THEN
numofquests = VAL(MID$(quest$, 4, 2))
questlength = VAL(MID$(quest$, 7, 2))
col = (80 - questlength) \ 2
foundflag = 1
LOCATE 3, 5: PRINT "  At this time the computer-generated probabilities
       DO NOT AGREE with";
LOCATE 4, 5: PRINT "your preliminary diagnosis.  The following
       categories are particularly";
LOCATE 5, 5: PRINT "useful in differentiating your diagnosis from the
       others.  It may be";
LOCATE 6, 5: PRINT "helpful to review your input in these areas and make
       any changes you";
LOCATE 7, 5: PRINT "consider appropriate.";

    DATAHEADING$ = "Best Questions to Ask"
    CALL LocateCenter(1, DATAHEADING$)
    CALL headingPRINT(DATAHEADING$)

'        CALL frame(row , col - 1, numofquests, questlength, frametyp)
CALL BoxSelections(row, col - 1, numofquests, questlength)

FOR i = 1 TO numofquests
  LINE INPUT #filenum, quest$
  'decrypt string
  CALL decipher(quest$)
  LOCATE row, col
  PRINT quest$
  row = row + 1
NEXT i
      END IF
      CLOSE #filenum
      IF foundflag = 0 THEN
LOCATE row, col
PRINT "    At this time, the computer-generated probabilities DO NOT
       AGREE with";
PRINT "your preliminary diagnosis.  However, in this case, there are
       no";
```

```
PRINT "specific questions to ask which would differentiate your
       preliminary";
PRINT "diagnosis from the current program-generated diagnosis.";

      END IF
      CALL TextPause
    END IF
END SUB

SUB experimental
'       This sub states that the progrm is experimental and is not
'       for distribution.
CLS
LOCATE 10, 10
PRINT "This program is EXPERIMENTAL and is not for distribution."
CALL TextPause
END SUB

SUB GetCase (filnam$, whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$,
       STARTIME$, STARTDATE$, HMDX%, SIMULATE%, sex$)
'       This routine opens the .DAT file filnam$ and retrieves the
'       appropriate case 'whichcase'.
'       NOTE:  if whichcase = 0 then the case retrieved is the last
'               stored case.
'              If it is returned as 0, then file did not exist.

      'Globals
      '       - MALE$   (constant in include file )
      '       - FEMALE$ (constant in include file )


      OPEN filnam$ FOR RANDOM AS #1 LEN = 128
      '  File format for variables.
      FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$
'  If no previous case has been entered, beep, close
' the file and request more user input.
N% = LOF(1) / 128
IF N% = 0 THEN
  SOUND 150, 4
  LOCATE 15, 10
  PRINT "No previous case stored.";
  CLOSE #1
  whichcase% = 0
  EXIT SUB
END IF
IF whichcase% = 0 OR whichcase% > N% THEN
  whichcase% = N%
```

```
END IF

'  Get a record from the file.
GET #1, whichcase%

'   Put case data into variables.
SSN$ = LSSN$: AGE$ = LAGE$: a$ = LVAR$: OTHER$ = LOTH$
STARTIME$ = LTIM$: STARTDATE$ = LDAT$
HMDX = CVI(LHMD$): SIMULATE = CVI(LSIM$)
'   Close the file.
      CLOSE #1
      ' unpack data in a$ into VARIABLE%()
      CALL UnPackArray(a$, VARIABLE%())
      ' update sex
      IF VARIABLE%(2) = 1 THEN
sex$ = FEMALE$
      ELSE
sex$ = MALE$
      END IF


END SUB

REM $DYNAMIC
SUB InitializeColors (graphmode$, monmode$)

'GLOBAL - GRAPHICS, ScrnMode, forecolor, backcolor, infocolor
'GLOBAL - textcolor, questioncolor, responsecolor, hpresponsecolor
'GLOBAL - graphcolor, helpcolor, hpframecolor, bargraph()
'GLOBAL - red,green, brown, white, black, yellow
'GLOBAL - ltred, ltgreen, ltcyan, ltmagenta, ltblue

  SHARED mainheadingcolor, mainframecolor, mainframe
  SHARED trainingheadingcolor, trainingframecolor, trainingframe

'       Set up graphics mode default (checked for CGA or EGA in
       graphmode$)
IF graphmode$ = "C" THEN
  GRAPHICS = 2
  ScrnMode = 2
  IF monmode$ = "C" THEN
    '  CGA and color monitor
    mainheadingcolor = red
    mainframecolor = green
    mainframe = 1            'select single frame for default pages.
    trainingheadingcolor = brown
    trainingframecolor = 1
    trainingframe = 2            'select double frame for training
       pages.
  forecolor = white
  backcolor = black
  textcolor = white
  questioncolor = yellow
  responsecolor = white
  hpresponsecolor = white
  hpframecolor = 1
  infocolor = green
  graphcolor = white
  helpcolor = white
  bargraph(1) = 1
  bargraph(2) = 1
  bargraph(3) = 1
  bargraph(4) = 1
  bargraph(5) = 1
  bargraph(6) = 1
  bargraph(7) = 1
ELSE
  '   CGA, but no color monitor
  mainheadingcolor = white
  mainframecolor = white
  mainframe = 1              'select single frame for default pages.
  trainingheadingcolor = white
  trainingframecolor = white
  trainingframe = 2              'select double frame for training
     pages.
  forecolor = white
  backcolor = black
  textcolor = white
  questioncolor = white
  responsecolor = white
  hpresponsecolor = white
  hpframecolor = 1
  infocolor = white
  graphcolor = white
  helpcolor = white
  bargraph(1) = 1
  bargraph(2) = 1
  bargraph(3) = 1
  bargraph(4) = 1
  bargraph(5) = 1
  bargraph(6) = 1
  bargraph(7) = 1
END IF
ELSE
GRAPHICS = 9
ScrnMode = 9
IF monmode$ = "C" THEN
  '  EGA and color monitor
```

```
    mainheadingcolor = red
    mainframecolor = green
    mainframe = 1            'select single frame for default pages.
    trainingheadingcolor = brown
    trainingframecolor = brown
    trainingframe = 2            'select double frame for training
       pages.
    forecolor = white
    backcolor = black
    textcolor = white
    questioncolor = yellow
    responsecolor = white
    hpresponsecolor = white
    hpframecolor = blue
    infocolor = green
    graphcolor = white
    helpcolor = white
    bargraph(1) = ltred
    bargraph(2) = ltgreen
    bargraph(3) = ltcyan
    bargraph(4) = ltmagenta
    bargraph(5) = ltblue
    bargraph(6) = yellow
    bargraph(7) = red
  ELSE
    '   EGA, but no color monitor
    mainheadingcolor = white
    mainframecolor = white
    mainframe = 1            'select single frame for default pages.
    trainingheadingcolor = white
    trainingframecolor = white
    trainingframe = 2            'select double frame for training
       pages.
    forecolor = white
    backcolor = black
    textcolor = white
    questioncolor = white
    responsecolor = white
    hpresponsecolor = white
    hpframecolor = white
    infocolor = white
    graphcolor = white
    helpcolor = white
    bargraph(1) = white
    bargraph(2) = white
    bargraph(3) = white
    bargraph(4) = white
    bargraph(5) = white
    bargraph(6) = white
    bargraph(7) = white
  END IF
END IF

END SUB

REM $STATIC
SUB PutCase (whichcase%, VARIABLE%(), SSN$, AGE$, OTHER$, STARTIME$,
       STARTDATE$, HMDX%, SIMULATE%, MAXNUM%, MAXPROB%)
'       This routine opens the .DAT file based on SIMULATE and saves the
'       appropriate case in record 'whichcase'.
'       NOTE:  if whichcase = 0 then the case is appended to the
'               stored cases.

      IF SIMULATE = 0 THEN
filnam$ = "SIMUL.DAT"
      ELSE
filnam$ = "REAL.DAT"
      END IF

      OPEN filnam$ FOR RANDOM AS #1 LEN = 128
      '  File format for variables.
      FIELD #1, 11 AS LSSN$, 2 AS LAGE$, 26 AS LVAR$, 40 AS LOTH$, 5 AS
       LTIM$, 10 AS LDAT$, 2 AS LHMD$, 2 AS LSIM$, 2 AS LNUM$, 2 AS
       LPRO$

N% = LOF(1) / 128
' If 0, then append case to end of stored cases.
IF whichcase% = 0 THEN
  whichcase% = N% + 1
END IF

CALL PackArray(a$, VARIABLE%())

'   Left justifies the variables in the field and moves the
' DATA into a random buffer file.
LSET LSSN$ = SSN$: LSET LAGE$ = AGE$: LSET LVAR$ = a$
LSET LOTH$ = OTHER$: LSET LTIM$ = STARTIME$
LSET LDAT$ = STARTDATE$: LSET LHMD$ = MKI$(HMDX)
LSET LSIM$ = MKI$(SIMULATE): LSET LNUM$ = MKI$(MAXNUM)
LSET LPRO$ = MKI$(MAXPROB)

'  Save the record.
PUT #1, whichcase%

      CLOSE #1

END SUB
```

```
 SUB SelectDatabase (lookatfemale, VARIABLE())
      'Check if female and ask which database if appropriate responses
      ' have been entered.

      'GLOBAL - frametype
      '       - FEMALE$ (constant in include file )

      SHARED sex$

      DIM checkvarlocation(6), Choices$(1, 2)

       lookatfemale = 0        'ABD database as a default for females
                 'not meeting criteria and all males.

      IF sex$ = FEMALE$ THEN

checkvarlocation(1) = 32  ' Site of Pain at present - CENTRAL
checkvarlocation(2) = 29  ' Site of Pain at present - LOWER
checkvarlocation(3) = 26  ' Site of Pain at present - RLQ
checkvarlocation(4) = 27  ' Site of Pain at present - LLQ
checkvarlocation(5) = 34  ' Site of Pain at present - R FLANK
checkvarlocation(6) = 35  ' Site of Pain at present - L FLANK
CriteriaMet = 0
FOR i = 1 TO 6
  IF VARIABLE(checkvarlocation(i)) = 1 THEN
    CriteriaMet = 1
    EXIT FOR
  END IF
NEXT i
SCREEN 0, 1, 0, 0
   CLS
framtyp = frametype
IF CriteriaMet = 1 THEN         'female and gyn database looks good.
  CALL frame(7, 9, 3, 60, framtyp)
  LOCATE 8, 10
  PRINT "    Remember the gynacological database is available and is";
  LOCATE 9, 10
  PRINT "especially suitable for this patient.  It compares your case";
  LOCATE 10, 10
  PRINT "with 1,000 similar previous patients.";
  CALL TextContinuePrompt
  CALL GetKey(a$)
  menuheading$ = "Select Database"
  Choices$(1, 1) = "General Abdominal Database"
  Choices$(1, 2) = "Gynecological Database    "
  HELPFILE$ = "Habgy.dat"
  menurow = 16
  menucol = Centered(Choices$(1, 1))
  NR% = 2
  resplength% = 2
  LOCATE 25, 5
  PRINT SPACE$(70);
    DO
    CALL MenuSummaryPage(menurow, menucol, NR%, resplength%, exitchar$,
       menuheading$, Choices$(), HELPFILE$)
  LOOP UNTIL exitchar$ = ""
  PRINT "Choose desired database"
  IF NR% = 1 THEN
    lookatfemale = 0
  ELSE
    lookatfemale = 1
  END IF

ELSE            'female, but abd database looks better.
  CALL frame(9, 9, 4, 60, framtyp)
  LOCATE 10, 10
  PRINT "    At this time, your patient appears to have";
  LOCATE 11, 10
  PRINT "non-gynecological abdominal pain.  However, the";
  LOCATE 12, 10
  PRINT "gynecological database is available as an option";
  LOCATE 13, 10
  PRINT "on the Diagnostic Summary Page.";
  CALL TextContinuePrompt
  CALL GetKey(a$)
END IF
      END IF

      ERASE checkvarlocation, Choices$

END SUB

REM $DYNAMIC
SUB SetTrainingColors (TRAINING)
'       This routine sets the different display colors between training
'        and other displays.

   ' Global vars  - headingcolor, framecolor, frametype

   SHARED mainheadingcolor%, mainframecolor%, mainframe%
   SHARED trainingheadingcolor%, trainingframecolor%, trainingframe%


    IF TRAINING = 0 THEN
      headingcolor% = mainheadingcolor
      framecolor% = mainframecolor
      frametype = mainframe
   ELSE
```

```
      headingcolor% = trainingheadingcolor
      framecolor% = trainingframecolor
      frametype = trainingframe
   END IF

END SUB

REM $STATIC
SUB STATSexSSNAgeDate (STFLAG, TRAINING, SIMULATE, sex$, SSN$, AGE$,
       STARTDATE$, STARTIME$, VARIABLE%())
'       This routine allows input for sex, ssn, age, date,and time.

'init sex
  IF VARIABLE(1) = 1 THEN
    sex$ = "M"
  ELSEIF VARIABLE(2) = 1 THEN
    sex$ = "F"
  ELSE
    sex$ = "_"
  END IF

'init time and date
 IF STARTIME$ = "" THEN STARTIME$ = LEFT$(TIME$, 5)
 IF STARTDATE$ = "" THEN STARTDATE$ = DATE$

  tempsex$ = sex$
  tempssn$ = SSN$
  tempage$ = AGE$
  tempdate$ = STARTDATE$
  temptime$ = STARTIME$

  SCREEN 0, 1, 0, 0
  CLS

  blankstring$ = "_"
  pointtoinput = 0
  finishpage = 0
  mainrow = 6
  maincol = 26
  errorcode = 0

  SexPageHeading1$ = "Sex / SSN / Age / Date / Time"
  SexPageHeading2$ = "Data Entry Page"
  SexHeading$ = "Patient's Sex [ ]"
  SSNHeading$ = "Patient's SSN [           ]"
  AgeHeading$ = "Patient's age [  ]"
  DateHeading$ = "Date of exam  [          ]"
  TimeHeading$ = "Time of exam  [     ]"
  Sexhelp$ - "Enter 'M' for male or 'F' for female.|This must be
       answered."
  Agehelp$ - "Enter the age of the patient ( between 0 and 99)."
  sexerror$ - " Only M or F accepted."
  ssnerror$ - "Only numbers accepted for the SSN."
  ageerror$ - "Invalid age.  Must be between 0 and 99, inclusive."
  dateerror$ - "Invalid date.  Use format MM-DD-YYYY"
  timeerror$ - "Invalid time.  Use 24 hour format."

  IF TRAINING - 1 OR SIMULATE - 0 THEN
'        Modify helpstrings if either in training mode or simulated
       mode.
'        I know that SIMULATE should equal 1 for consistency, but it
       doesn't.
     SSNhelp$ - "A Social Security number has been chosen for you.|There
       is no need to change it.|With a REAL case, you would enter the
       patient's SSN here."
     Datehelp$ - "Today's date has been chosen for you.|There is no need
       to change it.|With a real CASE, you would enter the date of the
       exam here."
     Timehelp$ - "The current time has been chosen for you.|There is no
       need to change it.|With a real CASE, you would enter the time of
       the exam here."

   ELSE
     SSNhelp$ - "Enter the Social Security Number.|The hyphens will be
       added automatically.|A numeric SSN must be present to continue."
     Datehelp$ - "The current date has been chosen.|If this is
       incorrect, change it."
     Timehelp$ - "The current time has been chosen.|If this is
       incorrect, change it."
   END IF

'page heading
      CALL LocateCenter(2, SexPageHeading1$)
      headingPRINT (SexPageHeading1$)
      CALL LocateCenter(3, SexPageHeading2$)
      headingPRINT (SexPageHeading2$)
'frame data
      CALL SetFrameColor
      framtyp - frametype
      CALL frame(mainrow - 1, maincol - 2, 9, 28, framtyp)
      CALL SetNormalColor

'initially show question
LOCATE mainrow, maincol
questionPRINT (SexHeading$)
LOCATE mainrow + 2, maincol
questionPRINT (SSNHeading$)
```

```
LOCATE mainrow + 4, maincol
questionPRINT (AgeHeading$)
LOCATE mainrow + 6, maincol
questionPRINT (DateHeading$)
LOCATE mainrow + 8, maincol
questionPRINT (TimeHeading$)

LOCATE mainrow, maincol + 15
PRINT sex$;
LOCATE mainrow + 2, maincol + 15
PRINT SSN$;
LOCATE mainrow + 4, maincol + 15
PRINT AGE$;
LOCATE mainrow + 6, maincol + 15
PRINT STARTDATE$;
LOCATE mainrow + 8, maincol + 15
PRINT STARTIME$;

'               finishpage = 0    cycle thru for more input
'               finishpage = 1    exit sex/ssn/etc page

  DO
    SELECT CASE pointtoinput
      CASE 0
'male/female stuff
LOCATE mainrow, maincol + 15
CALL templatehelp(Sexhelp$, tempsex$, "M", blankstring$, rc, sexerror$,
       errorcode)
'               rc = 0   CR
'                    1   Esc
'                    2   up arrow
'                    3   down arrow
IF rc = 1 THEN
  finishpage = 1
ELSEIF rc = 0 OR rc = 3 THEN
  pointtoinput = 1
END IF

      CASE 1
'ssn stuff
LOCATE mainrow + 2, maincol + 15
CALL templatehelp(SSNhelp$, tempssn$, "###-##-####", blankstring$, rc,
       ssnerror$, errorcode)
IF rc = 1 THEN
  finishpage = 1
ELSEIF rc = 0 OR rc = 3 THEN
  pointtoinput = 2
ELSEIF rc = 2 THEN
  pointtoinput = 0
```

```
END IF

      CASE 2
'age stuff
LOCATE mainrow + 4, maincol + 15
CALL templatehelp(Agehelp$, tempage$, "##", blankstring$, rc, ageerror$,
       errorcode)
IF rc = 1 THEN
  finishpage = 1
ELSEIF rc = 0 OR rc = 3 THEN
  pointtoinput = 3
ELSEIF rc = 2 THEN
  pointtoinput = 1
END IF
'Check for valid age >=1 and <=99
valage = VAL(tempage$)
IF valage < 0 OR valage > 99 THEN
  SOUND 900, 1
  finishpage = 0
  pointtoinput = 2
  errorcode = 1
END IF

      CASE 3
'date stuff
LOCATE mainrow + 6, maincol + 15
CALL templatehelp(Datehelp$, tempdate$, "##-##-####", blankstring$, rc,
       dateerror$, errorcode)
IF rc = 1 THEN
  finishpage = 1
ELSEIF rc = 0 OR rc = 3 THEN
  pointtoinput = 4
ELSEIF rc = 2 THEN
  pointtoinput = 2
END IF
'Check for valid date
'skip this for stat program.
'IF NOT (ValidDate%(tempdate$)) THEN
'  SOUND 900, 1
'  finishpage = 0
'  pointtoinput = 3
'  errorcode = 1
'END IF

      CASE 4
'time stuff
LOCATE mainrow + 8, maincol + 15
CALL templatehelp(Timehelp$, temptime$, "##:##", blankstring$, rc,
       timeerror$, errorcode)
```

```
IF rc = 2 THEN
  pointtoinput = 3
  ELSE
  finishpage = 1
END IF
'Check for valid time
'skip this for stat program.
'IF NOT validtime(temptime$) THEN
'  SOUND 900, 1
'  finishpage = 0
'  pointtoinput = 4
'  errorcode = 1
 'ELSE
'       Routine to check if all responses entered are OK
  IF rc <> 2 AND rc <> 1 THEN
' get confirmation
    CALL scrollup(17, 2, 23, 79, 0, backcolor)
    LOCATE 20, 26
    PRINT "Are These correct? (Y/N) [ ]";
    LOCATE 20, 52
    CALL GetUCResponse(ch$, "YN")
' if not OK then cycle back
    IF ch$ = "N" THEN
      finishpage = 0
      pointtoinput = 0
    END IF
  END IF
'END IF

      CASE ELSE
'shouldn't get here.
finishpage = 1

      END SELECT
    LOOP UNTIL finishpage = 1

'       Since using temp values, don't need to change anything if Escape
'       was pressed.  Check for other than Escape
    IF rc <> 1 THEN
'       Check if any variables are different from  original.  If so,
       then
'       do update STFLAG
      IF tempsex$ <> sex$ THEN
STFLAG = 1
sex$ = tempsex$
IF tempsex$ = "M" THEN
  VARIABLE(1) = 1
  VARIABLE(2) = 0
ELSE                            'if not male, then by def, female.
```

```
  VARIABLE(1) = 0
  VARIABLE(2) = 1
END IF
      END IF
      IF tempssn$ <> SSN$ THEN
STFLAG = 1
SSN$ = tempssn$
      END IF
      IF tempage$ <> AGE$ THEN
STFLAG = 1
AGE$ = tempage$
agevar = VAL(AGE$)
IF agevar > 79 THEN agevar = 79
agevar = INT(agevar / 10) + 3
FOR i = 3 TO 10
  VARIABLE(i) = 0
NEXT i
VARIABLE(agevar) = 1
      END IF
      IF tempdate$ <> STARTDATE$ THEN
STFLAG = 1
STARTDATE$ = tempdate$
      END IF
      IF temptime$ <> STARTIME$ THEN
STFLAG = 1
STARTIME$ = temptime$
      END IF
    END IF


END SUB

FUNCTION Translate% (HMDX)
'This routine translates the value of HMDX to a number between 1 and 7,
'the gyn database, otherwise it returns unchanged.
'       HMDX    the variable to modify
'       lookatfemale    flag for female database selection.

  SHARED lookatfemale

  IF lookatfemale = 0 THEN
    tempval = HMDX
  ELSE
    tempval = HMDX - 7
  END IF
  Translate = tempval

END FUNCTION
```

```
 SUB TXMenu (MAXNUM, sex$)
'       This routine displays the Treatment menu. Upon selection, the
'       treatment is displayed.

      SHARED VERSION$

      DIM Choices$(1, 12)
30004 REM  Tx protocol routine
      Choices$(1, 1) = "APPENDICITIS               "
      Choices$(1, 2) = "NON-SPECIFIC ABDOMINAL PAIN"
      Choices$(1, 3) = "RENAL COLIC                "
      Choices$(1, 4) = "PERFORATED DUODENAL ULCER  "
      Choices$(1, 5) = "CHOLECYSTITIS              "
      Choices$(1, 6) = "SMALL BOWEL OBSTRUCTION    "
      IF sex$ = MALE$ THEN
Choices$(1, 7) = "EXIT DISPLAY               "
resplength% = 7
      ELSE
Choices$(1, 7) = "PELVIC INFLAMMATORY DISEASE"  'dx # 10
Choices$(1, 8) = "URINARY TRACT INFECTION    "  'dx # 11
Choices$(1, 9) = "OVARIAN CYST               "  'dx # 12
Choices$(1, 10) = "ECTOPIC PREGNANCY          " 'dx # 13
Choices$(1, 11) = "INCOMPLETE ABORTION        " 'dx # 14
Choices$(1, 12) = "EXIT DISPLAY               " 'dx # 15
resplength% = 12
      END IF

    ' New method for menu
    SELECT CASE MAXNUM
      CASE 0, 7, 15     'none marked, or other dx
NR% = 1
      CASE 8            ' appi in gyn mode
NR% = 1
      CASE 9            'nonsap in gyn mode
NR% = 2
      CASE ELSE
NR% = MAXNUM
    END SELECT

    DATAHEADING$ = "Abdominal Pain Diagnosis Program" + VERSION$
    menuheading$ = "Treatment Summary"
    HELPFILE$ = "HP11.DAT"
    DO
      CALL MenuEntryPage(NR%, resplength%, exitchar$, DATAHEADING$,
       menuheading$, Choices$(), HELPFILE$)
      ' decrypt and print treatment text.
      IF NR <> resplength% THEN
TXfile$ = "TX" + MID$(STR$(NR%), 2) + ".DAT"
thispage% = VideoPage%
```

```
CALL DisplayEncryptedFile(TXfile$, thispage%)
      END IF
    LOOP UNTIL NR% = resplength%
    ERASE Choices$

END SUB
```

CRYPTDAT.BAS

```
DECLARE FUNCTION Exists% (FIL$)
DEFINT A-Z
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)

'       This routine converts the data file PHRASE.DAT from the original
'ASCII version to the encrypted version.
CLS
PRINT "This program encrypts or decrypts the ..DAT files used in ABDX."
PRINT "(PHRASE.DAT, BESTQUES.DAT, and ABDSX.DAT)"
PRINT
PRINT "What do you desire:"
PRINT
PRINT "  1.  Encrypt ASCII file to encrypted file."
PRINT "  2.  Decrypt .DATA file to ASCII file."
PRINT
DO
  LOCATE , 1
  PRINT "Enter your choice >      ";
  LOCATE , 21
  resp$ = INPUT$(1)
  IF resp$ = "" OR resp$ = CHR$(13) THEN
    END
  END IF
LOOP UNTIL INSTR("12", resp$) > 0
  CLS


'encrypt file
IF resp$ = "1" THEN
  PRINT "Enter name of ASCII file to encrypt. > ";
  LINE INPUT infile$
  IF infile$ = "" THEN END
  IF NOT Exists%(infile$) THEN
    LOCATE 10, 10
    PRINT infile$; " does not exist!  Aborting."
    END
  END IF
  PRINT "Enter name of output encrypted file. > ";
  LINE INPUT outfile$
  IF outfile$ = "" THEN END
  IF Exists%(outfile$) THEN
    LOCATE 10, 10
    PRINT outfile$; " already exists!  Aborting."
    END
  END IF
```

```
ELSE
  ' decrypt .DAT file
  PRINT "Enter name of .DAT file to decrypt. > ";
  LINE INPUT infile$
  IF infile$ = "" THEN END
  IF NOT Exists%(infile$) THEN
    LOCATE 10, 10
    PRINT infile$; " does not exist!  Aborting."
    END
  END IF
  PRINT "Enter name of output ASCII decrypted file. > ";
  LINE INPUT outfile$
  IF outfile$ = "" THEN END
  IF Exists%(outfile$) THEN
    LOCATE 10, 10
    PRINT outfile$; " already exists!  Aborting."
    END
  END IF
END IF

  OPEN infile$ FOR INPUT AS #1
  OPEN outfile$ FOR OUTPUT AS #2
  DO WHILE NOT EOF(1)
    LINE INPUT #1, a$
    IF resp$ = "1" THEN
      PRINT a$
      CALL encipher(a$)
    ELSE
      CALL decipher(a$)
      PRINT a$
    END IF
    PRINT #2, a$
  LOOP
  CLOSE #2
  CLOSE #1
  END

REM $DYNAMIC
FUNCTION Exists% (FIL$)
'       This function checks for the existance of the file fil$.  It
'       returns TRUE (non-zero) if present and false (zero) if not
       found.

  CONST FALSE = 0
  filenum = FREEFILE
  OPEN "R", filenum, FIL$, 1
  N% = LOF(filenum)
  CLOSE filenum
  IF N% = 0 THEN
```

CRYPTDAT.BAS (cont'd)

```
  booltest% - FALSE
 ELSE
  booltest% - NOT FALSE
 END IF
 Exists% - booltest%

END FUNCTION
```

CONVTEXT.BAS

```
DECLARE SUB encipher CDECL (a$)
DECLARE SUB decipher CDECL (a$)
DECLARE SUB SplitEm (orgstring$, string1$, string2$, splitchar$)
DECLARE SUB FFPresent (infile$, FFFlag%)
DECLARE SUB modifyFFfile (infile$, tempfile$)
DECLARE SUB encryptstringroutine (instring$, outstring$)
DECLARE SUB EncryptiontoASCII ()
DECLARE SUB ASCIItoEncryption ()
DECLARE SUB decryptstring (instring$, outstring$)
DECLARE FUNCTION Exists% (FIL$)
'Program is designed to take the ASCII version of a .DAT file for ABDX
'or CPDX and convert it to format used by those programs or vice vera.

'linecount - ptr for current line number of infile.

DEFINT A-Z
CLS
PRINT "This program converts an ASCII file to the encrypted data file"
PRINT "required by ABDX/CPDX.  Which do you desire?"
PRINT
PRINT "1.  Convert ASCII file to new encrypted data file."
PRINT "2.  Convert new encrypted data file to ASCII file."
PRINT
DO
  PRINT "Enter 1 or 2. >             ";
  LOCATE , 17
  a$ - INPUT$(1)
  IF a$ - "" OR a$ - CHR$(13) THEN
    END
  END IF
  aval - VAL(a$)
LOOP UNTIL aval - 1 OR aval - 2

IF aval - 1 THEN

  'ASCII to encrypted.
  CALL ASCIItoEncryption

ELSE
  ' encrypted to ASCII version
  CALL EncryptiontoASCII

END IF

SUB ASCIItoEncryption
'       This routine convert an ASCII file to the encrypted
```

```
'format required by ABDX/CPDX.  The user must mark the end of
'each display page with the character '|' on a line by itself.
'Mark the last page with two characters '||' on a line by themselves.

  CLS
  PRINT "Enter name of ASCII tx file for ABDX/CPDX. >";
  LINE INPUT infile$
  IF infile$ = "" THEN
    END
  ELSEIF NOT Exists%(infile$) THEN
    END
  END IF
  PRINT "Enter name of encrypted output file. >";
  LINE INPUT outfile$
  IF outfile$ = "" THEN
    END
  END IF
  IF Exists%(outfile$) THEN
    LOCATE 10, 10
    PRINT "Output file "; outfile$; " already exists!!!  Aborting."
    END
  END IF

  'check for FF's in file.  If present, then convert to "|" lines
  'in temporary file.
  tempfile$ = ""
  CALL FFPresent(infile$, FFFlag)
  IF FFFlag = 1 THEN    ' file uses FF
    tempfile$ = "T$emp.$at"
    CALL modifyFFfile(infile$, tempfile$)
    infile$ = tempfile$
  END IF

  pageflag$ = "|" ' identifies end of page if in col 1.
  DIM pageno(50)   ' max of 50 pages

  ' open  input file
  OPEN infile$ FOR INPUT AS #1

  'first pass thru infile to count number of pages.
  '     pageno(x) points to first line of page x.
  '     if = -1 then on first page.
  pageno(0) = -1
  pageno(1) = 1
  pagectr = 0
  linecount = 0
  DO WHILE NOT EOF(1)
    LINE INPUT #1, a$
    linecount = linecount + 1
  IF LEFT$(a$, 1) = pageflag$ THEN
    'end of page
    pagectr = pagectr + 1
    pageno(pagectr + 1) = linecount + 1
  END IF
LOOP
maxpages = pagectr
maxp$ = STR$(maxpages)
CLOSE #1

' open files
OPEN infile$ FOR INPUT AS #1
OPEN "R", 2, outfile$, 75
FIELD #2, 75 AS encrypt$

' second pass thru to modify program.
pagectr = 0
linecount = 0
lastpagestring$ = " "

DO WHILE NOT EOF(1)
  LINE INPUT #1, a$
  linecount = linecount + 1
  IF LEFT$(a$, 1) = pageflag$ THEN
    'end of page
    pagectr = pagectr + 1
    IF pagectr = maxpages THEN
      lastpagestring$ = "|"
    END IF
    ' PRINT #2, USING "|! ### Page ## of ##"; lastpagestring$,
    '                   pageno (pagectr - 1), pagectr, maxpages
    pageloc$ = RIGHT$(("   " + STR$(pageno(pagectr - 1))), 3)
    pnum$ = RIGHT$(("   " + STR$(pagectr)), 3)
    pagestuff$ = "Page" + STR$(pagectr) + " of" + maxp$
    a$ = "|" + lastpagestring$
    a$ = LEFT$((a$ + pagestuff$ + SPACE$(13)), 15)
    a$ = a$ + pageloc$
  END IF
  PRINT a$
  a$ = a$ + SPACE$(75)
  a$ = LEFT$(a$, 75)
  CALL encipher(a$)
  LSET encrypt$ = a$
  PUT #2
LOOP
CLOSE #1
CLOSE #2

'IF tempfile$ = infile$ THEN KILL tempfile$
```

```
END SUB

  SUB EncryptiontoASCII
'       This routine will convert an encrypted data file to a
'straight ASCII file.  It will place a '|' at the end of each page,
'and '||' at the end of the file.

  CLS
  PRINT "Enter name of encrypted input file for ABDX/CPDX. >";
  LINE INPUT infile$
  IF infile$ - "" THEN
    END
  ELSEIF NOT Exists%(infile$) THEN
    END
  END IF
  PRINT "Enter name of ASCII output file . >";
  LINE INPUT outfile$
  IF outfile$ - "" THEN
    END
  ELSEIF Exists%(outfile$) THEN
    LOCATE 10, 10
    PRINT "Output file "; outfile$; " already exists!!!  Aborting."
    END
  END IF

  ' open  input file
    OPEN "R", 1, infile$, 75
    FIELD #1, 75 AS B$
    maxRecNum - LOF(1) \ 75
    OPEN outfile$ FOR OUTPUT AS #2
    FOR RecNum - 1 TO maxRecNum
      GET #1, RecNum
      decryptedstring$ - B$
      CALL decipher(decryptedstring$)
      IF MID$(decryptedstring$, 1, 1) - "|" THEN
        ' end of page.
        'check for end of file.
        IF LEFT$(decryptedstring$, 2) - "||" THEN
          decryptedstring$ - "||"
        ELSE
          ' Convert to just "|" on line.
          decryptedstring$ - "|"
        END IF
      END IF
      PRINT decryptedstring$
      PRINT #2, decryptedstring$
    NEXT RecNum
    CLOSE #2
```

```
      CLOSE #1

END SUB

REM $DYNAMIC
FUNCTION Exists% (FIL$)
'       This function checks for the existance of the file fil$.  It
'       returns TRUE (non-zero) if present and false (zero) if not
       found.

  CONST FALSE = 0
  filenum = FREEFILE
  OPEN "R", filenum, FIL$, 1
  N% = LOF(filenum)
  CLOSE filenum
  IF N% = 0 THEN
    booltest% = FALSE
  ELSE
    booltest% = NOT FALSE
  END IF
  Exists% = booltest%

END FUNCTION

REM $STATIC
  SUB FFPresent (infile$, FFFlag)
'       This routine checks for a FF in the file infile$.  If present,
'it returns 1 in FFFlag, otherwise 0.
FFFlag = 0
filenum = FREEFILE
OPEN infile$ FOR INPUT AS #filenum
DO WHILE NOT EOF(filenum)
  LINE INPUT #filenum, a$
  IF a$ <> "" THEN
    IF INSTR(a$, CHR$(12)) > 0 THEN
      FFFlag = 1
      EXIT DO
    END IF
  END IF
LOOP
CLOSE filenum

END SUB

SUB modifyFFfile (infile$, tempfile$)
'       This file converts the FF's in infile$ to "|" in tempfile$.
'This is a quick hack, so it will only pick up the first FF in a line.
'Be aware.
```

```
FF$ = CHR$(12)
linecounter = 0
OPEN infile$ FOR INPUT AS #1
OPEN tempfile$ FOR OUTPUT AS #2
DO WHILE NOT EOF(1)
  LINE INPUT #1, a$
  linecounter = linecounter + 1
  IF a$ = "" THEN
    ' the TTYFF WORD printer driver always throws in a CR at the
      beginning
    ' of the file to ensure that the printer head is to the far left.
    ' We don't want that initial CR, since it wastes a line.
    IF linecounter > 1 THEN
      PRINT #2,
    END IF
  ELSE
    FFloc = INSTR(a$, FF$)
    IF FFloc = 0 THEN
      PRINT #2, a$
    ELSE
      CALL SplitEm(a$, string1$, string2$, FF$)
        IF string1$ <> "" THEN PRINT #2, string1$
        IF EOF(1) THEN
          PRINT #2, "||"
        ELSE
          PRINT #2, "|"
        END IF
        IF string2$ <> "" THEN PRINT #2, string2$
    END IF
  END IF
LOOP
CLOSE #2
CLOSE #1
END SUB

SUB SplitEm (orgstring$, string1$, string2$, splitchar$)
'       Splits orgstring$ into stringland string2 about splitchar$

string1$ = ""
string2$ = ""

IF orgstring$ = "" THEN
  EXIT SUB
END IF

IF splitchar$ = "" THEN
  string1$ = orgstring$
  EXIT SUB
END IF
```

CONVTEXT.BAS (cont'd)

```
orglen = LEN(orgstring$)
splitpos = INSTR(orgstring$, splitchar$)

IF splitpos = 0 THEN
  'FF not found.
  string1$ = orgstring$
ELSEIF splitpos = 1 THEN
  'FF at beginning
  string2$ = RIGHT$(orgstring$, orglen - 1)
ELSEIF splitpos = orglen THEN
  'FF at end
  string1$ = LEFT$(orgstring$, orglen - 1)
ELSE
  string1$ = MID$(orgstring$, 1, splitpos - 1)
  string2$ = MID$(orgstring$, splitpos + 1)
END IF

END SUB
```

INSTALL.BAS

```
DECLARE SUB AskQuestion (question$, response$, filter$, exitcode%)
DEFINT A-Z
'  This program installs the abdominal Pain diag program.
ON ERROR GOTO errorhandler
CLS
PROGTYPE$ = "Abdominal Pain"
BATCHNAME$ = "ABDOMEN"
EXECNAME$ = "ABDX.EXE"
SUBDIR$ = "ABDOMEN"
ARCFILE$ = "ABDXPAK.EXE"
BACKUPNAME$ = "ABDBKUP"
REALFILE$ = "REAL.DAT"
SIMULFILE$ = "SIMUL.DAT"

esc$ = CHR$(27)
YNESC$ = "YN" + esc$

a$ = PROGTYPE$ + " Diagnostic Program Installation"
LOCATE 1, (80 - LEN(a$)) \ 2
PRINT a$
LOCATE 5, 1
PRINT "This program will install the "; PROGTYPE$; " on your hard
       drive."
PRINT "Normally, just press the Enter key in response to each question
       asked"
PRINT "and the default values will be used."
PRINT
question$ = "Enter the letter of the hard drive on which the program
       will be installed."
driveletter$ = "D"
filter$ = "CDEFGHIJKLMNOP"
CALL AskQuestion(question$, driveletter$, filter$, exitcode)
PRINT
PRINT

row = CSRLIN
DO
  LOCATE row, 1
  PRINT "The subdirectory name used will be "; driveletter$; ":\";
       SUBDIR$
  PRINT
  question$ = "Is this OK?"
  YNresponse$ = "Y"
  filter$ = "YN"
  CALL AskQuestion(question$, YNresponse$, filter$, exitcode)
  IF YNresponse$ = "N" THEN
```

```
    PRINT
    PRINT "Enter the name of the subdirectory "; driveletter$; ":\";
    LINE INPUT SUBDIR$
    SUBDIR$ = UCASE$(SUBDIR$)
    subrow = CSRLIN - 1
    LOCATE subrow, 1: PRINT SPACE$(78)
    LOCATE row, 36: PRINT SPACE$(30)
  END IF
LOOP UNTIL YNresponse$ = "Y"

SUBDIR$ = driveletter$ + ":\" + SUBDIR$
CLS
PRINT "Creating subdirectory "; SUBDIR$
MKDIR SUBDIR$
CHDIR SUBDIR$
SHELL driveletter$ + ":"
' need input file of y's if subdir had old version of program in it.
PRINT

PRINT "Now copying program files.  (This may take a while.)"
SHELL "a:" + ARCFILE$ + " <A:Y.DAT >nul"
PRINT "Program files have been copied."

PRINT "Now copying the batch files."
OPEN "C:\" + BATCHNAME$ + ".BAT" FOR OUTPUT AS #1
PRINT #1, "ECHO OFF"
PRINT #1, driveletter$ + ":"
PRINT #1, "cd "; SUBDIR$
PRINT #1, EXECNAME$
PRINT #1, "cd\"
CLOSE #1
' create backup batch file.
OPEN "C:\" + BACKUPNAME$ + ".BAT" FOR OUTPUT AS #1
PRINT #1, "ECHO OFF"
PRINT #1, "COPY/V "; SUBDIR$; "\"; REALFILE$; " A:*.*"
PRINT #1, "COPY/V "; SUBDIR$; "\"; SIMULFILE$; " A:*.*"
CLOSE #1


PRINT " Batch files written."
PRINT
PRINT "Finished with installation of "; LCASE$(PROGTYPE$); " program."
PRINT
PRINT "  To start the "; LCASE$(PROGTYPE$); " program, enter ";
       BATCHNAME$; "."
PRINT
PRINT "  To backup the data files, put the backup disk into disk A, and"
PRINT "enter "; BACKUPNAME$; "."
```

```
END





errorhandler:
 'Error 75 - subdirectory already exists.
 IF ERR = 75 THEN
   PRINT "Subdirectory already exists."
   RESUME NEXT
 END IF
 ON ERROR GOTO 0

SUB AskQuestion (question$, response$, filter$, exitcode)
' This routine asks question$ at the current location, returning
' response$ as the result. Filter$ is list of permitted responses.
'exitcode =1 if esc from question.

filterstring$ = filter$ + CHR$(27) + CHR$(13)
exitcode = 0
  PRINT question$; "  [";
  row = CSRLIN
  col = POS(0)
  PRINT " ]";
  DO
    LOCATE row, col, 0
    PRINT response$;
    LOCATE row, col, 1
    a$ = INPUT$(1)
    a$ = UCASE$(a$)
    filtered = INSTR(filterstring$, a$)
    IF filtered = 0 THEN
      SOUND 450, 1
    END IF
  LOOP UNTIL filtered > 0
  IF INSTR(filter$, a$) > 0 THEN
    'valid new selection
    response$ = a$
  ELSEIF a$ = CHR$(27) THEN
    exitcode = 1
  ELSE
    'a$ = CHR$(13)
    'just pressed enter; selected the default
    ' don't need to do anything.
  END IF
```

INSTALL.BAS (cont'd)

```
  IF exitcode = 1 THEN
    BEEP
    END
  END IF

  LOCATE row, col, 0
  PRINT response$;

END SUB

SUB GetPhrase (question$, response$, exitcode)

END SUB
```

OLDCONV.BAS

```
DECLARE SUB SplitEm (orgstring$, string1$, string2$, splitchar$)
DECLARE SUB FFPresent (infile$, FFFlag%)
DECLARE SUB modifyFFfile (infile$, tempfile$)
DECLARE SUB encryptstringroutine (instring$, outstring$)
DECLARE SUB EncryptiontoASCII ()
DECLARE SUB ASCIItoEncryption ()
DECLARE SUB decryptstring (instring$, outstring$)
DECLARE FUNCTION Exists% (FIL$)
'Program is designed to take the ASCII version of a .DAT file for ABDX
'or CPDX and convert it to format used by those programs or vice vera.

'linecount = ptr for current line number of infile.

DEFINT A-Z
CLS
PRINT "This program converts an ASCII file to the encrypted data file"
PRINT "required by ABDX/CPDX.  Which do you desire?"
PRINT
PRINT "1.  Convert ASCII file to encrypted data file."
PRINT "2.  Convert encrypted data file to ASCII file."
PRINT
DO
  PRINT "Enter 1 or 2. >            ";
  LOCATE , 17
  LINE INPUT a$
  IF a$ = "" THEN
    END
  END IF
  aval = VAL(a$)
LOOP UNTIL aval = 1 OR aval = 2

IF aval = 1 THEN

  'ASCII to encrypted.
  CALL ASCIItoEncryption

ELSE
  ' encrypted to ASCII version
  CALL EncryptiontoASCII

END IF

SUB ASCIItoEncryption
'       This routine convert an ASCII file to the encrypted
'format required by ABDX/CPDX.  The user must mark the end of
'each display page with the character '|' on a line by itself.
```

```
'Mark the last page with two characters '||' on a line by themselves.

  CLS
  PRINT "Enter name of ASCII tx file for ABDX/CPDX. >";
  LINE INPUT infile$
  IF infile$ = "" THEN
    END
  ELSEIF NOT Exists%(infile$) THEN
    END
  END IF
  PRINT "Enter name of encrypted output file. >";
  LINE INPUT outfile$
  IF outfile$ = "" THEN
    END
  END IF
  IF Exists%(outfile$) THEN
    LOCATE 10, 10
    PRINT "Output file "; outfile$; " already exists!!!  Aborting."
    END
  END IF

  'check for FF's in file.  If present, then convert to "|" lines
  'in temporary file.
  tempfile$ = ""
  CALL FFPresent(infile$, FFFlag)
  IF FFFlag = 1 THEN    ' file uses FF
    tempfile$ = "T$emp.$at"
    CALL modifyFFfile(infile$, tempfile$)
    infile$ = tempfile$
  END IF

  pageflag$ = "|" ' identifies end of page if in col 1.
  DIM pageno(50)  ' max of 50 pages

  ' open  input file
  OPEN infile$ FOR INPUT AS #1

  'first pass thru infile to count number of pages.
  '     pageno(x) points to first line of page x.
  '     if = -1 then on first page.
  pageno(0) = -1
  pageno(1) = 1
  pagectr = 0
  linecount = 0
  DO WHILE NOT EOF(1)
    LINE INPUT #1, a$
    linecount = linecount + 1
    IF LEFT$(a$, 1) = pageflag$ THEN
      'end of page
      pagectr = pagectr + 1
      pageno(pagectr + 1) = linecount + 1
    END IF
  LOOP
  maxpages = pagectr
  maxp$ = STR$(maxpages)
  CLOSE #1

  ' open files
  OPEN infile$ FOR INPUT AS #1
  OPEN "R", 2, outfile$, 75
  FIELD #2, 75 AS encrypt$

  ' second pass thru to modify program.
  pagectr = 0
  linecount = 0
  lastpagestring$ = " "

  DO WHILE NOT EOF(1)
    LINE INPUT #1, a$
    linecount = linecount + 1
    IF LEFT$(a$, 1) = pageflag$ THEN
      'end of page
      pagectr = pagectr + 1
      IF pagectr = maxpages THEN
        lastpagestring$ = "|"
      END IF
      ' PRINT #2, USING "|! ### Page ## of ##"; lastpagestring$,
      '                    pageno (pagectr - 1), pagectr, maxpages
      pageloc$ = RIGHT$(("   " + STR$(pageno(pagectr - 1))), 3)
      pnum$ = RIGHT$(("   " + STR$(pagectr)), 3)
      pagestuff$ = "Page" + STR$(pagectr) + " of" + maxp$
      a$ = "|" + lastpagestring$
      a$ = LEFT$((a$ + pagestuff$ + SPACE$(13)), 15)
      a$ = a$ + pageloc$
    END IF
    PRINT a$
    a$ = a$ + SPACE$(75)
    a$ = LEFT$(a$, 75)
    CALL encryptstringroutine(a$, encryptedstring$)
    LSET encrypt$ = encryptedstring$
    PUT #2
  LOOP
  CLOSE #1
  CLOSE #2

  'IF tempfile$ = infile$ THEN KILL tempfile$

END SUB
```

```
SUB decryptstring (instring$, outstring$)
'       This routine decrypts the string instring$ into outstring$
      E$ = " &%David Southerland was here once upon a
       time.1EC07\X'40B$|:<~_ )@t?_~|.)["
      K = 73
      'clear extra spaces at end of record.
      'new method, but I don;t know if it is any faster.
'      instring$ = RTRIM$(instring$)
'      kk = LEN(instring$)
'      IF kk MOD 2 <> 0 THEN
'        kk = kk + 1
'        instring$ = instring$ + " "
'      END IF
36010 DO
        F$ = MID$(instring$, K, 2)
          IF F$ = "  " THEN
            K = K - 2
          END IF
      LOOP UNTIL K < 3 OR F$ <> "  "
36020 outstring$ = ""
      FOR I = K TO 1 STEP -2
        F$ = MID$(instring$, I, 2)
        G$ = MID$(E$, K + 1 - I, 2)
        F = CVI(F$)
        G = CVI(G$)
        H = F XOR G XOR &H3A73
        outstring$ = outstring$ + MKI$(H)
      NEXT I
END SUB

  SUB EncryptiontoASCII
'       This routine will convert an encrypted data file to a
'straight ASCII file.  It will place a '|' at the end of each page,
'and '||' at the end of the file.

  CLS
  PRINT "Enter name of encrypted input file for ABDX/CPDX. >";
  LINE INPUT infile$
  IF infile$ = "" THEN
    END
  ELSEIF NOT Exists%(infile$) THEN
    END
  END IF
  PRINT "Enter name of ASCII output file . >";
  LINE INPUT outfile$
  IF outfile$ = "" THEN
    END
  ELSEIF Exists%(outfile$) THEN
```

```
    LOCATE 10, 10
    PRINT "Output file "; outfile$; " already exists!!!  Aborting."
    END
  END IF

  ' open  input file
      OPEN "R", 1, infile$, 75
      FIELD #1, 75 AS B$
      maxRecNum = LOF(1) \ 75
      OPEN outfile$ FOR OUTPUT AS #2
      FOR RecNum = 1 TO maxRecNum
        GET #1, RecNum
        encryptstring$ = B$
        CALL decryptstring(encryptstring$, decryptedstring$)
        IF MID$(decryptedstring$, 1, 1) = "|" THEN
          ' end of page.
          'check for end of file.
          IF LEFT$(decryptedstring$, 2) = "||" THEN
            decryptedstring$ = "||"
          ELSE
            ' Convert to just "|" on line.
            decryptedstring$ = "|"
          END IF
        END IF
        PRINT decryptedstring$
        PRINT #2, decryptedstring$
      NEXT RecNum
      CLOSE #2
      CLOSE #1

END SUB

SUB encryptstringroutine (instring$, outstring$)
'       This routine decrypts the string instring$ into outstring$
      E$ = " &%David Southerland was here once upon a
       time.1EC07\X'40B$|:<-_ )@t?_-|.)["
     outstring$ = ""
     K = 73
     DO
210    F$ = MID$(instring$, K, 2)
       IF F$ = "  " THEN
         K = K - 2
         IF K < 3 THEN
           K = 3
         END IF
       END IF
     LOOP UNTIL F$ <> "  " OR K = 3

     FOR I = K TO 1 STEP -2
```

```
        F$ = MID$(instring$, I, 2)
        G$ = MID$(E$, I, 2)
        F = CVI(F$)
        G = CVI(G$)
        ' &h3A73 is added as an extra encryption step.
        H = F XOR G XOR &H3A73
        outstring$ = outstring$ + MKI$(H)
      NEXT I

END SUB

REM $DYNAMIC
FUNCTION Exists% (FIL$)
'        This function checks for the existance of the file fil$.  It
'        returns TRUE (non-zero) if present and false (zero) if not
       found.

  CONST FALSE = 0
  filenum = FREEFILE
  OPEN "R", filenum, FIL$, 1
  N% = LOF(filenum)
  CLOSE filenum
  IF N% = 0 THEN
    booltest% = FALSE
  ELSE
    booltest% = NOT FALSE
  END IF
  Exists% = booltest%

END FUNCTION

REM $STATIC
  SUB FFPresent (infile$, FFFlag)
'        This routine checks for a FF in the file infile$.  If present,
'it returns 1 in FFFlag, otherwise 0.
FFFlag = 0
filenum = FREEFILE
OPEN infile$ FOR INPUT AS #filenum
DO WHILE NOT EOF(filenum)
  LINE INPUT #filenum, a$
  IF a$ <> "" THEN
    IF INSTR(a$, CHR$(12)) > 0 THEN
      FFFlag = 1
      EXIT DO
    END IF
  END IF
LOOP
CLOSE filenum
```

```
END SUB

SUB modifyFFfile (infile$, tempfile$)
'       This file converts the FF's in infile$ to "|" in tempfile$.
'This is a quick hack, so it will only pick up the first FF in a line.
'Be aware.

FF$ = CHR$(12)
linecounter = 0
OPEN infile$ FOR INPUT AS #1
OPEN tempfile$ FOR OUTPUT AS #2
DO WHILE NOT EOF(1)
  LINE INPUT #1, a$
  linecounter = linecounter + 1
  IF a$ = "" THEN
    ' the TTYFF WORD printer driver always throws in a CR at the
      beginning
    ' of the file to ensure that the printer head is to the far left.
    ' We don't want that initial CR, since it wastes a line.
    IF linecounter > 1 THEN
      PRINT #2,
    END IF
  ELSE
    FFloc = INSTR(a$, FF$)
    IF FFloc = 0 THEN
      PRINT #2, a$
    ELSE
      CALL SplitEm(a$, string1$, string2$, FF$)
        IF string1$ <> "" THEN PRINT #2, string1$
        IF EOF(1) THEN
          PRINT #2, "||"
        ELSE
          PRINT #2, "|"
        END IF
        IF string2$ <> "" THEN PRINT #2, string2$
    END IF
  END IF
LOOP
CLOSE #2
CLOSE #1
END SUB

SUB SplitEm (orgstring$, string1$, string2$, splitchar$)
'       Splits orgstring$ into stringland string2 about splitchar$

string1$ = ""
string2$ = ""

IF orgstring$ = "" THEN
```

```
  EXIT SUB
END IF

IF splitchar$ = "" THEN
  string1$ = orgstring$
  EXIT SUB
END IF

orglen = LEN(orgstring$)
splitpos = INSTR(orgstring$, splitchar$)

IF splitpos = 0 THEN
  'FF not found.
  string1$ = orgstring$
ELSEIF splitpos = 1 THEN
  'FF at beginning
  string2$ = RIGHT$(orgstring$, orglen - 1)
ELSEIF splitpos = orglen THEN
  'FF at end
  string1$ = LEFT$(orgstring$, orglen - 1)
ELSE
  string1$ = MID$(orgstring$, 1, splitpos - 1)
  string2$ = MID$(orgstring$, splitpos + 1)
END IF

END SUB
```

PACKDATA.BAS

```
DECLARE FUNCTION Exists% (FIL$)
'Program Pack_DATABASE
'     This program takes the BASIC data statments in a file and converts
' them to a packed data file.  This program is used only for converting
' the ABD, CPDX database.  Any probability under 128 is placed in a
'single byte using CHR$().  If the probability is < 1 then, that number
       is
'multiplied by 10 (to get a whole number) and then added to 128.
'Conversion back will go like this:
'  modcheck=byte MOD 128
'  if modcheck = 0 then
'    prob = asc(byte)
'  else
'    prob = modcheck/10
'  end if

' NOTE that the input data file will contain BASIC DATA statments.
       Comments
'are allowed, but all REMs should be changed to '.  Also, if the comment
'is anywhere in the line then that particular line will be eliminated.
'Therefore, don't add comments at the end of a DATA statment line.
'
'  The format for the input data file is:
'OUTPUTFILE.DAT     <--- the output file name has to be on first line.
'7,152              <-- Second line contains # of diseases (cols) and
'                          the number of responses. The numbers here are
'                          examples for the male abd program.  There are
'                          7 diseases (dyspepsia and NONSAP get
       combined),
'                          and there are 152 responses on the program
       (count-
'                          ing sex and age)  The comments by each line
'                          would not be present in the actual input
       file.
  '  Next come two DATA lines containing the mantissa for each disease
       in the
'first line and the integer exponent in the second line. The exponent is
'always converted to negative, so you do not have to insert the negative
'sign in the data statement.  For example:
'DATA 1,2,3,4,5,6,7
'DATA 25,25,25,25,25,25,26
'  Again the comment quote would not be present in the actual file.
'  the a priori for disease (1) is 1 x 10^-25.
'   for disease (7), 7 X 10^-26.

DEFINT A-Z
```

```
CLS
PRINT "Enter the name of the BASIC DATA file to pack. >";
LINE INPUT infile$
IF infile$ = "" THEN END
IF NOT (Exists%(infile$)) THEN
  PRINT infile$; " NOT FOUND!!"
  END
END IF

OPEN infile$ FOR INPUT AS #1

INPUT #1, outfile$
PRINT "The name of the output packed data file is "; UCASE$(outfile$)
IF outfile$ = "" THEN
  CLOSE
  END
END IF
IF Exists%(outfile$) THEN
  BEEP
  PRINT outfile$; " ALREADY EXISTS!!"
  CLOSE
  END
END IF
PRINT
INPUT #1, arraywidth, arraylength
IF arraywidth = 0 OR arraylength = 0 THEN
  BEEP
  PRINT "Input file is in improper format!"
  PRINT "I cannot continue."
END IF


OPEN outfile$ FOR RANDOM AS #2 LEN = arraywidth
FIELD #2, arraywidth AS outline$

  recordnumber = 1
  WHILE NOT EOF(1)
    LINE INPUT #1, a$
    IF LEN(a$) <> 0 THEN
      startcomma = INSTR(a$, "DATA")
      IF INSTR(a$, "'") = 0 AND startcomma <> 0 THEN
        'got a data line.
        startcomma = startcomma + 4

        printstring$ = "output "
        outstring$ = ""
        PRINT a$
        FOR i = 1 TO arraywidth
          endcomma = INSTR(startcomma, a$, ",")
          IF endcomma = 0 THEN
            'last item on line
            stringvalue$ = MID$(a$, startcomma)
          ELSE
            stringvalue$ = MID$(a$, startcomma, endcomma - startcomma)
          END IF
          startcomma = endcomma + 1
          stringvariable! = ABS(VAL(stringvalue$))
          IF stringvariable! = 0 THEN stringvariable! = .1
          printstring$ = printstring$ + "  " + STR$(stringvariable!)
          IF stringvariable! < 1 THEN
            stringvariable! = stringvariable! * 10 + 128
          END IF
          outstring$ = outstring$ + CHR$(stringvariable!)
        NEXT i
        PRINT printstring$

        LSET outline$ = outstring$
        PUT #2, recordnumber
        recordnumber = recordnumber + 1

      END IF

    END IF
  WEND
  CLOSE #2
  CLOSE #1
  PRINT
  PRINT
  PRINT recordnumber - 1; " records printed, "; arraylength + 2;
  PRINT " records expected. (including a priori values)"

REM $DYNAMIC
FUNCTION Exists% (FIL$)
'       This function checks for the existance of the file fil$.  It
'       returns TRUE (non-zero) if present and false (zero) if not
       found.

  CONST FALSE = 0
  filenum = FREEFILE
  OPEN "R", filenum, FIL$, 1
  N% = LOF(filenum)
  CLOSE filenum
  IF N% = 0 THEN
    booltest% = FALSE
  ELSE
    booltest% = NOT FALSE
  END IF
  Exists% = booltest%
```

```
END FUNCTION
```

TRAINCAS.BAS

```
DEFINT A-Z
'       This program converts the 13 element array storing each of
''50 training cases for ABDX into 26 byte (2 bytes per integer) record,
' giving a total of 1300 bytes for the data file. (13 * 2 * 50)
DIM TrainingCase(12)
OPEN "abdtrn.dat" FOR BINARY AS #1
FOR NumCase = 1 TO 50
  CALL InitializeTrainingCase(NumCase, TrainingCase())
  FOR i = 0 TO 12
    caseval% = TrainingCase(i)
    PUT #1, , caseval%
  NEXT i
NEXT NumCase
CLOSE #1

' ____________________________________________________________________

SUB OrgInitializeTrainingCase (NumOfTrainingCase, TrainingCase())
'   This routine loads the array TrainingCase() with the desired
'  training case in compacted form.
      SELECT CASE NumOfTrainingCase
        CASE 1
          TrainingCase(0) = 33: TrainingCase(1) = 8201
          TrainingCase(2) = 3136: TrainingCase(3) = 8240
          TrainingCase(4) = 21017: TrainingCase(5) = 21704
          TrainingCase(6) = 4676: TrainingCase(7) = 21025
          TrainingCase(8) = 596: TrainingCase(9) = 1544
          TrainingCase(10) = 2: TrainingCase(11) = 0
          TrainingCase(12) = 0


        CASE 2
          TrainingCase(0) = 17: TrainingCase(1) = 1040
          TrainingCase(2) = 4736: TrainingCase(3) = 592
          TrainingCase(4) = 21061: TrainingCase(5) = 5448
          TrainingCase(6) = 18762: TrainingCase(7) = 26752
          TrainingCase(8) = 4690: TrainingCase(9) = 1280
          TrainingCase(10) = 1: TrainingCase(11) = 0
          TrainingCase(12) = 0

        CASE 3
          TrainingCase(0) = 33: TrainingCase(1) = 1
          TrainingCase(2) = 3138: TrainingCase(3) = 8240
          TrainingCase(4) = 21017: TrainingCase(5) = 21704
          TrainingCase(6) = 4676: TrainingCase(7) = 21025
          TrainingCase(8) = 596: TrainingCase(9) = 1544
          TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 4
  TrainingCase(0) = 17: TrainingCase(1) = 1032
  TrainingCase(2) = 5248: TrainingCase(3) = 8264
  TrainingCase(4) = 12825: TrainingCase(5) = 5328
  TrainingCase(6) = 4681: TrainingCase(7) = 21121
  TrainingCase(8) = 4682: TrainingCase(9) = 1280
  TrainingCase(10) = 1: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 5
  TrainingCase(0) = 33: TrainingCase(1) = 1026
  TrainingCase(2) = 2624: TrainingCase(3) = 552
  TrainingCase(4) = 12869: TrainingCase(5) = 5448
  TrainingCase(6) = 21066: TrainingCase(7) = 26880
  TrainingCase(8) = 4498: TrainingCase(9) = 1536
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 6
  TrainingCase(0) = 33: TrainingCase(1) = 16
  TrainingCase(2) = 3202: TrainingCase(3) = 560
  TrainingCase(4) = 21253: TrainingCase(5) = 5328
  TrainingCase(6) = 4677: TrainingCase(7) = 21122
  TrainingCase(8) = 652: TrainingCase(9) = 1537
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 7
  TrainingCase(0) = 1033: TrainingCase(1) = 16384
  TrainingCase(2) = 3136: TrainingCase(3) = 560
  TrainingCase(4) = 10822: TrainingCase(5) = 5416
  TrainingCase(6) = 4393: TrainingCase(7) = 26753
  TrainingCase(8) = 4690: TrainingCase(9) = 1536
  TrainingCase(10) = 1: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 8
  TrainingCase(0) = 65: TrainingCase(1) = 2
  TrainingCase(2) = 3336: TrainingCase(3) = 16432
  TrainingCase(4) = 3093: TrainingCase(5) = 5460
  TrainingCase(6) = 4433: TrainingCase(7) = 8721
  TrainingCase(8) = 653: TrainingCase(9) = 2592
  TrainingCase(10) = 1: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 9
  TrainingCase(0) = 129: TrainingCase(1) = 16386
  TrainingCase(2) = 1280: TrainingCase(3) = 592
  TrainingCase(4) = 12933: TrainingCase(5) = 19752
  TrainingCase(6) = 21064: TrainingCase(7) = 10272
  TrainingCase(8) = 661: TrainingCase(9) = 6688
  TrainingCase(10) = 8: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 10
  TrainingCase(0) = 17: TrainingCase(1) = 16386
  TrainingCase(2) = 3200: TrainingCase(3) = 16432
  TrainingCase(4) = 5145: TrainingCase(5) = 11444
  TrainingCase(6) = 4388: TrainingCase(7) = 21009
  TrainingCase(8) = 650: TrainingCase(9) = 6688
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 11
  TrainingCase(0) = 33: TrainingCase(1) = 64
  TrainingCase(2) = 2704: TrainingCase(3) = 584
  TrainingCase(4) = 5193: TrainingCase(5) = 13620
  TrainingCase(6) = 4744: TrainingCase(7) = 8737
  TrainingCase(8) = 405: TrainingCase(9) = 6784
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 12
  TrainingCase(0) = 2113: TrainingCase(1) = 0
  TrainingCase(2) = 3329: TrainingCase(3) = 1104
  TrainingCase(4) = 4629: TrainingCase(5) = 12969
  TrainingCase(6) = 21064: TrainingCase(7) = 21056
  TrainingCase(8) = 660: TrainingCase(9) = 19008
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 13
  TrainingCase(0) = 33: TrainingCase(1) = 2049
  TrainingCase(2) = 2624: TrainingCase(3) = 16424
  TrainingCase(4) = 5146: TrainingCase(5) = 21844
  TrainingCase(6) = 4392: TrainingCase(7) = 8769
  TrainingCase(8) = 653: TrainingCase(9) = 19008
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 14
  TrainingCase(0) = 17: TrainingCase(1) = 64
  TrainingCase(2) = 9488: TrainingCase(3) = 16464
  TrainingCase(4) = 5189: TrainingCase(5) = 21844
  TrainingCase(6) = 21640: TrainingCase(7) = 8736
  TrainingCase(8) = 8845: TrainingCase(9) = 18944
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 15
  TrainingCase(0) = 8257: TrainingCase(1) = 0
  TrainingCase(2) = 2640: TrainingCase(3) = 1064
  TrainingCase(4) = 21061: TrainingCase(5) = 21808
  TrainingCase(6) = 2384: TrainingCase(7) = 10258
  TrainingCase(8) = 661: TrainingCase(9) = 6657
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 16
  TrainingCase(0) = 4161: TrainingCase(1) = 0
  TrainingCase(2) = 9480: TrainingCase(3) = 16528
  TrainingCase(4) = 5146: TrainingCase(5) = 21204
  TrainingCase(6) = 3204: TrainingCase(7) = 22561
  TrainingCase(8) = 652: TrainingCase(9) = 6688
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 17
  TrainingCase(0) = 129: TrainingCase(1) = 4
  TrainingCase(2) = 2640: TrainingCase(3) = 560
  TrainingCase(4) = 5146: TrainingCase(5) = 5460
  TrainingCase(6) = 5266: TrainingCase(7) = 26753
  TrainingCase(8) = 660: TrainingCase(9) = 6720
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 18
  TrainingCase(0) = 1089: TrainingCase(1) = 4096
  TrainingCase(2) = 9344: TrainingCase(3) = 1064
  TrainingCase(4) = 12873: TrainingCase(5) = 5416
  TrainingCase(6) = 5265: TrainingCase(7) = 26689
  TrainingCase(8) = 1362: TrainingCase(9) = 5376
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 19
  TrainingCase(0) = 129: TrainingCase(1) = 258
  TrainingCase(2) = 9344: TrainingCase(3) = 560
  TrainingCase(4) = 10821: TrainingCase(5) = 13512
  TrainingCase(6) = 4680: TrainingCase(7) = 22657
  TrainingCase(8) = 594: TrainingCase(9) = 5384
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 20
```

```
  TrainingCase(0) = 1057: TrainingCase(1) = 0
  TrainingCase(2) = 1154: TrainingCase(3) = 16529
  TrainingCase(4) = 12821: TrainingCase(5) = 2888
  TrainingCase(6) = 4682: TrainingCase(7) = 21121
  TrainingCase(8) = 594: TrainingCase(9) = 1296
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 21
  TrainingCase(0) = 1041: TrainingCase(1) = 0
  TrainingCase(2) = 3204: TrainingCase(3) = 560
  TrainingCase(4) = 10778: TrainingCase(5) = 13640
  TrainingCase(6) = 4388: TrainingCase(7) = 26648
  TrainingCase(8) = 594: TrainingCase(9) = 1288
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 22
  TrainingCase(0) = 16417: TrainingCase(1) = 0
  TrainingCase(2) = 1156: TrainingCase(3) = 16434
  TrainingCase(4) = 13338: TrainingCase(5) = 5424
  TrainingCase(6) = 4681: TrainingCase(7) = 26760
  TrainingCase(8) = 594: TrainingCase(9) = 5392
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 23
  TrainingCase(0) = 65: TrainingCase(1) = 2
  TrainingCase(2) = 9346: TrainingCase(3) = 16432
  TrainingCase(4) = 10777: TrainingCase(5) = 21832
  TrainingCase(6) = 5264: TrainingCase(7) = 25105
  TrainingCase(8) = 594: TrainingCase(9) = 5648
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 24
  TrainingCase(0) = 1057: TrainingCase(1) = 0
  TrainingCase(2) = 3140: TrainingCase(3) = 584
  TrainingCase(4) = 10821: TrainingCase(5) = 11440
  TrainingCase(6) = 4680: TrainingCase(7) = 22657
  TrainingCase(8) = 596: TrainingCase(9) = 5640
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 25
  TrainingCase(0) = 16393: TrainingCase(1) = 0
  TrainingCase(2) = 1156: TrainingCase(3) = 16433
  TrainingCase(4) = 12825: TrainingCase(5) = 13608
  TrainingCase(6) = 5264: TrainingCase(7) = 25352
  TrainingCase(8) = 658: TrainingCase(9) = 1296
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 26
  TrainingCase(0) = 1089: TrainingCase(1) = 256
  TrainingCase(2) = 16960: TrainingCase(3) = 2120
  TrainingCase(4) = 10773: TrainingCase(5) = 11048
  TrainingCase(6) = 9360: TrainingCase(7) = 26754
  TrainingCase(8) = 1364: TrainingCase(9) = 5376
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 27
  TrainingCase(0) = 1057: TrainingCase(1) = 4096
  TrainingCase(2) = 17024: TrainingCase(3) = 2192
  TrainingCase(4) = 18981: TrainingCase(5) = 11048
  TrainingCase(6) = 8488: TrainingCase(7) = 25218
  TrainingCase(8) = 1612: TrainingCase(9) = 6400
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 28
  TrainingCase(0) = 16401: TrainingCase(1) = 256
  TrainingCase(2) = 8960: TrainingCase(3) = 2180
  TrainingCase(4) = 3337: TrainingCase(5) = 11050
  TrainingCase(6) = 19016: TrainingCase(7) = 25120
  TrainingCase(8) = 1420: TrainingCase(9) = 6400
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 29
  TrainingCase(0) = 1089: TrainingCase(1) = 256
  TrainingCase(2) = 17536: TrainingCase(3) = 8240
  TrainingCase(4) = 2633: TrainingCase(5) = 5290
  TrainingCase(6) = 8517: TrainingCase(7) = 21252
  TrainingCase(8) = 1380: TrainingCase(9) = 6400
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 30
  TrainingCase(0) = 16401: TrainingCase(1) = 4096
  TrainingCase(2) = 16960: TrainingCase(3) = 2312
  TrainingCase(4) = 4677: TrainingCase(5) = 13521
  TrainingCase(6) = 2340: TrainingCase(7) = 22658
  TrainingCase(8) = 594: TrainingCase(9) = 6664
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0
```

```
CASE 31
  TrainingCase(0) = 65: TrainingCase(1) = 258
  TrainingCase(2) = 9280: TrainingCase(3) = 560
  TrainingCase(4) = 2597: TrainingCase(5) = 3370
  TrainingCase(6) = 9354: TrainingCase(7) = 26692
  TrainingCase(8) = 596: TrainingCase(9) = 5378
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 32
  TrainingCase(0) = 33: TrainingCase(1) = 16
  TrainingCase(2) = 3202: TrainingCase(3) = 648
  TrainingCase(4) = 12874: TrainingCase(5) = 13520
  TrainingCase(6) = 4496: TrainingCase(7) = 21528
  TrainingCase(8) = 354: TrainingCase(9) = 1544
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 33
  TrainingCase(0) = 16401: TrainingCase(1) = 0
  TrainingCase(2) = 578: TrainingCase(3) = 16457
  TrainingCase(4) = 21125: TrainingCase(5) = 3272
  TrainingCase(6) = 5266: TrainingCase(7) = 21521
  TrainingCase(8) = 612: TrainingCase(9) = 9736
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 34
  TrainingCase(0) = 257: TrainingCase(1) = 2
  TrainingCase(2) = 1154: TrainingCase(3) = 16466
  TrainingCase(4) = 21029: TrainingCase(5) = 11464
  TrainingCase(6) = 5264: TrainingCase(7) = 21570
  TrainingCase(8) = 418: TrainingCase(9) = 9736
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 35
  TrainingCase(0) = 33: TrainingCase(1) = 4
  TrainingCase(2) = 3137: TrainingCase(3) = 584
  TrainingCase(4) = 5189: TrainingCase(5) = 5290
  TrainingCase(6) = 2633: TrainingCase(7) = 22664
  TrainingCase(8) = 674: TrainingCase(9) = 1544
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 36
  TrainingCase(0) = 65: TrainingCase(1) = 1
  TrainingCase(2) = 17538: TrainingCase(3) = 8240
  TrainingCase(4) = 13385: TrainingCase(5) = 13480
  TrainingCase(6) = 8528: TrainingCase(7) = 21122
  TrainingCase(8) = 338: TrainingCase(9) = 1288
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 37
  TrainingCase(0) = 2065: TrainingCase(1) = 0
  TrainingCase(2) = 1153: TrainingCase(3) = 16454
  TrainingCase(4) = 13445: TrainingCase(5) = 3280
  TrainingCase(6) = 8777: TrainingCase(7) = 5256
  TrainingCase(8) = 2389: TrainingCase(9) = 5632
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 38
  TrainingCase(0) = 9: TrainingCase(1) = 8193
  TrainingCase(2) = 16960: TrainingCase(3) = 2180
  TrainingCase(4) = 10777: TrainingCase(5) = 11048
  TrainingCase(6) = 2340: TrainingCase(7) = 25121
  TrainingCase(8) = 588: TrainingCase(9) = 2561
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 39
  TrainingCase(0) = 9: TrainingCase(1) = 8193
  TrainingCase(2) = 16960: TrainingCase(3) = 2180
  TrainingCase(4) = 10777: TrainingCase(5) = 11048
  TrainingCase(6) = 2340: TrainingCase(7) = 25124
  TrainingCase(8) = 588: TrainingCase(9) = 2561
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 40
  TrainingCase(0) = 33: TrainingCase(1) = 2064
  TrainingCase(2) = 17024: TrainingCase(3) = 16520
  TrainingCase(4) = 3145: TrainingCase(5) = 11562
  TrainingCase(6) = 8776: TrainingCase(7) = 25108
  TrainingCase(8) = 8788: TrainingCase(9) = 17920
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 41
  TrainingCase(0) = 16449: TrainingCase(1) = 0
  TrainingCase(2) = 17028: TrainingCase(3) = 1104
  TrainingCase(4) = 2581: TrainingCase(5) = 2890
  TrainingCase(6) = 2377: TrainingCase(7) = 26884
  TrainingCase(8) = 610: TrainingCase(9) = 1544
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0
```

```
CASE 42
  TrainingCase(0) = 1033: TrainingCase(1) = 4096
  TrainingCase(2) = 3136: TrainingCase(3) = 584
  TrainingCase(4) = 21546: TrainingCase(5) = 5448
  TrainingCase(6) = 24870: TrainingCase(7) = 25344
  TrainingCase(8) = 1684: TrainingCase(9) = 6400
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 43
  TrainingCase(0) = 9: TrainingCase(1) = 8
  TrainingCase(2) = 1282: TrainingCase(3) = 562
  TrainingCase(4) = 12873: TrainingCase(5) = 3368
  TrainingCase(6) = 2450: TrainingCase(7) = 25154
  TrainingCase(8) = 356: TrainingCase(9) = 6664
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 44
  TrainingCase(0) = 17: TrainingCase(1) = 8208
  TrainingCase(2) = 640: TrainingCase(3) = 4357
  TrainingCase(4) = 4681: TrainingCase(5) = 5449
  TrainingCase(6) = 18761: TrainingCase(7) = 25120
  TrainingCase(8) = 588: TrainingCase(9) = 2561
  TrainingCase(10) = 2: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 45
  TrainingCase(0) = 65: TrainingCase(1) = 4112
  TrainingCase(2) = 4736: TrainingCase(3) = 1156
  TrainingCase(4) = 11301: TrainingCase(5) = 11592
  TrainingCase(6) = 9348: TrainingCase(7) = 25348
  TrainingCase(8) = 1636: TrainingCase(9) = 6400
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 46
  TrainingCase(0) = 129: TrainingCase(1) = 8
  TrainingCase(2) = 1092: TrainingCase(3) = 16465
  TrainingCase(4) = 13337: TrainingCase(5) = 5448
  TrainingCase(6) = 18761: TrainingCase(7) = 25216
  TrainingCase(8) = 588: TrainingCase(9) = 5384
  TrainingCase(10) = 0: TrainingCase(11) = 0
  TrainingCase(12) = 0

CASE 47
  TrainingCase(0) = 16513: TrainingCase(1) = 0
  TrainingCase(2) = 17028: TrainingCase(3) = 2308
```

```
          TrainingCase(4) = 12837: TrainingCase(5) = 3368
          TrainingCase(6) = 9361: TrainingCase(7) = 8722
          TrainingCase(8) = 17059: TrainingCase(9) = 6400
          TrainingCase(10) = 0: TrainingCase(11) = 0
          TrainingCase(12) = 0

        CASE 48
          TrainingCase(0) = 8209: TrainingCase(1) = 0
          TrainingCase(2) = 3140: TrainingCase(3) = 592
          TrainingCase(4) = 13353: TrainingCase(5) = 13640
          TrainingCase(6) = 9360: TrainingCase(7) = 25122
          TrainingCase(8) = 610: TrainingCase(9) = 6664
          TrainingCase(10) = 0: TrainingCase(11) = 0
          TrainingCase(12) = 0

        CASE 49
          TrainingCase(0) = 1033: TrainingCase(1) = 0
          TrainingCase(2) = 17028: TrainingCase(3) = 2308
          TrainingCase(4) = 10777: TrainingCase(5) = 19624
          TrainingCase(6) = 8584: TrainingCase(7) = 22660
          TrainingCase(8) = 610: TrainingCase(9) = 6416
          TrainingCase(10) = 0: TrainingCase(11) = 0
          TrainingCase(12) = 0

        CASE 50
          TrainingCase(0) = 17: TrainingCase(1) = 2056
          TrainingCase(2) = 9280: TrainingCase(3) = 16464
          TrainingCase(4) = 12869: TrainingCase(5) = 13640
          TrainingCase(6) = 8848: TrainingCase(7) = 25730
          TrainingCase(8) = 8804: TrainingCase(9) = 5632
          TrainingCase(10) = 0: TrainingCase(11) = 0
          TrainingCase(12) = 0
        CASE ELSE
          'should never get here, since range checking is done earlier.
          CLS
          PRINT "selected case "; CASENUM; " which does not exist."
          STOP
      END SELECT

'       end of InitializeTrainingCase()
END SUB
```

TXTMAKE.BAS

```
DECLARE SUB ErrorStuff (num%, click%)
DEFINT A-Z
KEY OFF
'This program will take an input ASCII file and create the appriopriate
' .TXT file for use by CPDX or ABDX.
' The format for the input file is:
'|H14.TXT
'##1
'  text, blah,blah
'blah, blah.
'##END
'##2
' text for question 2. blah
'blah.
'##END
'|H24.TXT
' same stuff again.

CLS
PRINT "   This program will take an input ASCII file and create the"
PRINT "appropriate .TXT help file for use by ABDX or CPDX."
PRINT
PRINT "  The format for the input file is:"
PRINT
PRINT
       "Öááááááááááááááááááááááááááááááááááááááááááááááááááááááááááááááá
       ááÇ"
PRINT "°|H14.TXT
       °"
PRINT "°##1
       °"
PRINT "°  text, blah,blah
       °"
PRINT "°blah, blah.
       °"
PRINT "°##END
       °"
PRINT "°  These two lines will not appear in the text file H14.TXT,
       since°"
PRINT "°they are not within  the ##1 and ##END.
       °"
PRINT "°##2
       °"
PRINT "° text for question 2. blah
       °"
```

```
PRINT "°blah.
       °"
PRINT "°##END
       °"
PRINT "°|H24.TXT
       °"
PRINT "°##1
       °"
PRINT "° As above, but this time, use a different file.
       °"
PRINT "°##END
       °"
PRINT
       "âááááááááááááááááááááááááááááááááááááááááááááááááááááááááááááááá
       ááì"
COLOR 0, 7: LOCATE 25, 27: PRINT "Press any key To Continue.";
COLOR 7, 0
a$ = INPUT$(1)
CLS
PRINT
PRINT "Note that the text file name begins with the character |."
PRINT " The number of the question begins with ##"
PRINT " and the end of the question text ends with ##END."
PRINT

PRINT "  Enter name of the input ASCII file. > ";
INPUT infile$
IF infile$ = "" THEN
  BEEP
  END
END IF

'check for input file existence.
OPEN infile$ FOR RANDOM AS #1 LEN = 1
N = LOF(1)
CLOSE #1
IF N = 0 THEN
  PRINT "File ;infile$;"; NOT found.; Aborting.; ""
  END
END IF

' open input file for real.

OPEN infile$ FOR INPUT AS #1
questnum = 0:            ' The question number.
InAQuestion = 0          ' Text is in a question 0-false, 1-true.
screenlines = 0          ' line counter for display screen.
LinesPerScreen = 23      ' Max lines per screen.
fileopen = 0:             ' output file is open 0-false, 1-true.
```

```
click = 0:                ' infile absolute line counter.
DO WHILE NOT EOF(1)

  LINE INPUT #1, inline$
  click = click + 1
  inlength = LEN(inline$)
  IF inlength > 1 THEN
    char1$ = LEFT$(inline$, 1)
    char2$ = MID$(inline$, 2, 1)
    char12$ = MID$(inline$, 1, 2)

    IF char1$ = "|" THEN
      ' file open/close.
      ' open file.   eg: |H14.TXT
        IF fileopen <> 0 THEN
          ' file is already open. Close it first.
          PRINT #2, questnum + 1; ", END"
          PRINT questnum + 1; ", END"
          PRINT "Closing file "; outfile$
          PRINT
          CLOSE #2
          fileopen = 0
          questnum = 0
          screenlines = 0
        END IF
        outfile$ = MID$(inline$, 2, inlength - 1)
        PRINT "Opening file "; outfile$
        PRINT
        OPEN outfile$ FOR OUTPUT AS #2
        fileopen = 1

    ELSEIF char12$ = "##" THEN
      'get question number
      '##1
      qstring$ = UCASE$(RIGHT$(inline$, inlength - 2))
      IF RTRIM$(qstring$) <> "END" THEN
        ' check for question number.
        qnum = VAL(qstring$)
        IF qnum <> questnum + 1 THEN
          'question numbers are out of sequence.
          CALL ErrorStuff(3, click)
        ELSE
          questnum = qnum
          InAQuestion = 1
          screenlines = 0
        END IF
      ELSE
        'end of question statements
        InAQuestion = 0
      END IF

    ELSE
      'normal help line of text.
      IF fileopen = 1 AND InAQuestion = 1 THEN
        ' text for output file.
        PRINT #2, questnum; ","; inline$
        PRINT questnum; ","; inline$
        screenlines = screenlines + 1
        IF LinesPerScreen < screenlines THEN
          CALL ErrorStuff(4, click)
          CLOSE #2
        END IF
      END IF
    END IF

  ELSE
    'len is less than 2
    IF fileopen = 1 AND InAQuestion = 1 THEN
      ' text for output file.
      PRINT #2, questnum; ", "; inline$
      PRINT questnum; ", "; inline$
      screenlines = screenlines + 1
      IF LinesPerScreen < screenlines THEN
        CALL ErrorStuff(4, click)
        CLOSE #2
      END IF
    END IF
  END IF

LOOP
CLOSE #1
IF fileopen <> 0 THEN
  'output file is open.  Add extra line number, then close it.
  PRINT #2, questnum + 1; ", END"
  PRINT questnum; ", END"
  CLOSE #2
END IF
END

SUB ErrorStuff (num, click)
' This routine prints out the error statements.

PRINT
PRINT
BEEP
PRINT "***** ERROR AT LINE "; click; " *****"

SELECT CASE num
```

```
  CASE 1
    PRINT "Output file is already open.  Aborting."
  CASE 2
    PRINT "Output file is already closed.  Aborting."
  CASE 3
    PRINT "Question numbers are out of sequence.  Aborting."
  CASE 4
    PRINT "Too many lines for display screen. Aborting."
  CASE 5
    PRINT "End of input file encountered before end of .TXT file.
       Aborting."
  CASE ELSE
    PRINT "Undefined error."
  END SELECT

CLOSE
END SUB
```

Appendix C
Definition File Listings

H14.TXT

```
1 ,  SITE OF PAIN at ONSET definition:
1 ,
1 ,     Mark the response in the left column which represents the
1 ,site at the time the pain started.  The corresponding area of the
1 ,diagram will then be highlighted.  Occasionally, pain isn't the
1 ,first symptom of an illness which later involves abdominal pain.
1 ,This would require an entry of NO PAIN.
2 ,  SITE OF PAIN at PRESENT definition:
2 ,
2 ,     Mark the response in the right column which represents the
2 ,site of pain at the time of examination.  The corresponding area
2 ,of the diagram will then be highlighted.
3 , END
```

H24.TXT

```
1 ,  TYPE OF PAIN definition:
1 ,
1 ,           INTERMITTENT
1 ,           STEADY
1 ,           COLICKY
1 ,
1 ,     It's especially important to distinguish between the types
1 ,of pain.  If the pain is constant and present the whole time,
1 ,mark STEADY on your data sheet.  If the pain is present the whole
1 ,time but varies rhythmically in intensity, mark COLICKY.  If your
1 ,patient has times when he's free of pain, mark INTERMITTENT.
2 ,  SEVERITY OF PAIN definition:
2 ,
2 ,           MODERATE
2 ,           SEVERE
2 ,
2 ,     Do not ask the patient directly and do not expect to rely on
2 ,the answer if you do.  If the pain is obviously intense and is
2 ,causing obvious distress such as sweating or shivering, then it
2 ,is SEVERE, otherwise it is MODERATE.  The difficulty in
2 ,distinguishing between mild and moderate pain is so great that we
2 ,prefer to call all pain moderate or severe.  Be especially wary
2 ,of relying on the patient's description since the threshold for
2 ,pain varies greatly between people.  A patient with mild or
2 ,moderate pain may make a great deal of fuss about it.  The
2 ,patient who is quiet may be in severe pain.  Use your own
2 ,judgment.
3 ,  AGGRAVATING FACTORS definition:
3 ,
3 ,           MOVEMENT
3 ,           COUGH
3 ,           BREATHING
3 ,           FOOD
3 ,           OTHER
3 ,           NONE
3 ,
3 ,   These are factors which make the pain worse.  The  patient
3 ,should be asked about each item specifically.  The patient should
3 ,be asked to actually perform some MOVEMENT, deep BREATHING and
3 ,COUGH to see if his pain is aggravated.
4 ,  PROGRESS OF PAIN definition:
4 ,
4 ,           BETTER
4 ,           SAME
4 ,           WORSE
4 ,
```

H24.TXT (cont'd)

```
4 ,     Pain which has subjectively remained unchanged is termed
4 ,SAME.  The categories of BETTER and WORSE refer to subjective
4 ,changes in pain prior to seeing the practitioner.  It is often
4 ,difficult to evaluate this category since abdominal pain can vary
4 ,so much.  In this instance, don't mark any of the choices.
4 ,Instead, write 'VARIABLE' on the data sheet and skip this entry
4 ,when performing data input.
5 ,  DURATION OF PAIN definition:
5 ,
5 ,                         12h or less
5 ,                         12-24h
5 ,                         24-48h
5 ,                         48+h
5 ,
5 ,     Acute abdominal pain is classically defined as pain
5 ,beginning less than 48 hours prior to presentation.  The term
5 ,DURATION refers to the length of time which the patient has had
5 ,the pain during this current episode.  Similar pain in the past
5 ,(i.e. at some time greater than 48 hours ago), should be
5 ,entered under the category of PREVIOUS SIMILAR PAIN.
6 ,  RELIEVING FACTORS definition:
6 ,
6 ,                         LYING STILL
6 ,                         VOMITING
6 ,                         ANTACIDS
6 ,                         FOOD
6 ,                         OTHER
6 ,                         NONE
6 ,
6 ,     These are factors that make the pain subside.  The patient
6 ,should be asked about each item specifically.
7 , END
```

H34.TXT

```
1 ,  NAUSEA definition:
1 ,
1 ,          PRESENT
1 ,          ABSENT
1 ,
1 ,     NAUSEA refers to the feeling of being sick to one's
1 ,stomach.  It is possible to have nausea without vomiting and
1 ,occasionally, vomiting without nausea.  This category refers to
1 ,the entire present episode, not just at the time he is seen by
1 ,the practitioner.
2 ,  VOMITING definition:
2 ,
2 ,          PRESENT
2 ,          ABSENT
2 ,
2 ,     Ask if the patient has vomited at any time during this
2 ,present episode of pain.
3 ,  BOWELS definition:
3 ,
3 ,          NORMAL
3 ,          CONSTIPATED
3 ,          DIARRHEA
3 ,          BLOOD IN STOOL
3 ,          MUCUS IN STOOL
3 ,
3 ,     Identify any recent change in the patient's bowel habits.
3 ,Remember that 'normal' bowel habits vary widely from one patient
3 ,to another.  BLOOD IN STOOL refers to either bright red blood
3 ,(hematochezia) or black, tarry stool (melena).  If the duration
3 ,of the present illness is too short to be certain of a choice
3 ,from this category, leave the entry blank on the data sheet and
3 ,skip this category when entering data.
4 ,  APPETITE definition:
4 ,
4 ,          DECREASED
4 ,          NORMAL
4 ,
4 ,     This category is intended to reflect any recent change in
4 ,the patient's appetite.  If the patient typically has a poor
4 ,appetite with no significant change during this current illness,
4 ,then his appetite is NORMAL.
5 ,  JAUNDICE definition:
5 ,
5 ,          PRESENT
5 ,          ABSENT
5 ,
```

H34.TXT (cont'd)

```
5 ,     The term JAUNDICE in the portion refers to a appearance of
5 ,yellow color to the patient's skin or sclera.  Mark PRESENT if
5 ,you note it now or has been noticed recently by the patient.
6 ,  URINATION definition:
6 ,
6 ,          NORMAL
6 ,          FREQUENCY
6 ,          PAINFUL
6 ,          DARK URINE
6 ,          BLOOD IN URINE
6 ,
6 ,     Note to any.recent change in urinary habits.  Mark FREQUENCY
6 ,if the patient relates that he has been urinating more often than
6 ,usual.  DARK URINE may be due to gall bladder illness but
6 ,occasionally indicates dehydration.
6 ,
6 ,     BLOOD IN URINE (either grossly or by Labstix) often
6 ,indicates a stone in the urinary tract, especially when
6 ,accompanied by flank pain.
7 , END
```

H44.TXT

```
1 ,  PREVIOUS INDIGESTION definition
1 ,
1 ,          YES
1 ,          NO
1 ,
1 ,     PREVIOUS INDIGESTION refers to gastrointestinal upset prior
1 ,to, or as a prodrome to the present episode.
2 ,  PREVIOUS SIMILAR PAIN definition:
2 ,
2 ,          YES
2 ,          NO
2 ,
2 ,     Mark YES if your patient has had a bout of similar
2 ,abdominal pain prior to the present episode.
3 ,  PREVIOUS SURGERY definition:
3 ,
3 ,          YES
3 ,          NO
3 ,
3 ,     Mark YES if the patient has had abdominal surgery.  Only
3 ,those operations where the peritoneum was entered should be
3 ,counted.  When in doubt, check for operative reports in the
3 ,patient's health record.
3 ,
3 ,     Prior history of inguinal herniorrhaphy may be very
3 ,important to the diagnosis, especially if there was incarceration
3 ,of bowel involved.  Note this history carefully, but do not enter
3 ,it as a YES in the program.
4 ,  PREVIOUS ILLNESS definition:
4 ,
4 ,          YES
4 ,          NO
4 ,
4 ,     This requires judgment on the part of the practitioner.
4 ,Answer YES if there are items in the history (including
4 ,hospitalizations) which are thought to be pertinent to the
4 ,present illness.  If YES, add details to the back of the data
4 ,sheet.
5 ,  TAKING MEDICATIONS definition:
5 ,
5 ,          YES
5 ,          NO
5 ,
5 ,     Mark YES if your patient is taking medication to relieve
5 ,symptoms associated with this episode only.  Any medication may
5 ,be of importance.  For example, if the patient is taking a non-
```

```
5 ,steroidal anti-inflammatory drug, such as Motrin, this is
5 ,important to know, because this may contribute to ulcer disease.
5 ,Record all medications carefully, but do not enter YES unless it
5 ,is associated with the present illness as described above.
6 , END
```

H54.TXT

```
1 ,  PERIODS definition:
1 ,
1 ,           NOT STARTED
1 ,           CEASED
1 ,           REGULAR
1 ,           IRREGULAR
1 ,
1 ,     Categorize the typical pattern of the patient's menstrual
1 ,cycle.  NOT STARTED is reserved for patient's who have never had
1 ,a period and CEASED indicates that the patient is post-
1 ,menopausal.  REGULAR does not necessarily mean every 28 days,
1 ,some women have slightly longer or shorter cycles.  Additionally,
1 ,women who are ordinarilly regular, who have missed their last
1 ,period may still be marked REGULAR.  IRREGULAR indicates a long
1 ,standing variable interval between periods or bleeding between
1 ,periods.
2 ,  LAST PERIOD definition:
2 ,
2 ,           NORMAL
2 ,           LATE/OVERDUE
2 ,
2 ,     Ask the patient for the date of the last period and the one
2 ,prior to that.  If the last period was more than one week past
2 ,its anticipated date, mark LATE/OVERDUE.  There may be some
2 ,variation from cycle to cycle, but a week is used here as the
2 ,cut-off point.
2 ,
2 ,     Also mark LATE/OVERDUE if the date of the last period
2 ,indicates that the patient is currently overdue for her period.
3 ,  VAGINAL DISCHARGE definition:
3 ,
3 ,     Does the patient have a discharge, not to include vaginal
3 ,bleeding, that is abnormal for her.
3 ,
3 ,     A scanty clear discharge of variable magnitude may be normal
3 ,for some women.
3 ,
3 ,     Note a report of increased flow or frequency of discharge,
3 ,or change in consistency, color or odor.
4 ,  PREGNANCY definition:
4 ,
4 ,           IMPOSSIBLE
4 ,           POSSIBLE
4 ,           CONFIRMED
4 ,
4 ,     IMPOSSIBLE - The patient has not had intercourse recently.
```

```
4 ,It is important to make the patient understand that any
4 ,intercourse however brief, particularly if the 'withdrawal'
4 ,technique of contraception is practiced, may result in a
4 ,pregnancy.  Although no form of birth control is 100% effective,
4 ,mark this response if you are confident that the patient is
4 ,practicing adequate contraception.
4 ,
4 ,     POSSIBLE - The patient is sexually active without adequate
4 ,contraception, or if you suspect improperly used or careless
4 ,contraceptive technique. Strongly supported by late menses.
4 ,
4 ,     CONFIRMED - Strong suspicion that the patient is pregnant by
4 ,history and examination, ideally confirmed by positive laboratory
4 ,tests.
5 ,  FAINT/DIZZY definition:
5 ,
5 ,     These terms may mean different things to different people.
5 ,In the context of this program DIZZY refers to lightheadedness
5 ,rather than the sensation of abnormal motion as is seen in
5 ,vertigo.
5 ,
5 ,     Record both history of actual fainting spells or complaints
5 ,of feelings of lightheadedness or dizziness as part of the
5 ,current illness.  This may indicate hypovolemia from blood loss
5 ,or dehydration or may be secondary to pregnancy.
6 ,  PREVIOUS GYN HISTORY definition:
6 ,
6 ,     Though a thorough gynecological history should be taken and
6 ,recorded, only enter YES for a history of prior sexually
6 ,transmitted disease, salpingitis, pelvic inflammatory disease,
6 ,ectopic pregnancy or gynecological surgery.
7 , END
```

H15.TXT

```
1 ,  TEMPERATURE definition:
1 ,
1 ,     Mark the appropriate range for temperature in degrees
1 ,Fahrenheit.
2 ,  PULSE definition:
2 ,
2 ,     Mark the appropriate range of values in beats per minute.
3 ,  BLOOD PRESSURE (Systolic) definition:
3 ,
3 ,     Mark the appropriate range of values in mmHg.
4 ,  BLOOD PRESSURE (Diastolic) definition:
4 ,
4 ,     Mark the appropriate range of values in mmHg.
5 , END
```

H25.TXT

```
1 ,  MOOD definition:
1 ,
1 ,          NORMAL
1 ,          DISTRESSED
1 ,          ANXIOUS
1 ,
1 ,     Although interpretation of MOOD is very subjective it
1 ,allows insight into the patient's psychological status.  There
1 ,is a subtle difference between DISTRESSED and ANXIOUS.  A
1 ,patient who is experiencing significant physical symptoms (such
1 ,as pain, nausea, vomiting) may be DISTRESSED, but the patient
1 ,who is primarily worried about his illness would be ANXIOUS.
2 ,  COLOR definition:
2 ,
2 ,          NORMAL
2 ,          PALE
2 ,          FLUSHED
2 ,          JAUNDICED
2 ,          CYANOTIC
2 ,
2 ,     FLUSHED, is an important finding but is meaningful only if
2 ,the patient's environment is not uncomfortably warm, which
2 ,might cause flushing normally.  A patient with JAUNDICE will
2 ,have yellowish skin and/or sclerae (whites of the eyes). .
2 ,Patients who are CYANOTIC will present with a blue tint,
2 ,especially around the lips and nail beds.
2 ,
2 ,     In some patients (especially blacks and orientals) these
2 ,findings are difficult to appreciate; sometimes examination of
2 ,the sclera and palm creases is helpful.  If doubt exists, skip
2 ,this category on the data sheet and when entering data into the
2 ,computer.
3 ,  WHITE BLOOD CELL COUNT definition:
3 ,
3 ,     Mark the appropriate range of values.
4 , END
```

H35.TXT

```
1 ,  INSPECTION definition:
1 ,
1 ,           NORMAL
1 ,           VISIBLE PERISTALSIS
1 ,           DECREASED ABDOMINAL MOVEMENT
1 ,
1 ,      Look at the abdomen closely without touching it.  VISIBLE
1 ,PERISTALSIS will appear as wave-like movement occasionally
1 ,accompanied by audible rushes.
1 ,
1 ,      To observe for DECREASED ABDOMINAL MOVEMENT, the patient
1 ,should lie on his back with knees flexed while the practitioner
1 ,holds his hand approximately 1 to 2 inches above the patient's
1 ,umbilicus.  The patient is then asked to raise his belly to touch
1 ,the practitioner's hand.  If the patient has difficulty doing
1 ,this, mark DECREASED ABDOMINAL MOVEMENT.
2 ,  SCARS definition:
2 ,
2 ,           PRESENT
2 ,           ABSENT
2 ,
2 ,      Mark PRESENT if the patient has either surgical scars or
2 ,scars indicating penetrating trauma to the abdomen.
3 ,  GUARDING definition:
3 ,
3 ,           PRESENT
3 ,           ABSENT
3 ,
3 ,      In the context of this program guarding is present when
3 ,there is voluntary tightening of the abdominal wall muscles,
3 ,when the practitioner attempts to palpate.
3 ,
3 ,      This can often be overcome by the gentle persuasion of the
3 ,examiner.  Guarding must be distinguished from involuntary
3 ,tightening of the abdominal muscles and cannot be overcome by
3 ,gentle persuasion and reassurance.  The latter is entered as
3 ,RIGIDITY.
4 ,  RIGIDITY definition:
4 ,
4 ,           PRESENT
4 ,           ABSENT
4 ,
4 ,      Rigidity is the involuntary tightening of the abdominal
4 ,wall muscles which cannot be overcome by the examiner's
4 ,reassurance.  It is an important finding since it often indicates
4 ,peritonitis.  Rigidity must be distinguished from voluntary
```

```
4 ,tightening of the abdominal muscles which can be overcome gentle
4 ,persuasion, which is entered as GUARDING.
5 ,  BOWEL SOUNDS definition:
5 ,
5 ,          NORMAL
5 ,          ABSENT
5 ,          HYPERACTIVE
5 ,
5 ,     This category requires judgment since the difference
5 ,between normal and abnormal bowel sounds may be difficult to
5 ,appreciate.  Use this rule: unless bowel sounds are markedly
5 ,increased or decreased, mark NORMAL.
5 ,
5 ,     Constant gurglings with periods of rushing, high-pitch
5 ,squeaks, or tinkling are HYPERACTIVE.  Constant gurglings alone
5 ,are probably NORMAL.  To mark absent you must listen for five
5 ,minutes hearing no, or only one or two gurglings, mark ABSENT.
5 ,One gurgling sound per minute, is probably normal.
6 ,  DISTENTION definition:
6 ,
6 ,          PRESENT
6 ,          ABSENT
6 ,
6 ,     Distention is a generalized swelling or bloating of the
6 ,abdomen and should be distinguished from a mass or localized
6 ,swelling.
7 ,  MASSES definition:
7 ,
7 ,          PRESENT
7 ,          ABSENT
7 ,
7 ,     A mass is a localized or discrete swelling, visible or
7 ,palpable on examination.  Generalized swelling of the abdomen is
7 ,entered as DISTENTION.
8 , END
```

H45.TXT

```
1 ,  TENDERNESS definition:
1 ,
1 ,     Enter the appropriate response for the location of most
1 ,significant tenderness.  The corresponding area of the diagram
1 ,will be highlighted.
2 , END
```

H55.TXT

```
1 ,  MURPHY'S SIGN definition:
1 ,
1 ,          PRESENT
1 ,          ABSENT
1 ,
1 ,     MURPHY'S SIGN may be elicited as the practitioner hooks his
1 ,fingers under the patient's right costal margin at the
1 ,midclavicular line and asks the patient to inspire deeply.  This
1 ,is best done with the patient lying on his back with his knees
1 ,flexed.  If the patient exhibits reflex inhibition of
1 ,inspiration, sometimes described as a catch, a positive Murphy's
1 ,sign is PRESENT.
2 ,  REBOUND TENDERNESS definition:
2 ,
2 ,          PRESENT
2 ,          ABSENT
2 ,
2 ,     There are three steps to eliciting rebound tenderness:
2 ,     1. Slowly depress the tender area.
2 ,     2. Hold until the pain lessens and the patient
2 ,        relaxes.
2 ,     3. Quickly withdraw to skin level only.
2 ,
2 ,     If the patient grimaces or demonstrates other obvious signs
2 ,of pain upon withdrawal, rebound is present.  The practitioner
2 ,should not have to ask the patient.
2 ,
2 ,     This maneuver should be saved for late in the exam, since
2 ,it may be difficult to get the patient to relax afterward.
3 ,  RECTAL EXAMINATION definition:
3 ,
3 ,          NORMAL
3 ,          MASS FELT
3 ,          LEFT TENDER
3 ,          RIGHT TENDER
3 ,          GENERAL TENDERNESS
3 ,
3 ,     The rectal exam is often neglected.  No abdominal exam is
3 ,complete without the rectal examination.
3 ,
3 ,     Some degree of discomfort is normal during the rectal exam.
3 ,The patient should be verbally prepared for the examination and
3 ,the examiner should be as gentle as possible.  The patient
3 ,should be asked if he feels pain as the practitioner moves his
3 ,finger right, left and centrally.  If there is tenderness on
3 ,both sides, mark GENERAL TENDERNESS even though one side might
```

```
3 ,be more tender than another. (NOTE: directions are in reference
3 ,to the patient's right and left.)
3 ,
3 ,     If the guaiac test, for occult blood in the stool, is
3 ,positive, the patient should be examined for obvious sources such
3 ,as hemorrhoids or a fissure.
4 ,  VAGINAL EXAMINATION definition:
4 ,
4 ,          NORMAL
4 ,          RIGHT TENDERNESS
4 ,          LEFT TENDERNESS
4 ,          CERVICAL TENDERNESS
4 ,          GENERAL TENDERNESS
4 ,          MASS
4 ,          BLOOD (CLOTS)
4 ,
4 ,     Vaginal examination should be performed if there is any
4 ,suspicion of gynecological disease.  This involves inspection,
4 ,bimanual internal examination and speculum examination.
4 ,
4 ,     Unilateral tenderness may be indicative of ovarian or tubal
4 ,involvement.  Cervical tenderness, particularly with cervical
4 ,motion may indicate ectopic pregnancy or PID.  Palpation with one
4 ,hand on the lower abdomen above the pubis and two fingers of the
4 ,other hand internally should help identify any masses.
4 ,Microscopic examination of any vaginal discharge may be useful,
4 ,but does not effect this program's diagnosis.
5 , END
```

Appendix D
Help File Source Listings

HABGY.ASC

Select Appropriate Female Database

At this time, you select the appropriate female database to use of the two available. The computer will high-light its recommendation, but you may override its decision.

The program recommends that you select the GYNECOLOGICAL DATABASE if the patient's age is between 15 and 50 and she presents with central, lower abdominal, or flank pain at present. The GENERAL DATABASE is suggested for all other patients.

NOTE - Of the thousand cases comprising the gynecological database, very few patients presented with upper abdominal pain. Of those very few patients, most had a UTI with both upper abdominal pain and flank pain.

HELP.ASC

WELCOME

Welcome to the world of computer-assisted diagnosis for abdominal pain. The following instructions should enable you to run me with little difficulty. Please remember that I am to be used as an aid only. I cannot take your place or make decisions for you.

I am designed to aid you in making a diagnosis of abdominal pain. I only look at the six most common causes of abdominal pain in the submarine crew population. Based on the answers you give me I will then give the probability for each of the causes I know. You will then be able to change any of the answers if you made a mistake when answering the questions.

I can reliably aid you in differentiating the 5 illnesses which represent both the most common and the most serious causes of acute abdominal pain.

In addition, a 6th category, non-specific abdominal pain, is intended to represent those cases which are non-surgical, not life-threatening, and, therefore, not reasons for evacuation.

IMPORTANT- Not all diseases causing acute abdominal pain are considered. Input of symptom complexes associated with other diseases will result in my diagnosis of one of the 6 categories most closely resembling that disease.

YOUR JUDGMENT MUST TAKE PRECEDENCE when any doubt exists. I do not have the capability to think or make the subjective evaluations which are so important in medical diagnosis.

When using the program, be sure always to look at the bottom line. This will usually tell you which commands you may use. Normally you will use the up, down, left, and right arrows to move the cursor in their respective positions.

Press the letter 'N' to take you to the next page. Press the letter 'P' to take you to the previous page, i.e., the one you just left. Press the RETURN/ENTER key to select your response which is highlighted by the cursor. The key is located on the right side of the keyboard, usually above the right shift key. Press the letter 'X' to exit the section and return to the previous menu. Press '?' (the question mark) to receive help.

If you are in the history or physical exam section, pressing the '?' will give you the definition of the question corresponding to the high-lighted response and pressing the TAB key will take you to the first response of each question. Try it when you get there. It can save you time.

HPOO.ASC

Options:

Select the response corresponding to your diagnosis. If your diagnosis is not listed, select OTHER . You will then be asked to enter your diagnosis. Type in your diagnosis followed by pressing the ENTER/RETURN key.

If OTHER was not selected then your response will be compared to the computer derived diagnosis. The computer will then state whether or not it agrees with you and if it doesn't, it will then check for key questions on which you and it disagreed. If any are present it will then print them.

HP10.ASC

Main Summary Page Help

This page is the main summary page. The graph to the left shows the computed probability for each diagnostic category. The tallest bar corresponds to the most likely diagnosis. The program is "most sure" of the diagnosis when it is greater than 90% and "less certain" when the probability is less than 90%. To help keep this in mind, a line is drawn across the graph at the 90% level.

The date and time of the exam are listed in the upper right hand corner. If the case is real, the patient's SSN will also be listed there. The type of case - real or simulated - is shown in the lower right hand corner along with the name of your vessel.

OPTIONS

1. CHANGE INPUT DATA - Choose this option if you need to go back and change any of your responses.

2. ANOTHER DIAGNOSIS - Choose this option to enter a new case, either real or simulated. Note - The current case is saved on disk and then all responses are erased from memory. This is equivalent to restarting the program.

3. DISPLAY TREATMENT - Select this option to access treatment protocols. Protocols exist for each diagnostic category.

4. DISPLAY H & P - Select this response if you want to list your history and physical exam symptom entries. History symptoms are given on one page and physical exam findings are given on another page. If you have a printer connected to the computer, you can get a hard copy of the symptom entries (while they are on the computer screen) by pressing the Shift key and the PRTSC key at the same time.

5. END INTERACTION - Choose this option to exit the program. You will be returned to the operating system of the computer so that you may run other programs. NOTE - If you desire to have the SF-600 printed for a patient entered into the program, you must have selected either this response or ANOTHER DIAGNOSIS to save the patient's data on the disk for later retrieval by the SF-600 printing program.

6. DISPLAY ABD DX'ES / DISPLAY GYN DX'ES - This response toggles back and forth between the regular abdominal disease data base and the gynecological database. If the GYN database is displayed, DISPLAY ABD DX'ES will be on the menu. If the non-gynecological abdominal pain graph is displayed, then DISPLAY GYN DX'ES will be on the menu.

   NOTE - This option is displayed only if the patient is female.

HP11.ASC

Options:

Choose the response for which you would like the treatment protocol. To exit this menu, choose the last response, EXIT THIS DISPLAY .

Initially the response pointed to is the diagnosis chosen by the computer. Use the arrow keys to select another choice.

Remember that pressing the SHIFT key and the PRTSC key at the same time will give you a copy of the current display screen printed on the printer (if you have one connected). This will give you a copy of the treatment plan page. You will need to check your specific computer manual for the PRTSC key since it's location varies from machine to machine.

HP12.ASC

Main Options:

1. REAL CASE - Select this response if you have an honest-to-goodness patient that you want the computer to diagnosis.

   NOTE- Real Cases are stored so that the data you entered can be used to print out the SF-600 entry for the patient. Only Real Cases are stored in this manner, so anytime you have a real patient be sure to mark him as such.

2. SIMULATED CASE - Select this response for training only. If you want to try different responses to see how the probabilities vary with changes in the history or physical, this is the case option to use. If you have a real patient, you can go ahead and enter him as a Real Case, so that his data is stored and then select a Another Diagnosis on the summary page so that you return to this menu and then can select Simulated Case to play around with his responses. Remember that although data is stored on the disk, it is not stored so that an SF-600 entry can be made.

3. TRAINING PROGRAM - Select this response if you desire to run the training program. This program will first give you a patient narrative. Using this, you then can fill out a data sheet, make your diagnosis, and then compare it to the computer derived diagnosis. This program is for your own training. No record is kept on how often the program is used or your scores.

4. LAST REAL CASE - Select this response and the computer will put the latest real case into memory. This will allow you to review which responses were given at the time. Also, if you have a real patient for whom you are doing serial exams because of a confusing or early presentation, you can call in the previous exam and update his change since the previous exam instead of having to reenter all of the history and physical each time. Remember to change the time of the exam so that you can keep the different cases separate.

5. LAST SIMULATED CASE - Select this response to have the computer get the last simulated case and place it in memory. This allows you to come back to the previous simulated case at a later date and continue with simulation.

6. INSTRUCTIONS / HELP - Select this response for much more in-depth instructions for using the Abdominal Pain Diagnosis Program in general. This selection includes what keys to use and how to move around to different parts of the program.

7. GENERATE SF600 - Select this response to run the SF600 generation routine. If your computer does not have much installed memory, or if

you have many memory-resident programs installed in the system, the routine may cause the computer to "crash" or "lock up" when you attempt to use it while in this program. If that happens, do not try to run the SF600 generation routine from this program until you either add more memory to the computer or you remove some of the memory-resident programs as appropriate.

8. DISPLAY TREATMENT - Select this option to access treatment protocols. Protocols exist for each diagnostic category considered by the program. Both male and female treatment protocols are shown here.

9. EXIT PROGRAM - Select this response to leave the program. No data will be stored at this exit.

HP13.ASC

OPTIONS:

1. GO TO HISTORY PAGES - This area consists of several pages dealing with the patient history. Here is entered all the data concerning the patient's present and past history relevant to the computer. There are 4 pages for male patients and 5 pages for female patients.

   NOTE - If on the first page, pressing 'P' will return you to the data entry menu. If on the last page, pressing 'N' will return you to the data entry menu. 'Q' or the 'Esc' key will return you to the data entry menu.

2. GO TO PHYSICAL EXAM PAGES - This area consists of 5 separate pages dealing with the patient's physical exam. The results of the physical exam are entered here. See NOTE above.

3. MAKE DIAGNOSIS - This selection will take you to the diagnosis area to view the computer's diagnosis. If you do not answer enough questions in the history and physical sections the computer will not have information to make a reasonable diagnosis and so the computer will not allow you to proceed until you have gone back and answered enough of the program.

   If you have answered enough questions you will then be asked for your diagnosis. You must enter your diagnosis before the computer will show you it's computed diagnosis. Next comes the summary page where you may look at a graphic representation of the computed probabilities, treatment protocols, history and physical summaries, etc.

   NOTE - If you chose the last real or simulated case, you will not be asked to give your diagnosis before going on to the summary unless you change any answers to any history, physical exam, or age questions.

4. GO TO SSN/AGE/TIME PAGE - Normally you probably will not use this selection if you have selected a real or simulated patient since you went through that page before getting to this menu. This page allows you to change the age, date, time, and SSN of the patient. Note that you do not need to change the SSN for a simulated patient. It will be set to 000-00-0000.

5. RETURN TO MAIN OPTION PAGE - This response will return you to the Main Option Page. Choose this option if you entered the Data Entry Option Page by mistake.

HPT10.ASC

TRAINING MAIN SUMMARY HELP

This is the training main summary page. It is very similar to the summary page on the ABDX program. The graph to the left shows the computed probability based on your data responses for each category. The bar graph shows pictorially the relative probability for each diagnostic category.

> Note - Since the display is based only upon the information you entered, the actual training case diagnostic probabilities will be correct only when you have entered all of the responses correctly.

For a real patient, you should be more sure of the diagnosis if the computed probability is greater than 90% and you should be less certain of the diagnosis if the probability is less than 90%. To help you keep this in mind, I have drawn a line across the graph at the 90% probability level.

The patient's SSN and date and time of the exam are listed in the upper right hand corner. Your score will be listed below the date. This score is only for your own use. No record of it is kept. The score is based on the ratio of your correct responses to the actual case data responses. The maximum score is 100% which means that your entered responses agreed with all of the actual responses of that particular case. The minimum score will be above 70, since you will not be able to continue to the summary page if you have too many incorrect responses. The type of case and case number will be shown in the lower right hand corner. Also, the name of the vessel to which the program is assigned will be shown in the lower right hand corner.

If you have a printer capable of printing graphics connected to the computer and the GRAPHICS.COM program has been installed so that a screen dump may be obtained, then you can get a copy of the summary page printed on the printer by pressing the CTRL key and the PRTSC key at the same time while the summary page is shown on the screen of the computer. If you do not understand this paragraph that is OK. This paragraph deals only with getting a printed copy of the summary screen including the bar graph. If you desire to keep a copy of the summary page, ask someone that has had experience with the IBM-compatible computers to help.

OPTIONS

1. SHOW MISSED ITEMS - Choose this response to see which items you missed. Those with the word "(omitted)" beside the item are correct responses which failed to select. All others displayed are responses which you should not have chosen.

2. CHANGE INPUT DATA - Choose this category if you need to go back and change any of your responses. Your case will not be stored on the disk if this option is selected.

3. ANOTHER CASE - Choose this response if you desire to try a new training case. Note - all responses are cleared when this response is chosen. This is equivalent to restarting the program.

4. DISPLAY TREATMENT - Treatment protocols are shown here. Protocols exist for each diagnostic category. You may select any of those.

5. DISPLAY H & P - Select this response if you desire to have your responses listed from the H & P. All of your responses from the history are listed on a page and those from the physical exam are listed on another page. If you have a printer connected to the computer on which I am running then you can get a copy printed by pressing the CTRL key and the PRTSC SC key at the same time while looking at each of the pages on the screen.

6. END INTERACTION - Choose this response if you are finished. You will be returned to the operating system of the computer so that you may run other programs.

HPT12.ASC

GENERAL INFORMATION

Welcome to the Abdominal Pain Diagnosis (ABDX) training program. This program is designed to help you to use the ABDX Program more effectively by allowing you to solve imaginary cases. The format for this program has been kept as close to that of the ABDX Program as reasonably possible. On similar screens, the training version will have double-lined boxes while the main ABDX displays will have single-lined boxes. Also, the screen heading colors will differ (if you have a color monitor).

As with the main ABDX Program, the basic instructions are listed at the bottom of each page of text on this screen. If you need more help, you may press the question mark " ? " key for more information in most cases (as you did to get here).

MAIN OPTIONS:

1. EASY CASE NARRATIVE - Choose this response for a straightforward patient narrative. You will be prompted to enter a case number. Currently there are 50 cases. The narrative will be presented in 2 pages: a history page and a physical exam page. If you have a printer connected to this computer you can get a printed copy of the case narrative by pressing the PRTSC key on your computer. This will cause the printer to print a copy of this screen (make sure your printer is on).

2. HARDER CASE NARRATIVE - Choose this response for a less straightforward patient narrative. You will be prompted to enter a case number. These are the same cases as those above. Only the narrative is different.

3. ENTER DATA - Choose this response when you are ready to enter the data on a case. If you have just finished reviewing a case, it will be assumed that that case is the one you want. However, you will be asked to confirm that. If you desire to use another case, enter the new case number. If you have not reviewed a case, then you will be asked to enter the desired case number. If you desire to return to the main option menu, when you are asked to enter a case number, just press the RETURN/ENTER key. Once you have selected a case under this option, you will be taken to the ABDX Program where you will enter the data. If you miss too many items for that particular case, you will not be able to continue to the computerized diagnostic section. This is to keep people from attempting to use real patients on the training program.

HSF00.ASC

SF600 Output Selection Page

Use the cursor keys to high-light the method to print the SF-600. Press ENTER/RETURN to select the high-lighted response.

Select CONSOLE to print the SF-600 only on the screen.

PRINTER will print the SF-600 on your printer. The printer routine is specifically designed to use actual SF-600 paper. The routine will automatically double space and use the margins found in the SETUP.DAT file. See your user's manual for information on changing the default margins if these are unsatisfactory.

FILE prints the SF-600 to a file, where you can later modify it with your favorite word processor.

Appendix E
Treatment Protocol Listings

TX1.ASC

Acute Appendicitis (APPEND)

1. **TREATMENT PROTOCOL AT A GLANCE** (see detailed discussion of treatment below; this section is for quick reference).

Definitive treatment of acute appendicitis is surgical.

Notify command of need for immediate MEDEVAC.

The treatment goals pending MEDEVAC:

a. bed rest
b. intravenous antibiotics
c. intravenous fluid and electrolyte replacement
d. intestinal rest-NPO
e. abdominal ice packs
f. analgesia

a. Place patient at strict bedrest with head of bed slightly elevated. Monitor vital signs, fluid intake and output, and progress of symptoms and examination.

b. Appropriate antibiotic therapy is crucial:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg).

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs."

c. Start IV D5NS or D5LR at 150 ml/hr and adjust rate as needed to maintain adequate hydration. Monitor urine output as an indicator of hydration status.

d. Make patient NPO. Place NG tube to low-intermittent suction.

e. Abdominal ice packs intermittently as tolerated.

f. Analgesics are usually contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness.

2. DISCUSSION

Acute appendicitis results from a bacterial invasion of the appendix with inflammation. Incidence is highest in adolescents and young adults, peaking between ages 15 and 24. Typically, the pain is initially poorly localized in the periumbilical region. This pain represents irritation of the lumen of the appendix itself. Gradually the pain shifts to the right lower quadrant, becoming constant and discrete as the inflamed appendix begins to irritate the surrounding peritoneum. Anorexia, fever, and diminished bowel sounds are usually noted; nausea, vomiting, and diarrhea are variable. Temperature, pulse, and respiratory rate may be only slightly elevated unless perforation occurs.

A history of right lower quadrant pain and a finding of right lower quadrant tenderness and guarding are the most important indicators in acute appendicitis. Since the appendix may lie dorsally, the rectal examination is also extremely important. The white blood cell count is usually elevated above 10,000 with a preponderance of neutrophils. The urinalysis may also reveal an increase in WBC's. There are no specific laboratory tests.

3. DIFFERENTIAL DIAGNOSIS

Other illnesses that produce findings that may mimic acute appendicitis and which should be considered are:

a. mesenteric adenitis
b. Meckel's diverticulitis
c. diverticulitis
d. regional enteritis
e. inguinal hernia
f. ureteral calculus
g. perforated ulcer
h. acute cholecystitis

a. Mesenteric adenitis can be indistinguishable from acute appendicitis, but the pain may not be as well localized and the white blood cell count may contain an increased fraction of lymphocytes. Initial medical management is the same as for appendicitis.

b. Meckel's diverticulitis may also closely mimic the presentation of acute appendicitis if the diverticulum is in its usual position in the right lower quadrant. Meckel's diverticulitis may also lead to other intra-abdominal complications including bowel obstruction and G.I. bleeding. The definitive treatment, as with appendicitis, is surgical; initial medical management is the same as outlined below for appendicitis.

c. Diverticulitis confined to the cecum can be hard to distinguish from appendicitis. However, it more commonly involves the descending (left) colon. These symptoms are often intermittent and less severe, but may be acute. This condition has been described figuratively as a "left-sided appendicitis." The presence of diverticular disease does not preclude development of acute appendicitis concomitantly. The initial medical treatment is the same as for appendicitis.

d. Regional enteritis attacks are usually preceded by bouts of crampy abdominal pain and diarrhea. An attack of regional enteritis involving the terminal ileum, located in the right lower quadrant, can be indistinguishable from acute appendicitis. The pre-existence of regional enteritis does not preclude development of acute appendicitis concomitantly. Initial medical management is the same as for appendicitis. Additional treatments include steroids and analgesics, but these should not be used unless the diagnosis is firm and a Medical Officer is consulted.

e. An inguinal hernia can be easily distinguished by its sharp, localized pain which radiates into the testicle and by a palpable lump in the inguinal canal accentuated by valsalva. Treatment is symptomatic pending surgical consultation ashore. Reduction should be attempted. If the hernia is not reducible (i.e. incarcerated), it will be necessary to follow the patient closely. If the hernia becomes strangulated, the patient may present with the symptoms of an acute abdomen. In this latter case use the medical management outlined for appendicitis.

f. Ureteral calculus - see RENAL COLIC

g. Perforated ulcer - see PERFORATED DUODENAL ULCER

h. Acute cholecystitis - see CHOLECYSTITIS

4. TREATMENT OF ACUTE APPENDICITIS

Definitive treatment of acute appendicitis is surgical. Medical management can, however, avert or delay serious morbidity in many cases. Treatment includes bowel rest and aggressive antibiotic therapy. The body's own primary defense consists of omental migration toward the inflamed appendix in an attempt to surround and confine the infectious process.

Arrange immediately to MEDEVAC the patient. The treatments outlined are intended to stabilize the patient while awaiting evacuation. However, MEDEVAC's are not always immediately available or operationally feasible. Always remain prepared to care for the patient for a prolonged period of time. The treatments as outlined are written for this contingency.

The treatment regimen should consist of the following:

a. bed rest
b. intravenous antibiotics
c. intravenous fluid and electrolyte replacement
d. intestinal rest - NPO
e. abdominal ice packs
f. analgesia

a. Place patient at strict bed rest with head of bed slightly elevated (semi-Fowler's position). Monitor vital signs, fluid intake and output, and progress of symptoms and examination.

b. Appropriate antibiotic therapy is crucial. Bowel flora will be predominantly gram-negative organisms, enterococci (Streptococcus fecaelis), and anaerobes. Specific intravenous antibiotic regimens are as follows:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patients with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs."

c. Intravenous fluids are utilized to maintain fluid and electrolyte requirements in the face of increased need and decreased intake. Dextrose 5% and Lactated Ringer's solution (D5LR) should be started at 150 ml/hour. Additional fluid replacement will be needed for the additional losses caused by fever, diaphoresis, emesis, diarrhea, and nasogastric suctioning. Fluid and electrolyte replacement must keep pace with losses to prevent dehydration and electrolyte imbalance. Monitor the urine output to maintain at least 1000 ml/day (or 1/2 ml per kg body wt per hour) and increase fluid intake as needed. Urine specific gravity and hematocrit are also good indicators of the adequacy of hydration.

d. Intestinal rest is accomplished by making patient NPO (nothing by mouth) and by nasogastric suctioning. Low-intermittent suction (1-3 psi) is most effective. Irrigation of the tube with small volumes of saline may be needed periodically to maintain patency.

e. Abdominal ice packs applied to the right lower quadrant may decrease inflammation and offer some psychological benefit to the patient. If used as an adjunct to the above treatment regimen, it should be continued only if subjective relief is obtained. Use intermittently to avoid a reduction in the rate of diffusion of antibiotics to the region.

f. Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness. However, it may become necessary to offer the patient some pain relief by judicious use of morphine or Demerol.

5. USUAL COURSE WITH TREATMENT

Response to the therapeutic regimen will be slow initially. The patient's discomfort will resolve over the first 48 hours, temperature and pulse will remain elevated for 48-72 hrs, and the leukocytosis will continue to rise well beyond 72 hours as the inflammatory response peaks and begins to resolve. By 96 hours the patient's symptoms should have abated, and he should be ambulatory. A clear liquid diet should be tolerated when bowel sounds return and the pain resolves. At this point, intravenous fluids can be decreased or discontinued as appropriate. Antibiotics must be continued for at least 10 days.

6. COMPLICATIONS AND THEIR MANAGEMENT

Perforation, diffuse peritonitis and/or abscess formation are the principal complications of concern. The incidence of these complications should be significantly reduced in patients expeditiously treated as outlined above.

a. Perforation and diffuse peritonitis will result in a rapid deterioration of the patient's condition as indicated by:

1) progressive tachycardia, fever, and tachypnea

2) a diffusely distended, tender, silent, rigid abdomen despite nasogastric suction,

3) difficulty maintaining adequate hydration and electrolyte balance despite vigorous fluid replacement.

These signs can develop with frightening rapidity during any stage of management of an acute appendicitis and are further indications for immediate MEDEVAC. Continue fluid and antibiotic therapy while observing for indications of shock. Septic shock is treated with supportive measures in an effort to maintain blood pressure and adequate tissue perfusion. Place the patient in Trendelenburg and use fluid boluses as necessary.

b. A right lower quadrant mass may indicate the formation of an abscess around the inflamed appendix. A mass may develop, however, as the inflamed appendix is walled off by omentum or adjacent viscera, and thereby represent resolution rather than a complication. If the mass is accompanied by improvement, an abscess is less likely. If the fever and abdominal examination fails to improve, an abscess may be forming. Fluids and antibiotics should be continued with close observation for signs of perforation. Abscess formation is not a critical event and can be managed surgically at a later date as long as medical management is continued.

TX2.ASC

Non-Specific Abdominal Pain (NONSAP)

The category of NONSAP is reserved for causes of abdominal pain which are non-surgical and not life threatening, and are amenable to symptomatic treatment. This category is not meant to cover all maladies not included in the remainder of the program, but, rather, is meant to include those illnesses which are likely to be self-limited and, therefore, not reasons for MEDEVAC. NONSAP includes some specific diagnoses requiring specific treatments. In addition NONSAP is used for those patients whose symptoms are non-specific and resolve spontaneously without a diagnosis being made.

Peptic ulcer disease (PUD) is included in this section. Its diagnosis and treatment are discussed below. One of the complications of PUD, perforated duodenal ulcer, is treated as a separate diagnosis and may be reviewed from the REVIEW TREATMENTS menu.

Peptic Ulcer Disease (PUD)

1. DISCUSSION

PUD will affect 1 person in 10 during his lifetime. The stomach normally secretes a highly corrosive acid to aid in the digestion of food. The stomach also has defenses to protect itself from these secretions. A disruption of this balance between acid secretion and the protective mechanisms can result in damage or destruction of the stomach lining. Emotional stress, cigarette smoking, alcohol abuse, and many drugs have been cited as risk factors for PUD. Some drugs may alter the stomach's protective lining and allow damage to occur. Family history may play a role. Contrary to popular belief a spicy diet plays no role in the development of ulcers. The stomach's normal secretions are far more acidic than any spice or food that the patient could eat. Ulcer disease tends to be chronic with exacerbations and remissions. The duration of symptoms will vary, most cases resolve without complication if properly treated.

The pain of ulcer disease is often described as a steady, gnawing, burning ache in the epigastrium radiating laterally, usually to the right. The pain characteristically occurs 1/2 to 2 1/2 hours after meals and is rapidly relieved by ingestion of food or antacids. The temporal relationship to meals and the history of relief of the pain with food are important diagnostically. Also characteristic is the history of pain in the middle of the night and the absence of pain upon awakening in the morning.

A history of melena or black, tarry stools is indicative of upper gastrointestinal bleeding. Ask the patient specifically about changes in their stool or bowel habits. Patients often fail to volunteer this history.

The physical examination may be non-specific except for tenderness corresponding to the location of the pain in the epigastrium. Test the stool for occult blood and draw blood for a hematocrit to rule out bleeding.

2. DIFFERENTIAL DIAGNOSIS

The vague initial symptoms of a new ulcer can sometimes cloud the diagnosis. Several other illnesses may produce similar pain in the epigastrium. A partial list of the more common ones follows:

a. gastritis
b. gastroenteritis
c. cholecystitis
d. pancreatitis
e. hepatitis

a. Gastritis is a diffuse more superficial inflammation and degeneration of the gastric mucosa. History will help differentiate it from PUD in that the symptoms are usually aggravated by food rather than relieved. Initial management is the same as for PUD.

b. Gastroenteritis may initially present as a crampy pain in the epigastrium but will quickly involve symptoms of nausea, vomiting, and diarrhea, thereby differentiating it from ulcer disease. Gastroenteritis is usually viral and accompanied by symptoms of fever, myalgias, and malaise. The course is usually self limiting and benign, but bowel rest and dehydration may necessitate IV hydration.

c. Cholecystitis - see CHOLE

d. Pancreatitis is associated with duodenal ulcer, cholecystitis, trauma, and alcoholism and will present as a boring epigastric pain radiating to the back. Signs and symptoms of peritoneal irritation are sometimes present. Antacids will not relieve the pain of pancreatitis. Food and particularly fatty foods will exacerbate the pain. Treatment should include bowel rest, with absolutely no fatty foods or alcohol permitted.

e. Hepatitis may initially present with a dull epigastric or right upper quadrant pain. The patient may complain of mild constitutional, flu-like symptoms or may feel quite well. Low grade fever may be noted. Jaundice may be noted, pay particular attention to the sclerae. "Coca-cola" urine is often the first sign. The liver edge is often apparent and may be tender on gentle palpation. Treat patient with bedrest and symptomatically. Though most causes of hepatitis are not highly infectious, isolation is advisable. Preventing contact with the patient's secretions is adequate. Some patient's may have a fulminant course and some rare cases are fatal. Those cases that require MEDEVAC should be self-evident.

3. TREATMENT OF PEPTIC ULCER DISEASE

The mainstay of ulcer therapy is the reduction of gastric acidity through neutralization and decreased secretion. This is accomplished with a regimen of antacids and a relatively new class of drugs, the H-2 blockers, that inhibit acid production. Food adequately buffers stomach acidity for approximately 90 minutes, after which the stomach is empty and secreting un-neutralized acid. Antacids taken 1 hour after meals will effectively buffer acid for an additional 2 hours.

A liquid antacid (Maalox, Mylanta) and Cimetidine (Tagamet), an H-2 blocker, are used according to the following regimen:

a. ACUTE CASE

Liquid antacid 30 ml at 1 and 3 hours after meals and once before bedtime. This regimen should be continued for 2 months.

Cimetidine (Tagamet) 300 mg PO qid for at least 4-6 weeks.

b. OCCASIONAL PAIN: 15-30 ml of liquid antacid PRN.

c. RECURRENCE: Treat as an acute case.

The diet should be modified to decrease fatty foods, as they stimulate acid secretion. Alcohol, caffeine, and smoking should be limited if not eliminated. Aspirin and other drugs in the non-steroidal anti-inflammatory drug (NSAID) class (Motrin, Tolectin, Indocin, etc.) should not be used except as absolutely necessary.

4. USUAL COURSE WITH TREATMENT

Most patients treated with the above regimen will have at least partial relief of their pain immediately. A gradual diminution of all symptoms within 2 to 4 weeks follows. Recurrences exacerbated by life-style (i.e. diet, alcohol, tobacco) are common and should be treated as acute cases.

5. COMPLICATIONS AND THEIR MANAGEMENT

The most common complications of peptic ulcer disease are intractable pain, hemorrhage, and perforation. These complications develop in 10-20% of symptomatic cases. Interestingly, 20-30% of all hemorrhage and perforation cases occur with no antecedent history of ulcer symptoms. Less often, the complications of obstruction and intractable pain will arise, requiring definitive (surgical) treatment.

Intractable pain may be an indication for surgery. The patient should be transported to a shore based facility when possible.

Hemorrhage will be evident as either vomitus which is colored bright red (hematemesis) if the bleeding is profuse or containing older, partially digested blood having a "coffee grounds" appearance. Black, tarry stool (melena) may occur if the bleeding is slow and subclinical.

Hematemesis is a medical emergency and can result in hypovolemic shock or death. It should be treated with the aim of maintaining the patient's blood pressure and fluid and electrolyte balance while preparing him for MEDEVAC. Start the IV with Normal Saline or Lactated Ringer's at maintenance rate (125-150 ml/hr). Be prepared for massive infusions to counteract the potentially massive blood loss from an ulcer which has eroded an artery within the duodenal wall. Additional fluid replacement should be calculated to compensate for the extra fluid losses from temperature elevation, diaphoresis, emesis, diarrhea, and nasogastric suctioning as well as estimated blood loss. Fluid and electrolyte replacement must keep pace with losses to prevent dehydration and electrolyte imbalance. Serial hematocrits will permit a rough estimate of the magnitude of the bleed. Remember, however, bleeding does not lower the hematocrit until the volume has been replaced either by IV solutions or the body's own fluid reserve. Monitoring of the urine output is a good indicator of the adequacy of hydration. If the bleeding is profuse, an attempt should be made to stop the hemorrhage using iced saline lavage. The patient should be kept NPO and a nasogastric tube placed. Iced saline lavage should be administered via the nasogastric tube until the lavage is clear.

Although melena is a cause for concern and investigation at the earliest convenience, it is not life-threatening and can be treated with the acute case protocol above and followed with serial hematocrits. Melena supports the diagnosis of PUD, but is not diagnostic in and of itself.

Perforation and obstruction should be managed by the protocols for PERFDU and SMBOBS, respectively.

TX3.ASC

Renal Colic (RCOLIC)

1. **TREATMENT PROTOCOL AT A GLANCE** (see detailed discussion of treatment below; this section is for quick reference).

Analgesia:

Demerol 50-150 mg IM q 3-4 hours

Hydration:

D5NS at 150-200 ml/hour

2. DISCUSSION

Renal colic is caused by the passage of a calculus or stone, through the renal pelvis or ureter or other cause of acute ureteral obstruction. It presents with the sudden onset of severe, sharp pain in the flank and upper abdomen with radiation to the lower abdomen, scrotum (or labia) or thigh area. The pain is intermittent, but without complete remission between attacks. This is described as wave-like fluctuations in the pain or colic. The pain typically is described as the worst ever experienced by the patient and has been said to be worse than child-birth. A history of milder symptoms or of past stones is often elicited. Nausea and vomiting may occur.

The physical examination reveals costo-phrenic angle and flank tenderness without rebound. Fever and chills as the stone passes are not uncommon. Gross or microscopic hematuria are usually evident. Pyuria or crystalluria may be noted. Some stones will be visible on plain films of the abdomen, others only on IVP.

3. DIFFERENTIAL DIAGNOSIS

Renal calculi are not always associated with classic flank pain or referred pain. Rule out other causes of intra-abdominal pain, especially the potentially surgical cases.

4. TREATMENT OF RENAL CALCULI

Treatment of uncomplicated renal colic consists of analgesics and hydration.

a. Large doses of narcotics may be necessary. The usual drug of choice is Demerol in the following dosage:

Meperidine (Demerol) 50-150 mg IM every 3-4 hours as needed.

Doses at the higher end of this range are typically needed. The subcutaneous route may be used, but will be slower acting. Nausea and vomiting are common and may be aggravated by the Demerol. For this reason Demerol is often given concomitantly with an anti-emetic. Most anti-emetics will potentiate CNS depressants, and will, therefore, necessitate lower doses of narcotics.

> CAUTION : Whenever narcotics are used, the opiate antagonist naloxone (Narcan) should be kept available. If the patient should demonstrate excessive CNS depression, respiratory depression, or hypotension, 1-2 ampules (0.4-0.8 mg) of naloxone should be given IV and the A-B-C's of resuscitation employed as necessary. If the episode is due to narcotic overdose, consciousness will return within a minute or two. The pain will likely return and can be re-treated. The maximum dose of Demerol may then have to be judiciously overstepped. The duration of action of naloxone is less than that of Demerol. Patients treated with naloxone for opiate overdose must be monitored for 3 or more hours.

b. Fluid therapy is instituted to increase urine volume and help the stone pass. D5NS at 150-200 ml/hour is used.

If nausea and vomiting are problematic, the patient should be kept NPO and, if severe enough, may require placement of a nasogastric tube.

Every effort should be made to recover the stone when it's passed. This can be accomplished by having the patient urinate through a strainer or gauze sponges. The stone should be saved for analysis in conjunction with an intravenous pyelogram (IVP) performed ashore. Most cases can be treated adequately underway.

5. USUAL COURSE WITH TREATMENT

The stone will usually pass spontaneously within minutes or hours. This will be marked by the sudden cessation of the pain. Hematuria, dysuria, and a dull ache in the flank may persist for a day or two.

Encouraging the patient to drink a sufficient quantity of liquids to keep his daily urine volume over 2500 ml has been thought to decrease the incidence of stone recurrence. The patient should receive a urology consult to determine factors which might predispose him to form another stone.

Anticipate submarine disqualification if applicable.

6. COMPLICATIONS AND THEIR MANAGEMENT

Renal stones which become impacted and do not pass spontaneously must be removed surgically. Renal colic lasting over 24 hours is unusual and is an indication for MEDEVAC. Pain lasting this long will be very exhausting for the patient.

Patients may develop urinary tract infections as a result of a calculus. This presents as dull residual flank pain, dysuria, and residual hematuria with urinary frequency, hesitancy and fever. Treatment is :

> Trimethoprim-Sulfamethoxazole (Bactrim, Septra) two tablets (one if double strength DS preparation) PO twice daily for 7 days.

If Sulfa allergic:

> Doxycycline (Vibramycin) 100 mg PO twice daily.

TX4.ASC

Perforated Duodenal Ulcer (PERFDU)

1. **TREATMENT PROTOCOL AT A GLANCE** (see detailed discussion of treatment below; this section is for quick reference)

Definitive treatment of perforated duodenal ulcer is surgical.

Notify command of need for immediate MEDEVAC.

The treatment goals pending MEDEVAC:

a. bed rest/observation
b. antibiotics
c. intravenous fluid and electrolyte replacement
d. intestinal rest.

a. Place patient at strict bedrest with head of bed slightly elevated. Monitor vital signs, fluid intake and output, and progress of symptoms and examination.

b. Appropriate antibiotic therapy is crucial:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

c. Start IV D5NS or D5LR at 150 ml/hr and adjust rate as needed to maintain adequate hydration. Monitor urine output as an indicator of hydration status.

d. Make patient NPO. Place NG tube to low-intermittent suction.

Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness.

2. DISCUSSION

Duodenal perforation usually presents with a sudden, steady, severe pain in the right upper quadrant or epigastrium. The patient will lie still in bed with his legs flexed since the pain is aggravated by movement or coughing. He may vomit initially although sustained vomiting is not characteristic. The patient will usually relate a past history of ulcer disease or ulcer symptoms.

Examination of the abdomen reveals a rigid, quiet abdomen with marked tenderness and rebound tenderness. The pain may radiate or migrate depending on the location of the gastric contents which have been allowed to spill into the peritoneal cavity. Peritonitis in the right lower quadrant by this mechanism may cloud the diagnosis and suggest appendicitis. Normal dullness to percussion overlying the liver may be lost due to free air overlying the liver in the supine position.

Perforation may occur adjacent to the pancreas or gall bladder, allowing gastric contents to irritate these organs and possibly cause pancreatitis or cholecystitis and their inherent signs and symptoms.

3. DIFFERENTIAL DIAGNOSIS

Other illnesses that produce symptoms mimicking duodenal perforation and which should be considered are:

a. acute pancreatitis
b. acute cholecystitis
c. acute appendicitis
d. diverticulitis.

a. Acute pancreatitis is associated with duodenal ulcer, cholecystitis, trauma, and alcoholism and will present as a boring epigastric pain radiating to the back. Signs and symptoms of peritoneal irritation are sometimes present. Antacids will not relieve the pain of pancreatitis. Food, particularly fatty foods, will exacerbate the pain. Treatment should include bowel rest with absolutely no fatty foods or alcohol permitted.

b. Acute cholecystitis - see CHOLECYSTITIS
c. Acute appendicitis - see APPENDICITIS

d. Diverticulitis may be present anywhere along the course of the colon including both upper quadrants and the epigastrium. It is more common, however to involve the descending (left) colon. These symptoms are often intermittent and less severe, but may be acute. This condition has been described figuratively as a "left-sided appendicitis".

4. TREATMENT OF PERFORATED DUODENAL ULCER

Acute perforation of any abdominal viscus is a catastrophic event requiring immediate stabilization and preparation for MEDEVAC. Broad-spectrum antibiotic coverage and fluid replacement are required without delay. The aim of treatment of duodenal perforation is to prevent shock and to limit bacterial seeding of the peritoneum which may result in abscess formation. Spread of gastric contents into the peritoneum after perforation occurs rapidly.

The treatment regimen should consist of the following:

a. antibiotics
b. intravenous fluid and electrolyte replacement
c. intestinal rest.

a. Appropriate antibiotic therapy is crucial. Bowel flora will be predominantly gram negative organisms, enterococci (Streptococcus fecaelis), and anaerobes. Specific intravenous antibiotic regimens are as follows:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

b. Intravenous fluids are utilized to maintain fluid and electrolyte requirements in the face of increased need and decreased intake. Dextrose 5% and Lactated Ringer's solution (D5LR) should be started at 150 ml/hour. Additional fluid replacement will be needed for the additional losses caused by fever, diaphoresis, emesis, diarrhea, and nasogastric suctioning. Fluid and electrolyte replacement must keep pace with losses to prevent dehydration and electrolyte imbalance. Monitor the urine output to maintain at least 1000 ml/day (or 1/2 ml per kg body wt per hour) and increase fluid intake as needed. Urine specific gravity and hematocrit are also good indicators of the adequacy of hydration.

c. Intestinal rest is accomplished by making patient NPO (nothing by mouth) and by nasogastric suctioning. Low-intermittent suction (1-3 psi) is most effective. Irrigation of the tube with small volumes of saline may be needed periodically to maintain patency. Remember anything by mouth may be released into the peritoneal cavity by way of the perforation.

NOTE: PAIN MEDICATIONS ARE CONTRAINDICATED since they will mask the symptoms thereby masking possible deterioration in the patient's condition difficult to detect. The head of the bed should be kept in bed in a slightly elevated (semi-Fowler's) position to prevent subdiaphragmatic involvement.

5. USUAL COURSE WITH TREATMENT

Improvement should follow sealing of the perforation and gradual resolution of the chemical peritonitis caused by the gastric contents. Antibiotic therapy and IV fluids may buy the necessary time to allow this to occur. Shock associated with intestinal perforation is, however, usually not well controlled and cases of perforation should be MEDEVAC'ed as soon as possible for surgical treatment.

6. COMPLICATIONS AND THEIR MANAGEMENT

The greatest dangers of duodenal perforation are shock and abscess formation. Spillage of gastric contents into the peritoneal cavity results in a chemical 'burn' leading to extravasation of large amounts of fluids into the peritoneal space. This may be followed by hypovolemic shock. The shock produced by the peritoneal irritation and fluid loss can be profound and occur with great rapidity. Worsening of the hypovolemic shock is a grave sign and requires an all out effort at fluid and electrolyte replacement and maintenance of blood pressure.

Dramatic symptomatic relief is often seen 6-10 hours after perforation as adhesions form and halt the flow of gastric contents into the peritoneum. This event will reduce the possibility of shock, but will not affect abscess formation since bacterial seeding occurs early. Abscesses can form in any area of the peritoneal space. Initial resolution followed by development of fever, tachycardia, and a general deterioration in the status of hydration and electrolyte balance despite vigorous fluid replacement indicates abscess formation and sepsis.

TX6.ASC

Small Bowel Obstruction (SMBOBS)

1. TREATMENT PROTOCOL AT A GLANCE (see detailed discussion of treatment below; this section is for quick reference)

Notify command of need for immediate MEDEVAC

Goals of treatment pending MEDEVAC:

a. prevent dehydration
b. treat for shock
c. decompress the distension
d. antibiotics

a. Start IV D5NS or D5LR at 150 ml/hr, adjust rate as needed to maintain adequate hydration. Monitor urine output as indicator of hydration status.

b. Bolus therapy may be required to maintain blood pressure if patient goes into shock. Monitor vital signs carefully.

c. Keep patient NPO. Place NG tube to low-intermittent suction.

d. Appropriate antibiotic therapy is crucial:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

Note: Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness.

2. DISCUSSION

Intestinal obstruction represents the prevention of passage of intestinal contents by mechanical blockage or failure of bowel motility. The causes include scarring or adhesions from previous surgery or peritonitis, entrapment of a section of bowel in a hernia, fecal impaction, intestinal parasites, carcinoma, twisting, or intussusception of the bowel, inflammatory disease of the bowel and paralytic ileus.

Patients with intestinal obstruction will typically present with abdominal pain, constipation, vomiting, abdominal distension, and failure to pass flatus. The abdominal pain is described as crampy and intermittent, occurring at 3-5 minute intervals. If the obstruction is proximal (close to the stomach), the vomiting may be profuse and the distention not as pronounced. Distal obstruction (closer to the colon or involving the colon itself) will present with less vomiting, but vomitus may contain fecal material. More distal obstruction will be associated with greater distension as intestinal gas and liquid contents accumulate behind the obstruction. Diarrhea may be evident early, followed by failure to pass either flatus or feces.

Examination will reveal abdominal distension, and hyperactive, high-pitched bowel sounds, often described as tinkling or rushes. Increased peristaltic activity may be visible proximal to the obstruction. Dehydration will be of variable severity, dependent on the duration of the complaint and the extent of the vomiting. Rectal examination should be performed to rule out fecal impaction which could be manually removed, detect lesions and test for occult blood (which might indicate the presence of a carcinoma). The white blood cell count will reveal an increased number of neutrophils.

3. DIFFERENTIAL DIAGNOSIS

The diagnosis of intestinal obstruction is usually easily made on the basis of physical examination. Early cases or partial obstructions or other conditions resulting in peritonitis, decreased bowel motility and distention may present diagnostic challenges. Other conditions to consider include:

a. acute appendicitis
b. acute cholecystitis
c. perforated duodenal ulcer
d. pancreatitis

a. Acute appendicitis - see APPEND
b. Acute cholecystitis - see CHOLE
c. Perforated duodenal ulcer - see PERFDU

d. Pancreatitis is associated with duodenal ulcer, cholecystitis, trauma, and alcoholism and will present as a boring epigastric pain radiating to the back. Signs and symptoms of peritoneal irritation are sometimes present. Antacids will not relieve the pain of pancreatitis. Food, particularly fatty foods, will exacerbate the pain. Treatment should include bowel rest, with absolutely no fatty foods or alcohol permitted.

4. **TREATMENT OF BOWEL OBSTRUCTION**

The aims of treatment of intestinal obstruction are to prevent dehydration and shock and to attempt to decompress the distension in preparation for MEDEVAC.

a. Intravenous fluids are utilized to maintain fluid and electrolyte requirements in the face of increased need and decreased intake. There is also a potentially massive fluid shift into the lumen of the obstructed bowel. D5NS or D5LR should be started at 150 ml/hour. Additional fluid replacement will be needed for the additional losses caused by fever, diaphoresis, emesis, diarrhea, and nasogastric suctioning. Fluid and electrolyte replacement must keep pace with losses to prevent dehydration and electrolyte imbalance. Monitor the urine output to maintain at least 1000 ml/day (or 1/2 ml per kg body wt per hour) and increase fluid intake as needed. Urine specific gravity and hematocrit are also good indicators of the adequacy of hydration.

b. Decompression of the stomach and relief of distension can be accomplished by placement of a nasogastric tube and keeping the patient NPO. Low-intermittent suction (1-3 psi) is most effective. Irrigation of the tube with small volumes of saline may be needed periodically to maintain patency.

c. Appropriate antibiotic therapy is crucial. Bowel flora will be predominantly gram negative organisms, enterococci (Streptococcus fecaelis), and anaerobes. Specific intravenous antibiotic regimens are as follows:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

d. Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness. However, it may become necessary to offer the patient some pain relief by judicious use of morphine or Demerol.

5. USUAL COURSE WITH TREATMENT

Intestinal obstruction is an indication for immediate MEDEVAC since there is no reliable way to determine if necrosis of the involved bowel has occurred due to strangulation of the blood supply. The mortality is greatly increased by a delay in definitive care. Obstruction ordinarily does not resolve spontaneously.

6. COMPLICATIONS AND THEIR MANAGEMENT

Strangulation with or without perforation and peritonitis is heralded by a change in the character of the pain which will be continuous and severe. Abdominal sounds will be diminished, and tachycardia, fever, tachypnea and a deterioration of the hydration status despite rigorous efforts will be evident. Morbidity is high in this scenario. Continue supportive care pending MEDEVAC.

Small Bowel Obstruction (SMBOBS)

1. **TREATMENT PROTOCOL AT A GLANCE** (see detailed discussion of treatment below; this section is for quick reference)

Notify command of need for immediate MEDEVAC

Goals of treatment pending MEDEVAC:

a. prevent dehydration
b. treat for shock
c. decompress the distension
d. antibiotics

a. Start IV D5NS or D5LR at 150 ml/hr, adjust rate as needed to maintain adequate hydration. Monitor urine output as indicator of hydration status.

b. Bolus therapy may be required to maintain blood pressure if patient goes into shock. Monitor vital signs carefully.

c. Keep patient NPO. Place NG tube to low-intermittent suction.

d. Appropriate antibiotic therapy is crucial:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose - 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose - 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

Note: Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness.

2. DISCUSSION

Intestinal obstruction represents the prevention of passage of intestinal contents by mechanical blockage or failure of bowel motility. The causes include scarring or adhesions from previous surgery or peritonitis, entrapment of a section of bowel in a hernia, fecal impaction, intestinal parasites, carcinoma, twisting, or intussusception of the bowel, inflammatory disease of the bowel and paralytic ileus.

Patients with intestinal obstruction will typically present with abdominal pain, constipation, vomiting, abdominal distension, and failure to pass flatus. The abdominal pain is described as crampy and intermittent, occurring at 3-5 minute intervals. If the obstruction is proximal (close to the stomach), the vomiting may be profuse and the distention not as pronounced. Distal obstruction (closer to the colon or involving the colon itself) will present with less vomiting, but vomitus may contain fecal material. More distal obstruction will be associated with greater distension as intestinal gas and liquid contents accumulate behind the obstruction. Diarrhea may be evident early, followed by failure to pass either flatus or feces.

Examination will reveal abdominal distension, and hyperactive, high-pitched bowel sounds, often described as tinkling or rushes. Increased peristaltic activity may be visible proximal to the obstruction. Dehydration will be of variable severity, dependent on the duration of the complaint and the extent of the vomiting. Rectal examination should be performed to rule out fecal impaction which could be manually removed, detect lesions and test for occult blood (which might indicate the presence of a carcinoma). The white blood cell count will reveal an increased number of neutrophils.

3. DIFFERENTIAL DIAGNOSIS

The diagnosis of intestinal obstruction is usually easily made on the basis of physical examination. Early cases or partial obstructions or other conditions resulting in peritonitis, decreased bowel motility and distention may present diagnostic challenges. Other conditions to consider include:

a. acute appendicitis
b. acute cholecystitis
c. perforated duodenal ulcer
d. pancreatitis

a. Acute appendicitis - see APPEND
b. Acute cholecystitis - see CHOLE
c. Perforated duodenal ulcer - see PERFDU

d. Pancreatitis is associated with duodenal ulcer, cholecystitis, trauma, and alcoholism and will present as a boring epigastric pain radiating to the back. Signs and symptoms of peritoneal irritation are sometimes present. Antacids will not relieve the pain of pancreatitis. Food, particularly fatty foods, will exacerbate the pain. Treatment should include bowel rest, with absolutely no fatty foods or alcohol permitted.

4. TREATMENT OF BOWEL OBSTRUCTION

The aims of treatment of intestinal obstruction are to prevent dehydration and shock and to attempt to decompress the distension in preparation for MEDEVAC.

a. Intravenous fluids are utilized to maintain fluid and electrolyte requirements in the face of increased need and decreased intake. There is also a potentially massive fluid shift into the lumen of the obstructed bowel. D5NS or D5LR should be started at 150 ml/hour. Additional fluid replacement will be needed for the additional losses caused by fever, diaphoresis, emesis, diarrhea, and nasogastric suctioning. Fluid and electrolyte replacement must keep pace with losses to prevent dehydration and electrolyte imbalance. Monitor the urine output to maintain at least 1000 ml/day (or 1/2 ml per kg body wt per hour) and increase fluid intake as needed. Urine specific gravity and hematocrit are also good indicators of the adequacy of hydration.

b. Decompression of the stomach and relief of distension can be accomplished by placement of a nasogastric tube and keeping the patient NPO. Low-intermittent suction (1-3 psi) is most effective. Irrigation of the tube with small volumes of saline may be needed periodically to maintain patency.

c. Appropriate antibiotic therapy is crucial. Bowel flora will be predominantly gram negative organisms, enterococci (Streptococcus fecaelis), and anaerobes. Specific intravenous antibiotic regimens are as follows:

1) Cefoxitin 2 gm IV every 6 hours; and

Gentamicin 80 mg IV every 8 hours; or

2) For patient's with a history of Penicillin or Cephalosporin allergy only:

Gentamicin 80 mg IV q8h; and

Metronidazole (Flagyl):

Loading dose 15 mg/kg body weight IV infused over one hour (approximately 1 gm for average sized male); and

Maintenance dose 7.5 mg/kg IV infused over one hour every 6 hours (approximately 500 mg)

Slow IV infusions are best accomplished by mixing in small bags of IV solution and running "piggy backs".

d. Analgesics are relatively contraindicated since they will mask the symptoms and make it difficult to follow the progress of the illness. However, it may become necessary to offer the patient some pain relief by judicious use of morphine or Demerol.

5. USUAL COURSE WITH TREATMENT

Intestinal obstruction is an indication for immediate MEDEVAC since there is no reliable way to determine if necrosis of the involved bowel has occurred due to strangulation of the blood supply. The mortality is greatly increased by a delay in definitive care. Obstruction ordinarily does not resolve spontaneously.

6. COMPLICATIONS AND THEIR MANAGEMENT

Strangulation with or without perforation and peritonitis is heralded by a change in the character of the pain which will be continuous and severe. Abdominal sounds will be diminished, and tachycardia, fever, tachypnea and a deterioration of the hydration status despite rigorous efforts will be evident. Morbidity is high in this scenario. Continue supportive care pending MEDEVAC.

TX7.ASC

Pelvic Inflammatory Disease (PID)

1. DISCUSSION

Pelvic inflammatory disease (PID) is a clinical syndrome characterized by the spread of micro-organisms from the vagina and cervix to the endometrium, fallopian tubes and related pelvic structures. Etiology may be bacterial or nonbacterial. Gonococcus and Chlamydia are the most common offending agents. Polymicrobial infections are common. Predisposing factors include: multiple sexual partners, IUD, and introduction of foreign bodies into the vagina.

History and examination reveal lower quadrant abdominal pain and tenderness, which may be bilateral or unilateral, though infection is usually present bilaterally. On pelvic exam, cervical motion tenderness, purulent cervical discharge, adnexal tenderness, plus one or more of the following is diagnostic:

a. positive gram stain for gonococcus.
b. temperature greater than 38°C (100.4°F).
c. WBC > 10,000/cu mm.
d. pelvic abscess on examination.

Other findings on physical examination may include liver tenderness, abdominal rigidity and/or guarding, abdominal distension, decreased bowel sounds, adnexal mass, vaginal bleeding, vaginal discharge, dysuria, and/or low back pain.

As a general rule, PID associated with gonococcal infections presents with more severe symptoms than those associated with nongonococcal infections. Many cases are asymptomatic.

2. DIFFERENTIAL DIAGNOSIS:

a. incomplete abortion and Septic abortion
b. pyelonephritis
c. appendicitis
d. diverticulitis
e. ruptured ovarian cyst
f. cholecystitis
g. pelvic thrombophlebitis
h. ectopic pregnancy
i. intestinal obstruction
j. endometriosis

a. Incomplete and septic abortion - see INCOMPLETE ABORTION

b. Pyelonephritis - see URINARY TRACT INFECTION

c. Appendicitis - see APPENDICITIS

d. Diverticulitis usually occurs in older individuals. It may occur on the right side, but most commonly is on the left. See APPENDICITIS.

e. Ruptured ovarian cyst - see OVARIAN CYST

f. Cholecystitis - see CHOLECYSTITIS

g. Pelvic thrombophlebitis typically presents with abdominal pain and recurrent spiking fever developing 2-21 days post-partum, or following abortion or pelvic surgery. The pain may be minimal and poorly localized. Tachycardia and tachypnea may be associated findings. The white count will range from normal to as high as 25,000/cu mm.

h. Ectopic pregnancy - see ECTOPIC PREGNANCY

i. Intestinal obstruction - see SMALL BOWEL OBSTRUCTION

j. Endometriosis has a nonspecific presentation including progressive dysmenorrhea, pelvic pain, and infertility. Premenstrual spotting, menometrorrhagia, rectal bleeding, and hematuria are common. On pelvic examination tender nodularities may be present. Definitive diagnosis is made with direct visualization by laparotomy or laparoscopy.

3. TREATMENT OF PELVIC INFLAMMATORY DISEASE

Treatment of PID requires broad spectrum antibiotic coverage, as well as supportive therapy.

a. Antibiotics: In the absence of peritoneal signs (stage I PID), therapy may be initiated with oral antibiotics. Suggested regimens are outlined below:

1) Ceftriaxone (Rocephin) 250 mg IM single dose; and Doxycycline (Vibramycin) 100 mg PO bid for 14 days.

2) Spectinomycin (Trobicin) 2 gm IM may be substituted for ceftriaxone if the patient is penicillin allergic.

The patient must be followed closely. If there is no improvement in 48 hours, or the initial evaluation reveals evidence of peritonitis (stage II PID), more aggressive therapy is required:

3) Ceftriaxone 500 mg IM or IV q 6 hours; and Doxycycline 100 mg IV q 12 hours (if available, not currently on AMAL) or 100 mg PO bid.

(Cefoxitin 2 gm IV q 6 hours may be substituted for ceftriaxone.)

4) Metronidazole (Flagyl) 1 gm IV q 12 hours may be substituted

for ceftriaxone if the patient is penicillin allergic.

Continue IV antibiotics for a minimum of 2 days after improvement is noted, but for at least 4 days. Switch to the appropriate PO regimen and continue the doxycycline for a total of 14 days.

b. Supportive therapy:

1) Bed rest with elevation of the head.

2) Abstinence from sexual intercourse throughout the course of the treatment.

3) Encourage po fluid intake.

4) Analgesics acetaminophen and codeine (Tylenol 3) 1-2 tablets PO q4 hours PRN.

MEDEVAC and inpatient therapy must be sought if any of the following conditions exist in conjunction with PID:

a. nulliparity
b. presence of tubo-ovarian mass or abscess
c. pregnancy
d. uncertain diagnosis
e. gastrointestinal symptoms
f. peritonitis in upper quadrants
g. presence of an intrauterine device (IUD)
h. recent history of abdominal or pelvic surgery
i. inadequate response to outpatient therapy.

4. USUAL COURSE WITH TREATMENT:

Recovery can be anticipated in the vast majority of patients given appropriate treatment. There is a high degree of recurrence of this disease with future sexually transmitted infections. Scarring of the fallopian tubes secondary to the infection increases the risk of infertility and ectopic pregnancy. Twenty percent may develop chronic pelvic pain without evidence of infection.

5. COMPLICATIONS AND THEIR MANAGEMENT:

a. Tubo-ovarian abscess formation (stage III PID) and intra-abdominal rupture (stage IV PID) are the most serious complications of PID. The mortality of a ruptured abscess ranges from 10-50%. The diagnosis must be suspected if the patient presents with signs of generalized peritonitis or septic shock, especially if accompanied by a high, spiking fever, chills, tachycardia, and hypotension. This condition is a medical emergency requiring immediate MEDEVAC. Prompt intervention with fluid resuscitation to restore blood pressure may be required.

Therapy will include IV antibiotics as outlined above.

Add: Gentamicin 2 mg/kg body weight IV followed by 1.5 mg/kg IV q 8 hours.

b. Fitz-Hugh-Curtis Syndrome is a perihepatitis resulting from the spread of infection into the peritoneal cavity and to the liver capsule. The patient will present with sudden onset of severe sharp pain in the right upper quadrant and liver edge tenderness. The pain may be aggravated by deep breathing, coughing, or movement, and may be referred to the shoulder. Fever, chills, sweats, nausea and headache are common. Presentation may mimic cholecystitis. This condition is a medical emergency which requires immediate MEDEVAC.

c. Infertility occurs in approximately 20% of patients with PID secondary to tubal occlusion. The extent of tubal occlusion increases with repeated bouts of PID and the severity of the infections. Counsel all female STD patients on these risks.

d. Ectopic pregnancy may result from damage to the fallopian tubes. PID is the most common cause of ectopic pregnancy. A patient recovering from PID should be referred for counseling and follow-up with OB-GYN. See ECTOPIC PREGNANCY.

TX8.ASC

Urinary Tract Infection (UTI)

1. DISCUSSION

Urinary tract infections are divided into lower tract infections involving the bladder (cystitis) and more serious upper tract infections involving the kidneys (pyelonephritis). The offending organism is most commonly E. coli from the patient's own natural flora, though many others are seen. Ninety-five percent of the cases arise as bacteria ascends the urethra, while 5% arise via other routes including hematogenous spread and secondary to bladder catheterization. UTI's are more common in the female due to the shorter, wider urethra. Predisposing factors include decreased resistance to infection, urinary stasis from obstruction or abnormal bladder function, pregnancy, kidney stones, neurogenic bladder, and diabetes.

Signs and symptoms of cystitis include suprapubic abdominal pain, dysuria, and urinary frequency. Fever, nausea, and vomiting are inconstant findings. On microscopic examination the urine will exhibit >8 white blood cells (WBC)/high power field (hpf) and one bacteria per oil emersion field (equivalent to 100,000 bacteria/ml). Routine urinalysis may show an increase in pH, positive leukocytes, and positive nitrite (if organism is urea splitting).

Pyelonephritis may result from the reflux of urine up the ureters. Fever (>101^{o}F), rigors, nausea, vomiting, flank pain, and tenderness are indicative of kidney involvement. Patients with pyelonephritis, if not adequately treated, may become septic. Labs may demonstrate an increased WBC count on CBC, hematuria, proteinuria, ketonuria, and white blood cell casts on microscopic urine examination.

2. DIFFERENTIAL DIAGNOSIS

The differential diagnosis of cystitis includes:

a. urethritis
b. vaginitis
c. genital herpes

a. Urethritis is more common in males and presents with similar symptoms and lab findings. Five to eight WBC/hpf on microscopic urine examination is sufficient for diagnosis. Treatment is the same as for cystitis.

b. Vaginitis may have a similar presentation with a complaint of dysuria, additionally the patient may have evidence of vaginal discharge and vaginal itching.

c. Genital herpes presents with characteristic lesions in most primary cases. However, recurrent episodes may present with symptoms similar to cystitis. Occasionally, dysuria may precede the appearance of the lesions in both primary and recurrent cases making diagnosis difficult.

The differential diagnosis of pyelonephritis includes:

a. renal contusion
b. perinephric abscess
c. glomerulonephritis
d. cholecystitis
e. appendicitis
f. kidney stones
g. renal neoplasm

a. Renal contusion secondary to trauma may produce flank pain and hematuria. A history of trauma makes the diagnosis clear.

b. Perinephric abscess see Complications of UTI (below)

c. Glomerulonephritis is inflammation and damage to the microscopic structure of the kidney. There are a number of different causes and manifestations. WBC's, RBC's and red and white cell casts may be seen. The urine is usually sterile. Proteinuria may be pronounced.

d. Cholecystitis see CHOLECYSTITIS

e. Appendicitis see APPENDICITIS

f. Kidney stones see RENAL COLIC

g. Renal neoplasms (tumor) may present with hematuria, flank pain, and fever. White cells and bacteria will not be seen in the urine ordinarily. A neoplasm may cause urinary stasis, producing an infection, however.

3. TREATMENT OF URINARY TRACT INFECTION

a. The treatment of cystitis involves the administration of fluids and antibiotics.

1) Fluids: The patient should be advised to drink several (8-10) glasses of water per day to help flush out the urinary tract as well as to prevent dehydration.

2) Antibiotics: Single dose regimens for uncomplicated cystitis include:

a) Amoxicillin 3 grams PO, or

b) Trimethoprim-Sulfamethoxazole (Bactrim,Septra) one double strength (DS) tablet PO

3) Ten day regimens should be employed if the patient:

is pregnant.

is unresponsive to initial single dose therapy.
has evidence of upper urinary tract infection.
has evidence of urologic abnormalities.
is a child.
is male.
is unreliable or non-compliant.
is greater than six days past the onset of symptoms.

Ten day antibiotic regimens include:

a) Trimethoprim/Sulfamethoxazole (Septra, Bactrim) one DS tablet PO BID (not for use during pregnancy or while breast feeding).

b) Ampicillin 250 mg PO QID (use this drug in pregnant and breast feeding women).

c) Sulfisoxazole (Gantrisin) 1.5 gm PO QID (not for use during pregnancy or while breast feeding).

d) Nitrofurantoin (Furdantin, Macrodantin) 100 mg PO QID x 10 days (not for use during pregnancy or while breast feeding).

4) Pyridium is a drug that is excreted in the urine and acts as a topical anesthetic on the bladder and urethral mucosa.

Phenazopyridine (Pyridium) 200 mg PO TID x 2-3 days. Do not use for more than three days. Longer courses may mask the symptoms of continued infection.

Warn the patient that this drug may cause their urine to be red or orange.

b. The treatment of pyelonephritis involves antibiotics and fluid administration. Patients with mild illness who are not toxic (septic) and able to maintain hydration orally can be managed locally with close follow-up. Patients who are toxic or who cannot maintain good hydration with oral fluids or who fail to improve should be evacuated for inpatient IV hydration, and antibiotics. If the patient is pregnant they should be MEDEVAC'ed for inpatient care.

The treatment for uncomplicated pyelonephritis is:

1) Fluids: The patient should be instructed to drink 3-4 L of water or other clear fluids per day to increase urine output and maintain hydration.

2) Antibiotics

a) Ampicillin 1 gm PO QID x 10 days. (use in pregnant and breast feeding women).

b) Trimethoprim/Sulfamethoxazole (Bactrim, Septra) one DS tablet PO BID x 10 days. (Not for use during pregnancy or while breast feeding).

c) Sulfisoxazole (Gantrisin) 1 gram PO QID x 10 days. (Not for use during pregnancy or while breast feeding).

3) Pyridium may be used as outlined above.

4. USUAL COURSE WITH TREATMENT:

Patients with uncomplicated cystitis usually begin to feel relief within 48 hours, sooner if Pyridium is used. The patient must be instructed to return if the symptoms continue. The importance of drinking large amounts of fluids cannot be over stressed. The patient must be instructed to complete the course of antibiotics and continue the medications even after the symptoms improve. A follow-up urinalysis should be done after completion of the antibiotic course.

Patients with pyelonephritis may notice relief of their urinary symptoms (dysuria, frequency etc.) before their constitutional symptoms and flank pain. Improvement should be noted within 48 hours. Worsening symptoms should be treated aggressively. Permanent kidney damage can result.

5. COMPLICATIONS AND THEIR MANAGEMENT:

a. Complications of cystitis include:

1) Pyelonephritis see PYELONEPHRITIS above.

2) Septic shock - see BACTEREMIA under complications of pyelonephritis.

3) Kidney stones see RENAL COLIC

b. Complications of pyelonephritis include:

1) Bacteremia is the spread of bacteria into the blood stream. This may be asymptomatic, or the patient may demonstrate signs of septicemia with high, spiking fevers, shaking chills (rigors), nausea, vomiting and prostration. Some cases may progress to septic shock, with tachycardia, tachypnea, and hypotension. These patients are gravely ill and require aggressive care pending MEDEVAC.

Treatment for bacteremia/septicemia:

a) notify command of the need for MEDEVAC

b) start IV D5NS or D5LR

c) Ampicillin 1 gram IV q 4 hours; and Gentamicin 80 mg IV q 8 hours

d) Remain prepared for large volume fluid resuscitation should patient's blood pressure begin to fall.

2) Perinephric abscess primarily occurs in association with recurrent infections, immunocompromise, diabetes, or urinary tract obstruction. The patient may present up to 2-3 weeks following an episode of pyelonephritis with persistent fever, vague constitutional symptoms, and flank or chest pain. Such abscesses require MEDEVAC and inpatient treatment and may require surgical drainage. Treat with the above IV antibiotic regimen pending MEDEVAC.

3) Acute renal failure (ARF) is a rare complication of pyelonephritis most often seen in severely dehydrated patients. It may be transient or permanent. It may present as a marked decrease in urine output despite evidence of adequate hydration. It requires immediate MEDEVAC for intervention. Use of IV fluids in the interim must be closely monitored. It is possible to fluid overload the patient and produce respiratory compromise.

TX9.ASC

Ovarian Cyst

1. DISCUSSION

Benign ovarian cysts are fluid filled masses located on the ovary. They are most common in women in the child-bearing years and are under hormonal influence. They may be filled with clear fluid or blood, vary greatly in size, and may be unilateral or bilateral. There are different types of ovarian cysts with differing potentials for later malignancy. They are usually asymptomatic unless torsion (twisting) or rupture occurs. Large cysts may cause aching pelvic pain, dyspareunia, and, occasionally, abnormal uterine bleeding. On pelvic exam, the ovary may be slightly enlarged and tender to palpation.

Some cysts have thin walls and are prone to rupture during the pelvic examination. This rupture occasionally causes transient pain and very rarely significant bleeding. Other types of cysts are less prone to rupture, but may cause significant bleeding if they do. The rupture of a cyst can cause peritoneal irritation from fluid or blood. The pain associated with this irritation tends to be constant and exacerbated by motion.

With torsion of a cyst, the pain may be mild and intermittent if the torsion is incomplete, or severe and constant if infarction occurs. An infarcted, gangrenous cyst may cause peritonitis.

There are a number of malignancies that are cystic. All patients suspected of having ovarian cysts should, therefore, be referred to a Gynecologist at the earliest opportunity.

2. DIFFERENTIAL DIAGNOSIS

a. ectopic pregnancy
b. acute appendicitis
c. pelvic inflammatory disease
d. diverticulitis

a. Ectopic Pregnancy see ECTOPIC PREGNANCY. If an adnexal mass is palpated and there is a possibility that the patient is pregnant (eg: history of unprotected intercourse) MEDEVAC. Err on the side of over-diagnosing ectopic pregnancy. Failure to do so could be catastrophic.

b. Acute appendicitis see APPENDICITIS.

c. Pelvic inflammatory disease see PID.

d. Diverticulitis most commonly involves the descending (left) colon. Symptoms are often intermittent and moderate, but may be acute. This condition has been described figuratively as a "left-sided appendicitis", but may present with a pelvic or lower abdominal mass and peritoneal signs, thus mimicking a gynecological problem.

3. TREATMENT OF OVARIAN CYST

Treatment of ovarian cysts is symptomatic and supportive. For intermittent pain due to torsion, Motrin 400 mg PO qid should provide relief. The pain associated with the majority of ruptured cysts will resolve spontaneously in several hours. If the pain does not subside, consider one of the complications of ovarian cyst and carefully monitor the patient accordingly.

4. USUAL COURSE WITH TREATMENT

The symptoms of ovarian cysts will usually resolve spontaneously, but it may take up to 8 weeks, as the cyst reabsorbs. If the cyst ruptures, symptoms should resolve in a matter of hours if the bleeding is not severe.

5. COMPLICATIONS AND THEIR MANAGEMENT

a. Rupture may result in significant hemorrhage in rare cases. These cases will become obvious as symptoms persist and become more severe. Immediate MEDEVAC is required. Two IV lines (18g or larger) should be placed with NS or LR at KVO. The patient should be closely monitored for signs of shock (tachycardia, tachypnea, hypotension, etc.) and volume resuscitation implemented accordingly. If pain is severe, narcotic analgesics may be used judiciously, but care should be taken to avoid hypotension.

b. Malignancy must be ruled out. Refer all cases to a Gynecologist as soon as possible.

c. Pregnancy: Presence of any of the symptoms of an ovarian cyst in conjunction with pregnancy requires immediate MEDEVAC to rule out ectopic pregnancy.

TX10.ASC

Ectopic Pregnancy

1. DISCUSSION

Ectopic pregnancy is the implantation of a fertilized ovum outside the fundus of the uterus. It is a medical emergency which requires immediate MEDEVAC and surgical intervention. It is the leading cause of maternal death in the first trimester of pregnancy, with delay in diagnosis contributing to the majority of these deaths. The fertilized ovum may implant in the fallopian tube, ovary, abdominal viscera, or the cervix. Regardless of the site of implantation, the fetus grows until it ruptures the supporting structure. This rupture results in massive bleeding which may ultimately exsanguinate the patient. Signs and symptoms may occur prior to or following the rupture. There is often a history of PID or other gynecologic infections.

Typically, menses are 1 to 2 weeks late, with spotting and intermittent cramping pelvic pain. Rupture may be preceded by increased vaginal bleeding. Abdominal pain will increase and may initially present as pressure in the pelvis; it is variable in character and intensity. Hemorrhage may be gradual or catastrophic, resulting in rapid onset of shock.

On examination, the uterus is enlarged, but smaller than anticipated for dates. There will be cervical motion tenderness and an adnexal mass may be present. Absence of the latter does not rule out the diagnosis. Intraperitoneal bleeding may produce generalized abdominal pain and tenderness, rebound, and other signs of peritonitis. The peritonitis may, however, be lateralized to the side of the rupture. Vital sign derangement will reflect the degree of hemorrhage.

2. DIFFERENTIAL DIAGNOSIS

Other illnesses and conditions which may produce symptoms mimicking ectopic pregnancy are listed below.

Err on the side of over-diagnosing ectopic pregnancy. Failure to do so could be catastrophic.

a. pelvic inflammatory disease
b. acute appendicitis
c. ovarian cyst
d. threatened/incomplete abortion
e. endometriosis
f. diverticulitis

a. Pelvic inflammatory disease see PID

b. Acute appendicitis see APPENDICITIS

c. Ovarian cyst see OVARIAN CYST

d. Threatened/Incomplete abortion see INCOMPLETE ABORTION

e. Endometriosis has a nonspecific presentation, including progressive dysmenorrhea, pelvic pain, and infertility. Premenstrual spotting, meno-metrorrhagia, rectal bleeding, and hematuria are common. On pelvic examination tender nodularities may be present. Definitive diagnosis is made with direct visualization by laparotomy or laparoscopy.

f. Diverticulitis most commonly involves the descending (left) colon. Symptoms are often intermittent and moderate, but may be acute. This condition has been described figuratively as a "left-sided appendicitis", but may present with a pelvic or lower abdominal mass and peritoneal signs, thus mimicking a gynecological problem.

3. TREATMENT OF ECTOPIC PREGNANCY

Regardless of whether the diagnosis of ectopic pregnancy is made before or after rupture, arrange for immediate MEDEVAC.

Definitive, life-saving care is surgical.

Treatment pending MEDEVAC is supportive and adjusted according to the severity of the hemorrhage.

The treatment regimen for the patient with no signs or symptoms of rupture or shock include:

a. Establish intravenous access with two large-bore (18 ga or larger) catheter with NS or LR at 100-125 ml/hour.

b. Carefully monitor for signs of rupture and/or shock.

c. Avoid the use of analgesics. Masking symptoms of a ruptured ectopic pregnancy could be catastrophic.

The treatment regimen for the patient with any sign of rupture and/or shock include:

a. Immediate placement of two large-bore (18 ga or larger) catheters and fluid administration (NS or LR) to maintain blood pressure. Rapid infusion of several liters may be necessary. If facilities for transfusion are available, consider transfusing whole blood as needed.

b. Oxygen 2 liters/minute via nasal canula.

c. Avoid the use of analgesics as they may exacerbate hypotension.

d. Use of MAST suit to assist in the treatment of profound shock may be indicated.

4. USUAL COURSE WITH TREATMENT

The overall mortality rate of ectopic pregnancy is less than 1% when definitive, surgical treatment is available. Delay in obtaining treatment, either due to misdiagnosis or to delay in the patient's seeking care, contributes significantly to the morbidity.

Failure to maintain adequate volume infusion and to support the patient's blood pressure may result in death.

5. COMPLICATIONS AND THEIR MANAGEMENT

The most serious complication of an ectopic pregnancy is hemorrhaging and shock secondary to rupture. Once the diagnosis of ectopic pregnancy is made, or even in the case where it cannot be ruled out, MEDEVAC must be arranged at once. Should rupture occur without immediate access to surgical care, the prognosis is poor.

TX11.ASC

Spontaneous Abortion

1. DISCUSSION

Abortion is defined as the delivery of the products of conception (POC) prior to the twentieth week of gestation. In spontaneous abortion this occurs without outside inducement or influence. Spontaneous abortion may be classified as threatened, inevitable, incomplete, or complete.

Threatened abortion: any bleeding or cramping during first 20 weeks of gestation. Cervix is closed. No passage of POC.

Inevitable abortion: intolerable pain, or bleeding that threatens maternal well-being. Cervix may be partially open. No passage of POC.

Incomplete abortion: same as inevitable abortion except cervix will be partially open and there is some passage of POC. Retained POC may be visible in cervical os on speculum exam.

Complete abortion: history of pain and bleeding. All POC are passed. Cervix is closed. Uterus has shrunk back down to its normal size. A loss of the subjective signs of pregnancy may be reported. Bleeding at time of presentation may be limited to spotting.

Threatened abortion occurs in approximately 20 to 30% of pregnancies. Ten to fifteen percent of pregnancies actually end in spontaneous abortion. Most of these occur between the 8th and 14th weeks of pregnancy.

Frequently, the pregnancy has not been diagnosed prior to presentation with spontaneous abortion. Therefore, a discussion of the presentation must include the signs and symptoms of pregnancy. These include a history of unprotected intercourse or inadequate contraception, delay of menses greater than one week in an ordinarily regular cycle, breast engorgement or tenderness, nausea and vomiting, fatigue, abdominal enlargement and uterine enlargement on pelvic exam. Signs and symptoms will change with increasing gestational age. Several other signs may be noted by the experienced examiner on pelvic exam. A urine HCG, if available, is highly supportive of the diagnosis of pregnancy, but may be negative with spontaneous abortion.

Missed abortion is a related complication of early pregnancy. This is defined as a pregnancy which has been retained for 2 months or more following the death of the fetus. It occurs more commonly later in the first trimester or early second trimester. Missed abortion is manifested by the loss of subjective signs of pregnancy and a decrease in uterine size. The cervical os will be closed with no evidence of fresh bleeding. A brown discharge may be present. There is an absence of pain or tenderness. The pregnancy test will be negative.

2. DIFFERENTIAL DIAGNOSIS

The differential diagnosis of a spontaneous abortion includes:

a. ectopic pregnancy
b. ovarian cyst

a. Ectopic pregnancy see ECTOPIC PREGNANCY. If an adnexal mass is palpated and there is a possibility that the patient is pregnant (eg: history of unprotected intercourse) MEDEVAC. Err on the side of over-diagnosing ectopic pregnancy. Failure to do so could be catastrophic.

b. Ovarian cyst see OVARIAN CYST

3. TREATMENT OF SPONTANEOUS ABORTION

a. Threatened Abortion:

1) Place the patient at strict bed rest.

2) Notify command of need for MEDEVAC.

3) No intercourse, tampons, or other foreign objects in vagina.

4) Observe and quantify bleeding in units of 'pads soaked per hour', record in note.

5) Start a large bore IV (18g or larger) with NS or LR at KVO.

6) Monitor vital signs carefully to insure hemodynamic stability.

7) Avoid use of sedatives and analgesics, if possible, until viability of fetus is determined; however, diazepam (Valium) 2.5 mg IM q 3-4 hours for sedation or meperidine (Demerol) 50-75 mg IM q 3-4 hours for pain may be used if absolutely necessary.

b. Inevitable and Incomplete Abortion:

1) Institute treatment plan for threatened abortion including plans for MEDEVAC.

2) Start second large bore IV (18g or larger). Monitor vital signs for signs of hemodynamic compromise and infuse fluids as needed.

3) To control bleeding, if greater than 1-2 pads soaked per hour, use:

Oxytocin 10 units (1 ml) in 500 ml of D5NS to run at 10-40 drops per minute.

4) Preserve all passed products of conception and transport with patient for later examination. (Use formalin if available; if not a 50% solution of ethanol or isopropyl alcohol in saline may be used).

5) Diazepam (Valium) 2.5 mg IM q 3-4 hours PRN and meperidine (Demerol) 50-75 mg IM q 3-4 hours PRN may be used for sedation and analgesia respectively.

c. Complete abortion: As it is difficult to determine whether the abortion is complete without a gynecological examination, the protocol for inevitable or incomplete abortion should be followed.

4. USUAL COURSE WITH TREATMENT

With prompt medical treatment the prognosis for maternal survival is excellent. The patient may experience intermittent bleeding and cramping during the following week.

With the above measures approximately 50% of threatened abortions will be carried to term. In the majority of spontaneous abortions the fetus is either absent or has abnormalities incompatible with life. A single spontaneous abortion is usually ascribed to chance. You must use careful judgment, but some patients may find these facts reassuring. Since all of these patients must ultimately be referred for gynecological care, a discussion of further work-up is unnecessary here.

5. COMPLICATIONS AND THEIR MANAGEMENT

a. Septic abortion develops when the contents of the uterus become infected before, during, or after abortion. Prior to legalized abortion, this was most often associated with induced abortions by untrained persons.

In addition to the signs and symptoms of spontaneous abortion, the patient will present with fever (> 102°F), weakness, purulent, foul-smelling vaginal discharge, and cervical pain on pelvic exam. The patient may also be tachycardic and hypotensive due to the infection or blood loss. On CBC, the white count will be elevated. This is a life-threatening situation which require immediate MEDEVAC.

Add the following antibiotics to the inevitable/incomplete abortion regimen for the first 24 hours:

Penicillin 3 million u IV q 4 hours and

Gentamicin 1.5 mg/kg IV q 8 hours

b. Coagulation disorders including disseminated intravascular coagulopathy (DIC) may rarely develop, particularly with missed abortion. The patient may develop massive bleeding due to a consumption of their own clotting factors. This disorder requires very complex management including transfusions of fresh frozen plasma and platelets. Immediate MEDEVAC is warranted.

Appendix F
Miscellaneous Source Listings

ABDSX.DAT

```
MALE
FEMALE
AGE 0-9
AGE 10-19
AGE 20-29
AGE 30-39
AGE 40-49
AGE 50-59
AGE 60-69
AGE >69
PAIN ONSET RUQ
PAIN ONSET LUQ
PAIN ONSET RLQ
PAIN ONSET LLQ
PAIN ONSET UPPER 1/2
PAIN ONSET LOW HALF
PAIN ONSET RT HALF
PAIN ONSET LEFT HALF
PAIN ONSET CENTRAL
PAIN ONSET GENERAL
PAIN ONSET RT FLANK
PAIN ONSET LT FLANK
NO PAIN AT ONSET
PAIN NOW RUQ
PAIN NOW LUQ
PAIN NOW RLQ
PAIN NOW LLQ
PAIN NOW UPPER HALF
PAIN NOW LOWER HALF
PAIN NOW RIGHT HALF
PAIN NOW LEFT HALF
PAIN NOW CENTRAL
PAIN NOW GENERAL
PAIN NOW RT FLANK
PAIN NOW LT FLANK
NO PAIN NOW
PAIN INTERMITTENT
PAIN STEADY
PAIN COLICKY
PAIN IS MODERATE
```

ABDSX.DAT (cont'd)

```
PAIN IS SEVERE
MOVEMENT AGGRAVATES
COUGHING AGGRAVATES
BREATHING AGGRAVATES
FOOD AGGRAVATES
AGGRAVATED BY OTHER
NOTHING AGGRAVATES
PROGRESS - BETTER
PROGRESS - SAME
PROGRESS - WORSE
DURATION <12 HRS
DURATION 12-24 H
DURATION 24-48 H
DURATION 48+HRS
LYING STILL RELIEVES
VOMITING RELIEVES
ANTACIDS RELIEVE
FOOD RELIEVES
RELIEVED BY OTHER
NOTHING RELIEVES
NAUSEA PRESENT
NO NAUSEA
VOMITING PRESENT
NO VOMITING
BOWELS NORMAL
CONSTIPATION PRESENT
DIARRHEA PRESENT
BLOOD IN STOOLS
MUCUS IN STOOLS
APPETITE DECREASED
APPETITE NORMAL
JAUNDICE PRESENT
NO JAUNDICE
URINATION NORMAL
URINATION - FREQUENT
URINATION - PAINFUL
URINATION - DARK
BLOOD IN URINE
PREVIOUS INDIGESTION
NO PREV. INDIGESTION
PREV. SIMILAR PAIN
NO PREV. SIM. PAIN
PREV. ABD. SURGERY
NO PREV. ABD. SURG.
PREVIOUS ILLNESS(es)
NO PREVIOUS ILLNESS
TAKING MEDS
NOT TAKING MEDS
TEMP <98.6
```

ABDSX.DAT (cont'd)

```
TEMP 98.6 - 100.2
TEMP 100.3 - 102
TEMP >102
PULSE <80
PULSE 80-99
PULSE >99
SYST. BP <90
SYST. BP 90-129
SYST. BP >129
DIAST. BP <70
DIAST. BP 70-89
DIAST. BP >89
MOOD NORMAL
MOOD DISTRESSED
MOOD ANXIOUS
COLOR NORMAL
COLOR PALE
COLOR FLUSHED
COLOR JAUNDICED
COLOR CYANOTIC
WBC  < 8000
WBC 8,100-10,000
WBC 10,100-12,000
WBC 12,100-15,000
WBC >15,000
ABD INSPECT. NORMAL
VISIBLE PERISTALISIS
DECREASED ABD MOVE.
ABD SCARS PRESENT
NO ABDOMINAL SCARS
GUARDING PRESENT
NO GUARDING
RIGIDITY PRESENT
NO RIGIDITY
BOWEL SOUNDS NORMAL
BOWEL SOUNDS ABSENT
HYPER. BOWEL SOUNDS
ABDOMEN DISTENDED
NO ABD DISTENTION
MASS(es) PRESENT
NO ABD MASSES
TENDERNESS RUQ
TENDERNESS LUQ
TENDERNESS RLQ
TENDERNESS LLQ
TENDER UPPER HALF
TENDER LOWER HALF
TENDER RIGHT HALF
TENDER LEFT HALF
```

ABDSX.DAT (cont'd)

```
CENTRAL TENDERNESS
GENERAL TENDERNESS
TENDERNESS RT FLANK
TENDERNESS LT FLANK
NO TENDERNESS
MURPHY'S POSITIVE
MURPHY'S NEGATIVE
REBOUND PRESENT
NO REBOUND
RECTAL - NORMAL
RECTAL MASS
LT RECTAL TENDERNESS
RT RECTAL TENDERNESS
GEN. RECTAL TENDERNESS
PELVIC - NORMAL
RT PELVIC TENDERNESS
LT PELVIC TENDERNESS
CERVICAL TENDERNESS
GEN. PELVIC TENDERNESS
PELVIC - MASS
PELVIC - BLOOD (clots)
PERIODS NOT STARTED
PERIODS CEASED
PERIODS REGULAR
PERIODS IRREGULAR
L.M.P. NORMAL
L.M.P. LATE/OVERDUE
VAGINAL D/C PRESENT
VAGINAL D/C ABSENT
PREG. IMPOSSIBLE
PREG. POSSIBLE
PREG. CONFIRMED
FAINT/DIZZY PRESENT
FAINT/DIZZY ABSENT
PREV. GYN. HISTORY
NO PREV. GYN. HISTORY
```

BESTQUES.ASC

```
12  9 42 Appendicitis vs NSAP
The following features favor Appendicitis:
  Pain moving to RLQ.
  Aggravated by movement and coughing.
  Nausea, vomiting and anorexia.
  Facial Flushing.
  Focal RLQ tenderness.
  Rebound and guarding.
  Tender on R on rectal exam.
  Tender on R on vaginal exam.
13  7 49 Appendicitis vs PID
Is pain focal RLQ (Appx) or more diffuse (PID).
Nausea, vomiting AND anorexia favor Appendicitis.
Urinary symptoms, where present, favor PID.
Focal RLQ tenderness favors Appendicitis.
Flank tenderness favors Appendicitis.
Vaginal discharge favors PID.
Cervical tenderness favors PID.
14  6 57 Appendicitis vs UTI
Focal RLQ pain favors Appendicitis.
Flank pain favors UTI.
Frequency, dysuria favor UTI.
Focal RLQ tenderness favors Appendicitis.
Rebound, guarding favor Appendicitis.
Rectal tenderness, focal to R. side, favors Appendicitis.
15  5 53 Appendicitis vs Ovarian Cyst.
Pain MOVING to low Abd/RLQ favors Appendicitis.
Flushed, Temp. over 100.4F (38C) favors Appendicitis.
Scar, Distension favor Ovarian Cyst.
Abd. Mass favors Ovarian Cyst.
Abnormal bowel sounds favor Ovarian Cyst.
16  6 64 Appendicitis vs Ectopic
Focal RLQ pain favors Appendicitis. More diffuse, Ectopic.
Ectopic either presents SUDDENLY or after a few days.
Pallor favors Ectopic, Flushing favors Appendicitis.
Focal RLQ tenderness favors Appendicitis. More diffuse, Ectopic.
Late LMP, possible or confirmed pregnancy favors Ectopic.
Faintness/dizzyness favor Ectopic.
17  8 64 Appendicitis vs Incomplete Abortion
Pain localizing in RLQ favors Appendicitis.
Abortion pain usually mild/moderate, lower half.
Pain aggrav. by movement, coughing favors Appendicitis.
Nausea, vomiting, anorexia favor Appendicitis.
Tenderness - same considerations as for pain.
Rebound, guarding favor Appendicitis.
Rectal or vaginal tenderness on R favors Appendicitis.
```

```
Blood clots on vaginal exam + possible pregnancy favor Abortion.
23  6 42 NSAP vs PID
Pain outside lower half favors NSAP.
Tenderness outside lower half favors NSAP.
Rebound tenderness favors PID.
Vaginal discharge favors PID.
Previous History PID favors PID.
Rectal/vaginal tenderness favor PID.
24  5 36 NSAP vs UTI
Flank pain favors UTI.
Frequency/dysuria favor UTI.
Abnormal urine color favors UTI.
Flushed/feverish patient favors UTI.
Flank tenderness favors UTI.
25  7 52 NSAP vs Ovarian Cyst.
Focal pain RLQ favors Ovarian Cyst.
Pain aggravated by movement favors Ovarian Cyst.
Short history, severe pain favors Ovarian Cyst.
Focal lower quadrant tenderness favors Ovarian Cyst.
Rebound, Guarding, Rigidity favor Ovarian Cyst.
Mass, altered Bowel Sounds favor Ovarian Cyst.
Rectal/Vaginal tenderness favor Ovarian Cyst.
26  7 65 NSAP vs Ectopic
Ectopic pain tenderness usually lower half or one lower quadrant.
Pain aggrav. by movement, respiration favors ectopic.
Severe pain favors ectopic - often with previous mild pain.
Pallor favors ectopic, flushing favors NSAP.
Rebound, Guarding, Rigidity favor Ectopic.
LMP overdue, possible pregnancy, favors Ectopic.
Faintness or dizzyness favors Ectopic.
27  7 61 NSAP vs Incomplete Abortion
Pain of abortion usually lower half, short history, mild/nod.
Nausea, vomiting, anorexia favor NSAP.
Previous similar episodes favor NSAP.
In abortion, tenderness usually absent, or mild lower half.
Rebound, Guarding RARE in abortion.
In abortion, usually LMP late/overdue.  PG often confirmed.
Cardinal sign of abortion.  Blood + clots posterior vault.
34  7 49 PID vs UTI
Flank pain suggests UTI.
Frequency, dysuria may occur in either condition.
Flank tenderness suggests UTI.
Rebound tenderness suggests PID.
Vaginal discharge suggests PID.
Previous History PID suggests PID.
Cervical tenderness suggests PID.
35  7 48 PID vs Ovarian Cyst.
Focal pain constant in RLQ favors Ovarian Cyst.
Aggrav. of pain by movement favors Ovarian Cyst.
```

```
Severe pain favors Ovarian Cyst.
G.I. symptoms favor Ovarian Cyst.
Distension, mass favor Ovarian Cyst
Focal tenderness (RLQ) favors Ovarian Cyst.
Vaginal discharge, Cx tenderness favor PID.
36  7 56 PID vs Ectopic
Some days pain, suddenly worse, favors Ectopic.
Nausea, vomiting favor Ectopic.
Previous similar episodes favor PID.
Pallor + Tachycardia favors Ectopic.
Once ruptured rapid silent abdomen favors Ectopic.
Late/overdue period, possible pregnancy, favors Ectopic.
Faintness/dizzyness favors Ectopic.
37  8 52 PID vs Abortion
Pain of Abortion usually low 1/2, short duration.
Pain aggrav. by movt. coughing, resp. favors PID.
Previous episodes/past history of PID favor PID.
Temperature <98.6F (36.9C) favors Abortion.
Rectal, vaginal tenderness favor PID.
LMP overdue, pregnancy possible, favor Abortion.
Cardinal sign Abortion, blood clots posterior vault.
Rebound, guarding favor PID.
45  7 45 UTI vs Ovarian Cyst
Focal/RLQ pain/tenderness favor Ovarian Cyst.
Flank pain/tenderness favor UTI.
Frequency/dysuria favor UTI.
Flushing/pyrexia favor UTI.
Distension/mass favor Ovarian Cyst.
Rebound/guarding favor Ovarian Cyst.
Rectal/vaginal tenderness favor Ovarian Cyst.
46  7 52 UTI vs Ectopic
Flank pain/tenderness favor UTI.
Frequency/dysuria favor UTI.
Flushing favors UTI, pallor favors Ectopic.
Silent abdomen, decreased movement favors Ectopic.
Rebound, guarding, rigidity favor Ectopic.
Late/Overdue LMP, possible pregnancy, favor Ectopic.
Faintness or dizziness favor Ectopic.
47  8 56 UTI vs Abortion
Pain in Abortion usually low 1/2 moderate.
Flank pain/tenderness favors UTI.
Frequency/dysuria favors UTI.
"G.I. Symptom" favor UTI.
Flushing favors UTI.
Absence of tenderness favors Abortion.
Late overdue LMP/possible pregnancy favor Abortion.
Cardinal sign of abortion - blood/clots posterior vault.
56  6 55 Ovarian Cyst vs Ectopic
Ovarian Cyst pain usually starts focally, severe.
```

BESTQUES.DAT (cont'd)

```
Ectopic pain usually starts milder, diffuse.
Pallor, tachycardia favor Ectopic.
Focal lower quadrant tenderness favors Ovarian Cyst.
Possible pregnancy, late/overdue period favors Ectopic.
Faintness/dizziness favors Ectopic.
57  8 58 Ovarian Cyst vs Abortion
Pain of Abortion usually mild, diffuse, low 1/2.
Pain of Ovarian Cyst worse on movement, coughing, resp.
G.I. symptoms favor Ovarian Cyst.
Overt distress with pain favors Ovarian Cyst.
Silent abdomen/poor movement favors Ovarian Cyst.
Absence of tenderness, rebound, guarding favors Abortion.
Possible pregnancy, late/overdue period favors Abortion.
Cardinal sign of Abortion - blood + clots posterior vault.
67  1 58 Ectopic vs Abortion
Cardinal sign of Abortion - blood + clots posterior vault.
99
```

PHRASE.ASC

```
11 the right upper quadrant
12 the left upper quadrant
13 the right lower quadrant
14 the left lower quadrant
15 the upper half of the abdomen
16 the lower half of the abdomen
17 the right half of the abdomen
18 the left half of the abdomen
19 the periumbilical area
20 a rather diffuse area over the whole abdomen
21 the right flank area
22 the left flank area
23 no abdominal area
24 the right upper quadrant
25 the left upper quadrant
26 the right lower quadrant
27 the left lower quadrant
28 the upper half of the abdomen
29 the lower half of the abdomen
30 the right half of the abdomen
31 the left half of the abdomen
32 the periumbilical area
33 a diffuse area over the whole abdomen
34 the right flank area
35 the left flank area
36 no abdominal area
37 intermittent
38 fairly constant
39 colicky
40 a moderate pain
41 a severe pain
42 movement
43 coughing
44 breathing
45 food
46 OTHER
47 nothing the patient does
48 seems to be getting better
49 seems to be about the same as when it first began
50 seems to be getting worse
51 began less than 12 hours ago
52 began 12-24 hours ago
53 began 24-48 hours ago
54 began greater than 48 hours ago
55 lying still
56 vomiting
```

PHRASE.DAT (cont'd)

```
 57 antacids
 58 eating bland foods
 59 OTHER
 60 nothing he does
 61 nausea
 62 no nausea
 63 vomiting.
 64 no vomiting.
 65 normal bowel movements
 66 constipation
 67 diarrhea
 68 blood in his stool
 69 mucus in his stool
 70 decreased.
 71 normal.
 72  There is a history of jaundice in the recent past.
 73  There is no history of recent jaundice.
 74 normal urination
 75 frequency
 76 dysuria
 77 dark urine
 78 hematuria
 79  The patient has a history of previous indigestion
 80  There is no history of previous indigestion
 81 and he has had similar pain in the past.
 82 and he has had no similar pain in the past.
 83 abdominal surgery.
 84 no abdominal surgery.
 85 past GI illnesses.
 86 no pertinent G-I illnesses.
 87 medication for this illness.
 88 no medication for this illness.
 89 less than 98.6
 90 98.6-100.2
 91 100.3-102
 92 greater than 102
 93 less than 80
 94 80-99
 95 greater than 99
 96 less than 90/
 97 90-129/
 98 greater than 129/
 99 less than 70
100 70-89
101 greater than 89
102 normal.
103 distressed.
104 anxious.
105 normal
```

```
106 pale
107 flushed
108 jaundiced
109 cyanotic
110 less than 8,000
111 8,100-10,000
112 10,100-12,000
113 12,000-15,000
114 greater than 15,000
115  Inspection of the abdomen reveals no abnormalities.
116  Inspection of the abdomen reveals visible peristaltic waves.
117  Inspection of the abdomen reveals decreased abdominal movement.
118  Abdominal surgical scars are present.
119  There are no surgical scars on the abdomen.
120  Voluntary guarding is present
121  Voluntary guarding is absent
122 guarding (rigidity) is present.
123 guarding is absent.
124  Bowel sounds are normal.
125  Bowel sounds are absent.
126  Bowel sounds are hyperactive.
127  Abdominal distention is present.
128  There is no apparent abdominal distention.
129  There is an abdominal mass
130  There are no masses
131 the right upper quadrant
132 the left upper quadrant
133 the right lower quadrant
134 the left lower quadrant
135 the upper half of the abdomen
136 the lower half of the abdomen
137 the right half of the abdomen
138 the left half of the abdomen
139 the periumbilical area
140 a diffuse area over the whole abdomen
141 the right flank area
142 the left flank area
143 not noted
144 Murphy's sign is present.
145 Murphy's sign is not present.
146 tenderness is noted.
147 tenderness is absent.
148 no tenderness or mass
149 a mass
150 tenderness on the left side
151 tenderness on the right side
152 generalized tenderness
153  The pelvic examination is normal.
154 right adnexal tenderness
```

PHRASE.DAT (cont'd)

```
155 left adnexal tenderness
156 cervical tenderness
157 general tenderness
158 a pelvic mass
159  Clots are noted in the vaginal vault.
160 have not started.
161 have ceased.
162 are regular.
163 are irregular.
164  Her last period was normal
165  Her last period was late/overdue
166  She has had a recent vaginal discharge
167  She has had no recent vaginal discharge
168 is impossible.
169 is possible.
170 has been confirmed.
171 has
172 has not
173 a
174 no
```

Appendix G
MAKE Script file for creating ABDX.EXE

ABDX

```
abdx.obj:       abdx.bas include.bas
     bc abdx.bas /o;

abdxnara.obj:   abdxnara.bas include.bas
     bc abdxnara.bas /o;

abdxonly.obj:   abdxonly.bas include.bas
     bc abdxonly.bas /o;

abdxshar.obj:   abdxshar.bas include.bas
     bc abdxshar.bas /o;

abdxsub1.obj:   abdxsub1.bas include.bas
     bc abdxsub1.bas /o;

abdxsub2.obj:   abdxsub2.bas include.bas
     bc abdxsub2.bas /o;

abdxsub3.obj:   abdxsub3.bas include.bas
     bc abdxsub3.bas /o;

abdxsub4.obj:   abdxsub4.bas include.bas
     bc abdxsub4.bas /o;

template.obj:   template.bas include.bas
     bc template.bas /o;

dates.obj: dates.bas
     bc dates.bas /o;

fsf600.obj:     fsf600.bas include.bas
     bc fsf600.bas /o;

abdxsub5.obj:  abdxsub5.bas
     bc abdxsub5.bas /o/e/x;

abdxsub6.obj:  abdxsub6.bas include.bas
     bc abdxsub6.bas /o;

win.lib: fprint.obj intrpt.obj cipher.obj abdxsub5.obj
```

```
     lib win.lib -+fprint -+intrpt -+cipher -+abdxsub5;

abdx.exe:  abdx.obj abdxnara.obj template.obj dates.obj \
           abdxonly.obj abdxshar.obj \
       abdxsub1.obj abdxsub2.obj abdxsub3.obj \
           abdxsub4.obj abdxsub6.obj fsf600.obj win.lib
     link /e abdx abdxnara abdxsub1 abdxsub2 \
 abdxsub3 abdxsub4 abdxonly abdxshar fsf600 template \
 dates abdxsub6,,abdx.map,win.lib;
```

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS
NA

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
NSMRL Report 1148

5. MONITORING ORGANIZATION REPORT NUMBER(S)
NA'

6a. NAME OF PERFORMING ORGANIZATION
Naval Submarine Medical
Research Laboratory

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Naval Medical Research and Development
Command

6c. ADDRESS *(City, State, and ZIP Code)*
Box 900, Naval Submarine Base NLON
Groton, CT 06349-5900

7b. ADDRESS *(City, State, and ZIP Code)*
Naval Medical Command, National Capital
Region, Bethesda, MD 20814-5044

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Same as 7a

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
Same as 7b

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO.
63706N	M0095	005	5010

11. TITLE *(Include Security Classification)*
(U) ABDX - A Decision Support System for the Management of Acute Abdominal Pain - Version 3.0 - PROGRAMMER'S MANUAL

12. PERSONAL AUTHOR(S)
LCDR David Southerland, MC, USN, and Karen Fisherkeller

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM 10/88 TO 9/89

14. DATE OF REPORT *(Year, Month, Day)*
1989 October 31

15 PAGE COUNT

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
computer assisted; diagnosis; abdominal pain; appendicitis cholecystitis, perforated duodenal ulcer; renal colic;

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

At sea, the Independent Duty Corpsman (8402) is responsible for the diagnosis and management of illnesses. He must decide whether to treat the patient aboard the ship, or, if necessary, make recommendations regarding the evacuation of the patient. The corpsman's laboratory facilities are limited and, in most instances, he is unable to communicate with short-based medical facilities.

A computer based medical decision support system (ABDX) was developed at the Naval Submarine Medical Research Laboratory (NSMRL) to assist the corpsman in the diagnosis, triage, and management of patients who present with acute abdominal pain at sea. The original system was designed for use with the Tektronix 4051 to which the corpsman was to have access. In practice, the corpsman's access has been very limited due to the use of the machine by other departments for tactical programs. The medical support system was then modified and additions were made to allow the system to run on an IBM PC or

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

22a. NAME OF RESPONSIBLE INDIVIDUAL
Susan D. Monty, Publications Office

22b. TELEPHONE *(Include Area Code)*
(203) 449-3967

22c OFFICE SYMBOL
421

Previous editions are obsolete.
S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

18. cont. Small bowel obstruction; ectopic pregnancy; ovarian cyst; ovarian cyst; pelvic inflammatory disease; threatened abortion; urinary tract infection; and non-specific abdominal pain

19. cont.
compatible (MS-DOS) microcomputer. During FY89, a gynecological database for acute abdominal pain was incorporated into the program, resulting in a new product, ABDX version 3.0.

The original non-gynecological portion of the program is intended for use with males, between the ages of 17 and 70, and provides medical support for the 5 most common causes of serious abdominal pain in this population. The 5 diseases are: Appendicitis, Cholecystitis, Perforated Duodenal Ulcer, Renal colic, and Small Bowel Obstruction. In addition, a sixth category, Non-Specific Abdominal Pain, is used to represent those cases of non-serious abdominal pain. The gynecological portion of the program is used with females between the ages of 15 and 50, and provides medical support for the six most common causes of serious abdominal pain in this female population. The diseases are: Appendicitis, Ectopic Pregnancy, Ovarian Cyst, Pelvic Inflammatory Disease, Threatened Abortion, and Urinary Tract Infection. In addition, a seventh category, Non-Specific Abdominal Pain, is again used to represent those cases of non-serious abdominal pain.

The program consists of a diagnostic module, which provides diagnostic and treatment suggestions for each of the abdominal pain diseases, a training module, which tests the corpsman's accuracy in abstracting data from patient narratives, and a SF-600 generation module, which prints medical record entries based on patient data entered into the program.

This report is a programmer's reference manual for the ABDX system implemented on MS-DOS computers. The manual briefly describes the functions of programs designed for the corpsman user and other programs designed for the programmer making modifications to ABDX. The Microsoft QuickC, Microsoft QuickBASEC, and Microsoft MACRO assembly language source code listings for all the programs are listed. In addition, the formats for the help files and data files are described.

Familiarity with QuickBASIC is required to modify ABDX or to use this manual effectively to identify program malfunctions.

This report replaces NSMRL Report No. 1113.