

U. S. ARMY

Technical Memorandum 21-71

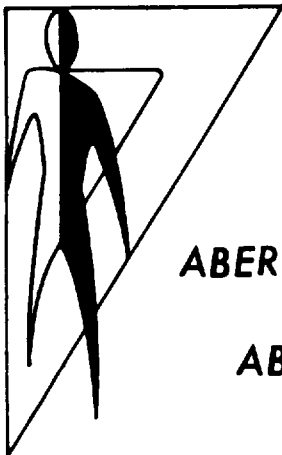
REFERENCE SET
DO NOT REMOVE

VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS

Michael Abbamonte
Selby H. Evans

November 1971
AMCMS Code 5011.11.85000

HUMAN ENGINEERING LABORATORIES



ABERDEEN RESEARCH & DEVELOPMENT CENTER

ABERDEEN PROVING GROUND, MARYLAND

Approved for public release;
distribution unlimited.

**Destroy this report when no longer needed.
Do not return it to the originator.**

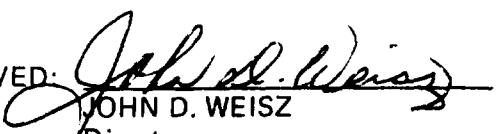
**The findings in this report are not to be construed as an official Department of the
Army position unless so designated by other authorized documents.**

**Use of trade names in this report does not constitute an official endorsement
or approval of the use of such commercial products.**

VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS

Michael Abbamonte
Selby H. Evans

November 1971

APPROVED: 
JOHN D. WEISZ
Director
Human Engineering Laboratories

HUMAN ENGINEERING LABORATORIES
U. S. Army Aberdeen Research & Development Center
Aberdeen Proving Ground, Maryland

Approved for public release;
distribution unlimited.

FOREWORD

This research was supported by the Department of Defense, Project THEMIS Contract (DAAD05-68-C-0176), under the Department of the Army, to the Institute for the Study of Cognitive Systems through the TCU Research Foundation. Further reproduction is authorized to satisfy needs of the U. S. Government.

The development of the VARGUS 7 program was supported by NIH Grant Number 11730-01.

The development of the VARGUS 9 program was supported in part by Grant Number MH-12223-01 from National Institute of Health and in part by THEMIS Contract DAAD05-68-C-0176.

The development of the VARGUS 10 program was supported in part by Grant Number MH-12223-01 from National Institute of Health and in part by THEMIS Contract DAAD05-68-C-0176.

ABSTRACT

This document describes a collection of computer programs developed for use in research on human pattern perception. The overall orientation which guided the development of the VARGUS (Variable Generator of Unfamiliar Stimuli) pattern-generation programs and the historical backgrounds of each category (VARGUS 7, 9 and 10) are related in the first section. The second section provides documentation, sample output and summary for each program and subroutine.

CONTENTS

FOREWORD	ii
ABSTRACT	iii
SECTION I. THREE PATTERN-GENERATION SYSTEMS: THEIR CHARACTERISTICS AND APPLICATIONS	1
VARGUS 7	2
VARGUS 9	5
VARGUS 10	9
REFERENCES	13
SECTION II. INTRODUCTION TO VARGUS PATTERN GENERATION PROGRAMS	17
APPENDIX	19
FIGURES	
1. Six Examples of Vargus 7 Patterns Plotted as Histograms and as Graphs	4
2. Examples of Vargus 9 Patterns in Histograms and Graphic Format	7
3. Examples of Polygons Produced by a Modification of the Vargus 9 System Developed by Aiken, L. S. & Brown, D. R.	8
4. Example of a Drawing Prepared by an Artist Under Instructions to Produce Familiar Figures Under Constraints Described in the Text	10
5. Examples of Vargus 10 Patterns	11
TABLES	
1. Vargus 7 Control Table: Selected Redundancy Values and Corresponding Values of $\underline{H}$ and $\underline{p_{max}}$	2
2. Vargus 9 Control Table: Values of $\underline{p_{max}}$ to Achieve Selected Levels of Constraint Redundancy	6

VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS

SECTION I

THREE PATTERN-GENERATION SYSTEMS: THEIR CHARACTERISTICS AND APPLICATIONS

This collection of programs has been developed for use in research on pattern perception. The overall orientation which guided the development of these particular programs is schema theory as described by Evans (22). A pattern, in the context of schema theory, is a set of common characteristics or attributes associated with things belonging to the same pattern class. Individual instances of the class would not be identical nor would a particular instance necessarily exhibit all of the characteristics of the pattern.

A good example of such a pattern is a person's handwriting. Each person has a number of distinctive features associated with his handwriting. No feature is perfectly reliable in the sense that it is reproduced exactly in each handwriting sample. But taken together, the set of features is sufficient to permit a person's handwriting to be recognized or identified -- at least by experts.

The programs were intended to provide components of a methodology for studying human pattern perception. They are designed to offer the researcher options in choosing levels of familiarity and in controlling relevant stimulus parameters. To a substantial degree, these two options are mutually exclusive, because familiar stimuli are presumably encoded by subjects in terms of previously learned characteristics. Thus a determination of the relevant parameters would have to be based on a knowledge of human encoding processes--a knowledge which is, of course, not yet available. Unfamiliar patterns, on the other hand, can be designed in such a way that the subjects will find and respond to the parameters the experimenter has chosen to make relevant. (This assertion is supported in the studies discussed in later sections of this paper.) These generating systems are, therefore, useful in providing a range of choice with respect to these two properties.

The programs seem to have some fairly general utility for perception research, and this laboratory has received a number of requests either for programs or for copies of the stimuli. This document has been prepared to make the programs themselves more generally available. None of the programs are particularly difficult or complex. Most of them could be written rather quickly by any reasonably competent programmer. Nevertheless, experimenters may find it convenient to have the programs already available and debugged.

This set of programs may be subdivided into three general categories depending on the main pattern-generation program. These categories are: VARGUS 7, VARGUS 9 and VARGUS 10 (the term VARGUS was originally an acronym for Variable Generator of Unfamiliar Stimuli). In subsequent sections, each of the generating systems will be described in detail and some of its applications will be noted.

VARGUS 7

The VARGUS 7 system was designed to produce very unfamiliar patterns, but ones which allowed for substantial control and manipulation of relevant pattern parameters. The theory behind the system is given by Evans (24). The VARGUS 7 generates segments of a Markov process, and the theory of Markov processes is used to achieve a number of features which might be desirable in the study of pattern perception. It allows the sampling of unfamiliar stimuli from a defined population and permits the independent control of uncertainty (in bits per stimulus) and of redundancy. In the long run, all pattern elements occur with equal frequency in every pattern position (1st element, 2nd element, etc.). Moreover, one schema can be substituted for another without altering these characteristics.

The schema is expressed in the form of favored transitional probabilities from element to element; taken together, the set of favored elements constitutes a Most Probable Sequence (MPS). Different schemata are produced by choosing different permutations of the elements to constitute the MPS. The common characteristics of VARGUS 7 patterns are thus local features (pairs or triples of elements), and these are the only features that can be used to identify a particular pattern.

In the basic generating program, the elements of the Markov process are the integers from 1 to 7. The resulting instances are strings of integers; final stimuli are produced by mapping the integers into other kinds of elements, as described below.

Table 1 gives parameter values to achieve selected levels of redundancy in generating VARGUS 7. The value $p_{\max}$ is used as the probability of the favored transition for each element; each of the other transitions is assigned a probability $(1 - p_{\max})/6$. Under these conditions, the redundancy and uncertainty of the Markov process are as given in Table 1.

TABLE 1
VARGUS 7 Control Table: Selected Redundancy Values
and Corresponding Values of H and $p_{\max}$

Redundancy (Percent)	H (Bits per Column)	$p_{\max}$
0	2.81	.1428
20	2.24	.52
30	1.97	.61
40	1.69	.69
50	1.42	.76
60	1.10	.83
70	.84	.88
80	.55	.93

VARGUS 7 stimuli have been used most commonly by mapping the strings of numbers into column heights in histograms or altitudes in graphs (Fig. 1). In this latter form, the stimuli have some resemblance to time series data and their common characteristics are like transients occurring at random times.

These stimuli have been used in a substantial number of studies calling for subjects to distinguish between stimuli representing different schemata. It has been well established that subjects can accomplish this task (26, 15, 25, 19, 7).

Rosser (37) presented stimuli similar to those produced by the VARGUS 7 in an auditory format using the elements of the Markov processes to determine particular tone; she also found discrimination between classes of patterns, as did Copeland (13) with similar auditory patterns generated by the VARGUS 7 system. More recently Pollack (34) translated similar Markov processes into pulse trains, converted them to sound, and again demonstrated that subjects could spontaneously discriminate between classes of patterns. Similar effects were demonstrated by Hollier and Evans (31), using a language-like mode of presentation in which the elements of the Markov process were mapped into syllables.

The stimuli generated by the VARGUS 7 system have also been used in modeling efforts designed to emulate the behavior described above (21, 4, 8).

In addition to the basic demonstration of schematic concept formation, the VARGUS 7 stimuli have been used in a number of studies investigating the effect of constraint redundancy (23) and the effect of knowledge of results on schematic concept formation (10, 18, 20, 11, 39, 5). These studies have tended to show that constraint redundancy is an important variable in this context, and that schematic concept formation is accomplished reasonably well at 70 percent redundancy, marginally at 50 percent redundancy, and probably not at all at 40 percent redundancy. These figures, of course, apply to the kinds of stimuli used in the experiments-standard VARGUS 7 stimuli containing 12 to 24 elements and represented as histograms. There is no reason to expect a redundancy measure to have enough generality to permit its application to other kinds of patterns.

The effects of knowledge of results appear to be a bit more complex, but there is some suggestion (3, 41) that knowledge of results is ineffective and may actually interfere with performance when the patterns are sufficiently redundant (e.g., 70%). At lower redundancies, knowledge of results appears to facilitate performance.

Schema theory proposes that subjects remember an individual stimulus by encoding it in terms of its deviations from the schema. The VARGUS 7 stimuli have been used in studies of this question (16, 17, 32); there appears to be some evidence in support of the schema-plus-correction hypothesis when reproduction tasks are used. On the other hand, a study by Brown and Rebbin (10) tends to suggest that in pattern-classification tasks, at least, encoding of the schema plays very little role in the classification of the VARGUS 7 stimuli. Since encoding the schema is presumably a prerequisite to memory storage in the form of schema plus correction, further research on this point seems to be needed.

A set of VARGUS 7 patterns in numerical form has been prepared and made available (2); for many research purposes this standard set of VARGUS 7 stimuli would be suitable. That monograph also described a measure which allows each stimulus to be represented in terms of a single number indicating how closely it conforms to the prototype. This measure is computed by the subroutine ADHER, listed along with VARGUS 7.

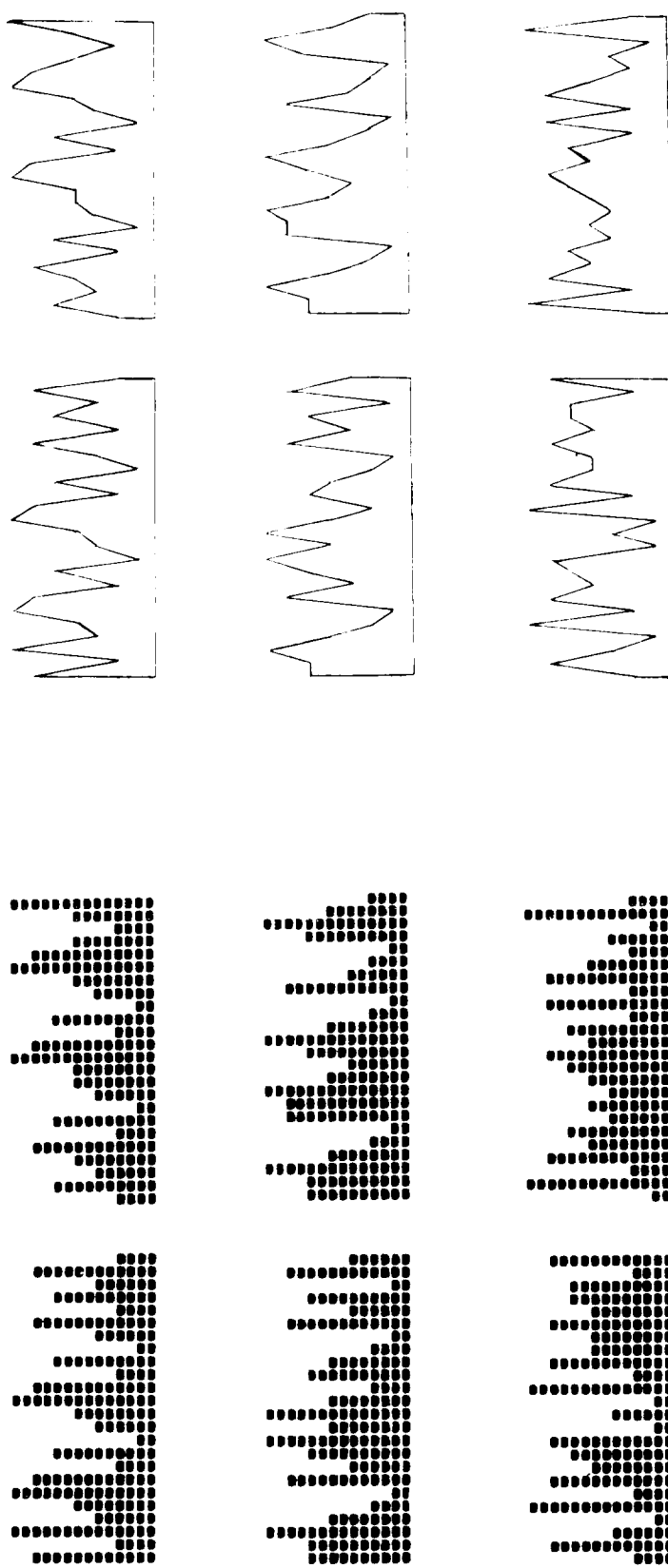


Fig. 1. SIX EXAMPLES OF VARGUS 7 PATTERNS PLOTTED AS HISTOGRAMS AND AS GRAPHS
 (The histogram plots were produced by an IBM 1401 printer using the suppress-space feature to overprint the characters zero, capital O and asterisk. Each row represents a different schema.)

VARGUS 9

The VARGUS 9 stimuli are generated in accordance with a method described by Evans and Mueller (27). As with the VARGUS 7, the stimuli are randomly sampled from a defined population, with information and redundancy subject to control and manipulation. The following additional objectives were also achieved:

1. The population is made up of variants which consist of independent and measurable deviations from a defined prototype.
2. It is reasonable to treat the deviations as interval scale deviations from a mean, so that interval scale measures can be applied.

In the VARGUS 9 system a prototype is a string of integers ranging in value from 3 to 7. Any such string of no more than 32 elements will serve, although one would normally choose the string so as to avoid any systematic ordering of the integers. Variants are produced by adding to the prototype integer a deviation term, which is an integer ranging from -2 to +2. The probability distribution for this deviation is unimodally and symmetrically distributed about zero.

The procedure for determining the distribution of the deviation, as suggested by Evans and Mueller (27), is to choose $p(0) = p_{\max}$ and then set the other values as follows:

$$\begin{aligned} p(1) &= p(-1) = (1 - p_{\max})/3, \\ p(2) &= p(-2) = (1 - p_{\max})/6. \end{aligned} \tag{1}$$

Under these rules, the required values of $p_{\max}$ for selected redundancy levels are as given in Table 2. In general, the redundancy levels for exhibiting schematic concept formation are similar to those for VARGUS 7; 50 percent redundancy is marginal, while 70 percent gives relatively good results. As with the VARGUS 7, these comments apply only to the kinds of stimuli used in the studies reported below.

VARGUS 9 stimuli have been studied principally in displays mapping their numbers into column heights of histograms or into altitudes on a graph (Fig. 2). In this form, the stimuli might be likened to personality profiles or to other graphic displays in which the variability from column to column is relatively independent. In contrast to the VARGUS 7, the common characteristics of the VARGUS 9 stimuli lie in global features and in the overall shape. While local features are to some extent usable, they are less reliable than in comparable VARGUS 7 stimuli.

These stimuli have also been used in studies of schematic concept formation (35, 29, 30). Again, the research has shown schematic concept formation to occur and there is some possibility that it is easier with VARGUS 9 stimuli than with VARGUS 7 stimuli (40). These stimuli have also been used in polar-coordinate form (33, 1) in which the numbers constituting the patterns are plotted as length of radii. In this form they constitute a closed polygon, and research with them has provided a link between schematic concept formation studies and the Brown and Owen psychophysics of form (12). Aiken and Brown (1) have extended this research with a modified version of the VARGUS 9 pattern-generating system (Fig. 3). One of the major results of this research has been the demonstration of subjects' sensitivity to the statistical distributions of features within the particular set of patterns on which judgments are being made. Similar effects have been shown by Dansereau, Fenker and Evans (14), who demonstrated that pattern storage in memory is substantially influenced by the clusters of patterns constituting schema families.

TABLE 2

VARGUS 9 Control Table: Values of $p_{\max}$ to Achieve
Selected Levels of Constraint Redundancy

Redundancy	$p_{\max}$
30%	.31
40%	.53
50%	.66
60%	.77
70%	.84

The perceived similarity of VARGUS 9 stimuli, when they are drawn from the same cluster, or schema family, appears to be determined largely by the Euclidean distance between the pair of stimuli (36); a correlation of .88 was found between judged similarity and interpoint distance. (This quantity is calculated by the subroutine DVAR, listed with VARGUS 9.) The strength of this relationship supports the assertion that interval scale measures are appropriate with VARGUS 9 stimuli and suggests that these stimuli might have some general usefulness in cases where it is desirable to control the subjective similarity of patterns.

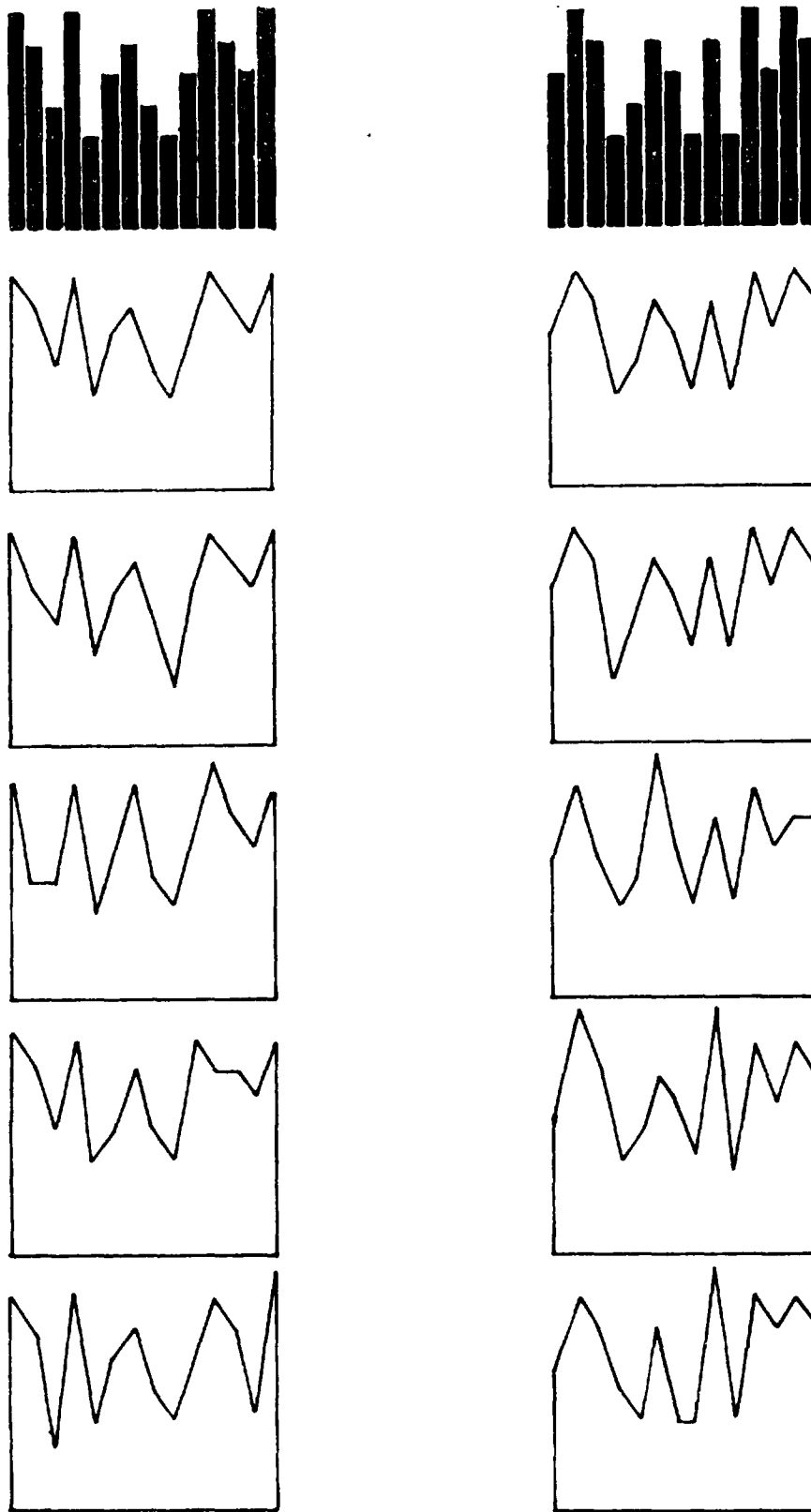


Fig. 2. EXAMPLES OF VARGUS 9 PATTERNS IN HISTOGRAMS AND GRAPHIC FORMAT
 (Each column represents a different schema. The histograms and the
 topmost graphs are prototypes. Instances below these represent
 increasing deviations from the prototypes.)



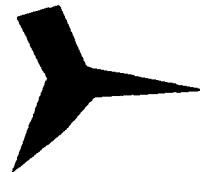
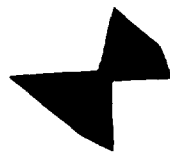
PROTO TYPES



LOW VARIABILITY



MODERATE VARIABILITY



HIGH VARIABILITY

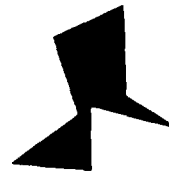


Fig. 3. EXAMPLES OF POLYGONS PRODUCED BY A MODIFICATION OF THE VARGUS 9 SYSTEM DEVELOPED BY AIKEN, L. S. & BROWN, D. R. (1)

VARGUS 10

The previous generating programs produced stimuli that were intended to be relatively unfamiliar. The VARGUS 10 system was designed to provide stimuli which are of intermediate familiarity. In particular, the VARGUS 10 patterns are intended to incorporate relatively familiar components into overall configurations which are relatively unfamiliar.

The method for obtaining familiar configurations was based on a method devised by Shannon and Weaver (38). In studying the transitional probabilities of English grammar, Shannon used humans as repositories of the statistical characteristics of grammar. In the same way, one might expect that humans are repositories of the statistical characteristics of the visual environment and that they use these characteristics when they draw figures. In accord with that logic, a commercial artist was engaged to draw a number of figures in such a way as to allow them to be decomposed into interchangeable components.

The original drawings were made in a 3 x 3 matrix (Fig. 4); the eight outer cells of the matrix had marks on the midpoints of the lines separating them from their adjoining cells. The artist was instructed to draw each figure by starting at one of the midpoints and drawing any line he pleased through the cell, so long as the line did not touch itself and did not cross the border of the cell except at the other midpoint, where the line could exit to the next cell. He was then to continue in the same fashion through the remaining cells until he returned to the starting point. He was encouraged to produce any familiar figures he could within those constraints.

The resulting drawings were then decomposed by taking the configuration in each cell as a separate unit. the matrix had four "corner" cells and four "side" cells. Since all corner cells, if properly rotated, had the same entry and exit points for the line, any corner cell could be used in any corner to construct a figure. The same condition applied to the side cells.

After the components were dissociated from their source figures, they were reviewed to eliminate components which were near duplicates of each other and to eliminate components which contained too much detail to be adequately represented by black dots in a 16 x 16 matrix. The resulting set--40 corner components and 34 side components--constituted the basic set out of which patterns could be generated.

The VARGUS 10 program selects components at random from these sets to fill each of the four corner cells and four side cells of a figure being constructed. The components are rotated to conform to the orientation of the assigned cells, and each component is then placed to comprise a 16 x 16 sub-matrix of a 48 x 48 matrix which contains the whole pattern (Fig. 5).

A number of associated programs have been prepared to manipulate the VARGUS 10 stimuli and to provide a measure--probably a crude one--of similarity. these programs are also provided in the VARGUS 10 section.

The VARGUS 10 generating system has not been the subject of as many published reports as have the VARGUS 7 and VARGUS 9. Stimuli generated with this system were used in conjunction with the elements of a model for human preprocessing by Zinser (42, 43), and they have been used in a study of feature selection by Hastings and Evans (28). This latter study established a methodology for obtaining judgments from people regarding the features they would use for discriminating patterns; the results indicated a preference for corner configurations and for projections.

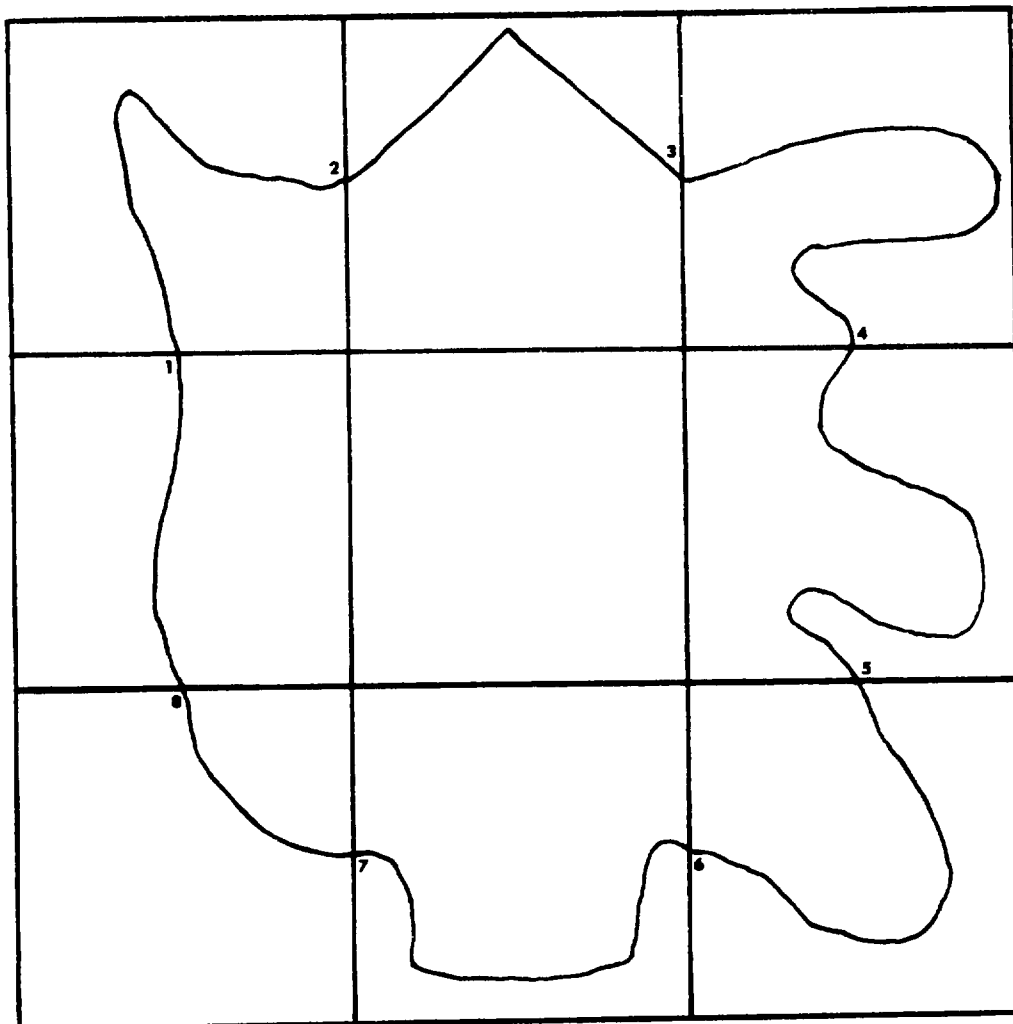


Fig. 4. EXAMPLE OF A DRAWING PREPARED BY AN ARTIST UNDER INSTRUCTIONS TO PRODUCE FAMILIAR FIGURES UNDER CONSTRAINTS DESCRIBED IN THE TEXT
(The drawing was termed "Old Witch.")

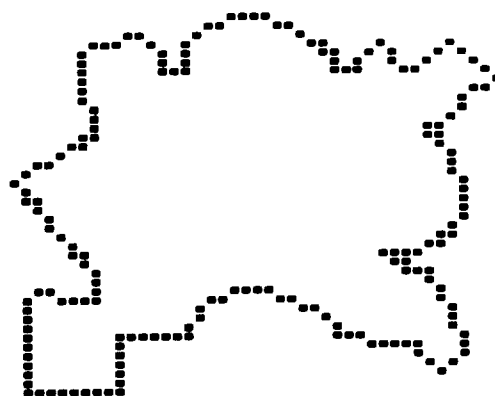
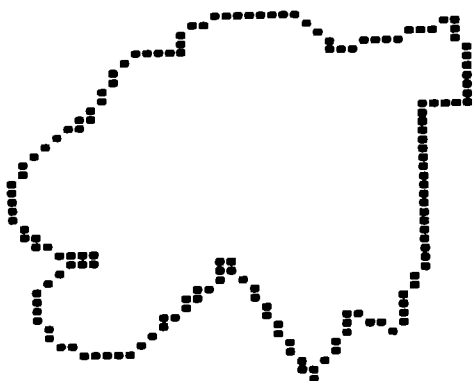
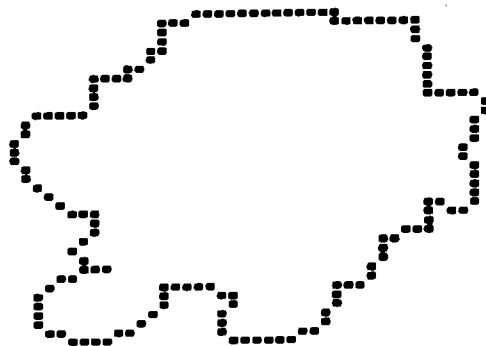
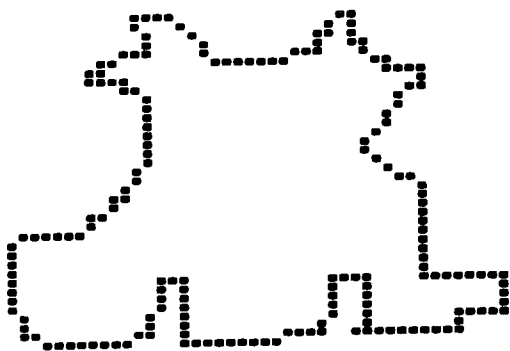
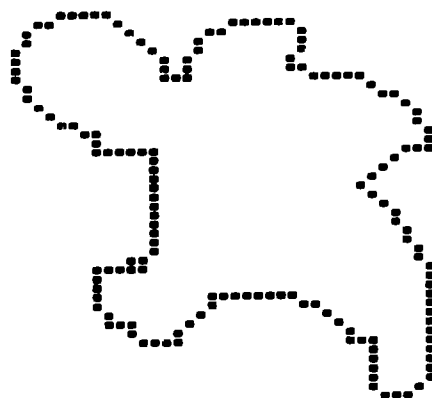
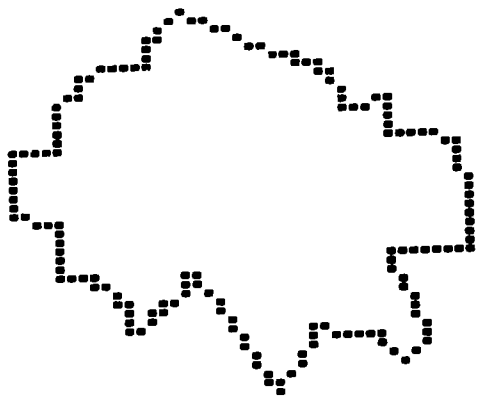


Fig. 5. EXAMPLES OF VARGUS 10 PATTERNS

A number of other studies of feature selection are possible with the VARGUS 10 stimuli; these studies have not been executed under the present research project only because other studies were of higher priority for the purposes of the project. For example, no study has been carried out to assess the adequacy of the similarity measure computed by CONG; doubtless, such a study would be of substantial interest.

One might assess the effectiveness of the overall similarity measure in predicting subjective similarity. One might also measure the similarity of the individual components and use these values in a regression equation to compare their predictive power with the predictive power of the overall similarity measure. (The difference between these two prediction methods is that the first assumes that the whole is the sum of its parts, while the second assumes that the whole is the best weighted linear combination of its parts.) One could also obtain subjective judgments of the similarities of the parts to determine the relative adequacy of these measures in predicting overall similarity as compared with the objective similarity measures. Such studies would shed additional light on the processes of feature selection and feature utilization.

REFERENCES

1. Aiken, L. S. & Brown, D. R. A feature utilization analysis of the perception of pattern class structure. Perception & Psychophysics, 1971, 9, 279-283(a).
2. Bersted, C. T., Brown, B. R., & Evans, S. H. A standard set of VARGUS 7 patterns at three levels of schematic redundancy. Psychonomic Monograph Supplements, 1968, 2 (13, Whole No. 29), 251-282.
3. Breckenridge, R. L., Jr., Rankin, W. C., & Wright, A. D. The effect of KR on ratings of pattern similarity. Psychonomic Science, 1969, 15, 223-224.
4. Brown, B. R. A model for schematic concept formation: Further development and assessment. Unpublished doctoral dissertation, Texas Christian University, 1969.
5. Brown, B. R. & Dansereau, D. F. Discrimination among schematic stimuli as a function of response mode, constraint redundancy and form of Markov rule. Psychonomic Science, 1969, 17, 197-198.
6. Brown, B. R. & Dansereau, D. F. Functional equivalence between same-different classifications and judged similarity of Markov patterns. Perception & Psychophysics, 1970, 7, 307-310.
7. Brown, B. R. & Evans, S. H. Perceptual learning in pattern discrimination tasks with two and three schema categories. Psychonomic Science, 1969, 15, 101-103.
8. Brown, B. R. & Evans, S. H. Further applications of the random adaptive module (RAM) system to schema theory. Technical Memorandum 4-70, Human Engineering Laboratories, U. S. Army Aberdeen Research & Development Center, Aberdeen Proving Ground, Md, 1970.
9. Brown, B. R. & Rebbin, T. J. Simultaneous versus sequential discriminations of Markov generated stimuli. Perception & Psychophysics, 1970, 8, 353-357.
10. Brown, B. R., Walker, D. W., & Evans, S. H. Schematic concept formation as a function of constraint redundancy and knowledge of results. Psychonomic Science, 1968, 11, 75-76.
11. Brown, B. R., Walker, D. W., & Evans, S. H. Concept formation as a function of perceptual pretraining and knowledge of results. Psychonomic Science, 1969, 14, 71-72.
12. Brown, D. R. & Owen, D. H. The metrics of visual form: Methodological dyspepsia. Psychological Bulletin, 1967, 68, 243-259.
13. Copeland, S. Pattern perception in the auditory modality. Unpublished master's thesis, Texas Christian University, 1970.
14. Dansereau, D. F., Fenker, R. M., & Evans, S. H. Visual pattern perception: Encoding and storage of schematic versus random patterns. Paper presented at American Psychological Association Conference, Miami Beach, January 1970.

15. Edmonds, E. M. & Edmonds, S. C. Schema mediation in categorization learning. Psychonomic Science, 1969, 14, 196.
16. Edmonds, E. M. & Evans, S. H. Schema learning without a prototype. Psychonomic Science, 1966, 5, 247-248.
17. Edmonds, E. M., Evans, S. H., & Mueller, M. R. Learning how to learn schemata. Psychonomic Science, 1966, 6, 177-178.
18. Edmonds, E. M., Mueller, M. R., & Evans, S. H. Effects of knowledge of results on mixed schema discrimination. Psychonomic Science, 1966, 6, 377-378.
19. Edmonds, E. M. & Mueller, M. R. Schema discrimination without external reinforcement. Perceptual and Motor Skills, 1967, 24, 436-438.
20. Edmonds, E. M. & Mueller, M. R. Effects of incidental training and reinforcement on mixed schema learning. Psychonomic Science, 1968, 10, 75-76.
21. Evans, S. H. A model for perceptual category formation. Part I. Unpublished doctoral dissertation, Texas Christian University, 1964.
22. Evans, S. H. A brief statement of schema theory. Psychonomic Science, 1967(a), 8, 87-88.
23. Evans, S. H. Redundancy as a variable in pattern perception. Psychological Bulletin, 1967(b), 67, 104-113.
24. Evans, S. H. VARGUS 7: Computed patterns from Markov processes. Behavioral Science, 1967(c), 12, 323-349.
25. Evans, S. H. & Arnoult, M. D. Schematic concept formation: Demonstration in a free sorting task. Psychonomic Science, 1967, 9, 221-222.
26. Evans, S. H. & Edmonds, E. M. Schema discrimination as a function of training. Psychonomic Science, 1966, 5, 303-304.
27. Evans, S. H. & Mueller, M. R. VARGUS 9: Computed stimuli for schema research. Psychonomic Science, 1966, 6, 511-512.
28. Hastings, D. W. & Evans, S. H. Selection of local features for pattern identification: An exploratory study. Technical Memorandum 5-70, Human Engineering Laboratories, U. S. Army Aberdeen Research & Development Center, Aberdeen Proving Ground, Md., 1970.
29. Hastings, M. A., Dansereau, D. F., & Dixon, T. R. Influence of pattern variance and overt responding on subjective categorization in a two schemata SCF task. Psychonomic Science, 1969, 16, 325-327.
30. Hastings, M. A. & Evans, S. H. Schematic concept formation as a function of stimulus display and mode of response. Technical Memorandum 4-70, Human Engineering Laboratories, U. S. Army Aberdeen Research & Development Center, Aberdeen Proving Ground, Md., 1970.

31. Hollier, J. & Evans, S. H. Schematic concept formation with linguaform patterns. Psychonomic Science, 1967, 9, 89-90.
32. Jones, E. C. & Holley, J. R. Schema utilization in pattern perception. Psychonomic Science, 1970, 18, 197-198.
33. Mavrides, C. M. & Brown, D. R. Schematic concept formation: Feature measures and constraint redundancy as predictors. Perception & Psychophysics, 1970, 7, 239-243.
34. Pollack, I. Discrimination of restrictions in sequentially blocked auditory displays: Shifting block designs. Perception & Psychophysics, 1971, 9, 335-338.
35. Rankin, W. C. & Evans, S. H. Facilitation of schematic concept formation as a function of two within-schema pretraining modes. Psychonomic Science, 1968, 13, 325-326.
36. Rankin, W. C., Markley, R. P., & Evans, S. H. Pythagorean distance and the judged similarity of schematic stimuli. Technical Memorandum 25-69, Human Engineering Laboratories, U. S. Army Aberdeen Research & Development Center, Aberdeen Proving Ground, Md., 1969.
37. Rosser, E. M. Categorization and discrimination of tone sequences. Unpublished doctoral dissertation, Harvard University, 1967.
38. Shannon, C. E. & Weaver, W. The mathematical theory of communication. Urbana, Illinois: The University of Illinois Press, 1949.
39. Tracy, J. F. & Evans, S. H. Supplementary information in schematic concept formation. Psychonomic Science, 1967, 9, 313-314.
40. Tracy, J. F. The effects of types of supplementary information, example, deviation, redundancy, and experience on categorization in a four category sorting task. Unpublished doctoral dissertation, Texas Christian University, 1971.
41. Wright, A. D. & Dixon, T. R. Knowledge of results in schematic concept formation. Professional Paper 17-68, The George Washington University, Human Resources Office, Alexandria, Virginia, 1968.
42. Zinser, O. A computer model of the initial stages of mammalian pattern processing. Unpublished doctoral dissertation, Texas Christian University, 1969.
43. Zinser, O. & Evans, S. H. A computer model of the initial stages of mammalian pattern processing. In Army Mathematics Steering Committee Proceedings of the 1970 Army Numerical Analysis Conference, ARO-D Report 71-1, U. S. Army Research Office, Durham, North Carolina, 1971.

SECTION II

INTRODUCTION TO VARGUS PATTERN GENERATION PROGRAMS

The purpose of this introduction to the documentation is to familiarize the reader with the overall content of the programs that follow and to discuss certain programming considerations not relevant to the description of the programs and thus not discussed in the documentation. The following pages contain the documentation for 21 programs and subroutines that have been used for the computer generation, measurement, manipulation and construction of the VARGUS 7, 9 and 10 patterns. These programs have been used on many occasions in the past, and the program listings in the documentation are exactly those programs which produced the sample outputs.

All programs are written in IBM 1800 Basic FORTRAN IV except program PAT which is written in AUTOCODER for the IBM 1401. In principle, the FORTRAN programs should be translated readily by FORTRAN compilers, regardless of the particular machine and implementation, especially since a USA Standard Basic FORTRAN IV is an accepted standard for the language. In fact, implementations of compilers and semantics of program statements frequently are different enough to prevent direct communication of programs between machines. We will now discuss some of the more common incompatibilities which the user may experience in attempting to utilize IBM 1800 Basic FORTRAN IV on other machines and briefly indicate how to correct them when they occur.

Probably the primary source of incompatibility a user will experience with these programs is the mixed mode arithmetic expressions. Standard FORTRAN does not permit mixed-mode expressions. The semantics of mixed expressions contained in these programs are as follows:

a. All terms in a mixed expression involving only floating-point variables are evaluated with floating-point arithmetic.

b. All terms in a mixed expression involving only integer-mode variables are evaluated with integer arithmetic.

c. When a mixed expression contains terms that involve both integer and floating-point variables, the integer quantities are converted to floating point before performing the floating-point arithmetic. Consider the mixed expression $A * B + I/J - C/(I + J)$. It is evaluated as follows:

a. $A * B$ is evaluated in floating point, and the product is a floating-point number.

b. I/J is an integer division so that any fractional remainder after division is lost, and the integer part of the quotient is saved as an integer-mode result.

c. In $C/(I + J)$, $(I + J)$ is evaluated with integer arithmetic, but before the division occurs the sum is converted to floating point, and the division is performed by floating-point arithmetic and yields a floating-point quotient.

d. The floating-point product of $A * B$ is then added to the quotient of I/J after the latter has been converted to floating point (the sum is a floating-point number), and finally,

e. The floating-point quotient of $C/(I + J)$ is then subtracted from the result of "d" to obtain a floating-point difference.

The user should note that an equivalent evaluation of this expression is not achieved by simply changing it to $A * B + \text{FLOAT}(I)/\text{FLOAT}(J) - C/(\text{FLOAT}(I) + \text{FLOAT}(J))$ since the middle quotient now may include a fractional part. When every term in an expression is either mixed or floating point (no integer terms) such a transformation will suffice. The numerical result of any mixed expression is always in floating-point mode.

Another source of incompatibility may result from the use of apostrophes to enclose literal data in Format statements. An equivalent representation may be achieved by using the H-format code.

Finally the DATA specification statement which initializes variables at compile time is not a USA Standard Basic FORTRAN IV feature. If this is a basis for an incompatibility, the initialization may be done with an input statement at execution time.

There are other ways in which IBM 1800 Basic FORTRAN IV differs from USA Standard Basic FORTRAN IV but they should not be the basis for incompatibilities. For example, 1800 FORTRAN IV variable names are limited to a length of five characters, whereas the more common length is six characters. Other incompatibilities of this innocuous sort are not discussed here.

For the readers' information, the logical unit numbers referenced in input-output statements in the program are associated with the following devices: logical unit 2 is the card reader-punch, logical unit 3 is the line printer, logical units 8 and 9 are tape drives. When no format number is associated with input-output from tape, all data transfer is in binary and involves no conversion.

One brief note should be made to users who may desire to convert these programs to FORTRAN II. In some programs, Type specification statements are used which alter the implicit mode of a variable, for example, INTEGER PATR permits PATR to be used as an integer mode variables. Since FORTRAN II does not implement Type statements, the desired mode may be obtained by adding an appropriate leading letter to the variable name, for example, KPATR. Also FORTRAN IV input-output statements which reference logical-unit numbers will have to be changed to reference Format statements only, and in output statements in particular the Format statement will have to be adjusted if no carriage control is needed. The apostrophes to define literal data will also have to be changed to an H-format code. Finally the DATA statement which initialize certain variables at compile time will have to be deleted and changed to initialization by means of an input statement.

An overview of the programs will illustrate their organization and interrelationships. In the Appendix, the term "Program Name" indicates the associated routine is a main program and the term "Subroutine Name" indicates the routine is a subroutine program. To the right of each routine's name is a brief statement of its purpose. When a routine's name is indented to the right relative to the routine name above it, the indented name is a subroutine called by the routine above it. It should be noted that there are subroutines below without an associated calling routine. Since the calling routines were relatively trivial (restricted largely to doing input and output) they were selectively excluded. Finally, sample executions are not included for each and every subroutine, since in many instances their function is minor and supporting a larger goal. Sample runs are included only for each composite of routines with a major objective.

APPENDIX

SUMMARY OF VARGUS PROGRAMS

Vargus 7

Program Name:	VARGUS 7E	Pattern generation.
Program Name:	ADHER	Pattern measurement.
Program Name:	PAT	Pattern construction.

Vargus 9

Program Name:	VARGUS 9	Pattern generation and measurement.
Program Name:	VARG 9	Pattern generation and measurement.
Program Name:	DVAR	Pattern measurement and construction.

Vargus 10

Program Name:	VARGUS 10A	Input submatrices and output pattern image.
Subroutine Name:	JOIN	Select submatrices.
Subroutine Name:	ROTAT	Rotate submatrices.
Subroutine Name:	MODUL	Perform modular arithmetic.

Subroutine Name:	CONG	Measure pattern similarity.
Subroutine Name:	BLUR	Blurs the input pattern.
Subroutine Name:	RSCAL	Rescale numbers in matrix.
Subroutine Name:	MIMAX	Find largest and smallest value in a matrix.

Subroutine Name: SOLID

Fill in figure defined
by a perimeter of ones.

Subroutine Name: WIDEN

Widen the lines of a
figure.

Subroutine Name: WIDER

Widen lines of a closed
figure inwardly.

Subroutine Name: SOLID

Subroutine Name: WIDEN

General Purpose Programs

Subroutine Name: RANDY

Pseudo-random number
generator.

Subroutine Name: PATOT

Output pattern matrix.

I. Identification

1. Program Name: VARGUS 7E
2. Category: Pattern Generation
3. Purpose: Generate strings of numbers containing
first-order sequential (Markov) dependency.
4. Date: September 8, 1970
5. Programmer: Selby Evans

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Elements of the Markov process are
the numbers 1 to 9.
Reuse of the same START card for the
random-number generator will produce
the same number sequence.

4. Input Data:

The first call to Subroutine RANDY results in the START card being read in. See RANDY documentation for details of its usage.

LNTH: The number of digits constituting a single pattern.

INTL: The identification number assigned to the first instance.

INST: The number of pattern instances desired.

LEAD: The first pattern element (digit from 1 to 9) in each pattern. If LEAD = 0, the first

pattern element is chosen at random
for each pattern.

NCØL: Number of elements in the Markov process.
Maximum allowable in this program is NCØL = 9.

P: An NCØL x NCØL Markov matrix. Each data-
card input to P is one row of P.

5. Output Data:

Final pattern is contained in array L and is output
on the printer.

6. Subroutines Required:

RANDY: A pseudo-random number generator.

III. Mathematical Methods and References

See Evans, S. H. "A model for perceptual category
formation." Unpublished doctoral dissertation, Texas
Christian University, 1964.

Evans, S. H. VARGUS 7: Computer patterns from Markov
Processes. Behavioral Science, 1967, 12, 323-328.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

// JOB                11000    30 JUN 71 13.404 HRS
// * U-2050-003
// FOR ABBA    30 JUN 71 13.405 HRS
*IOCS (CARD,1443 PRINTER,PLOTTER,DISK,MAGNETIC TAPE,KEYBOARD,TYPEWR
*NONPROCESS PROGRAM
*LIST ALL
*ONE WORD INTEGERS
C   VARGUS 7-E
      DIMENSION P(9,9),L(60)
      CALL RANDY(R)
      WRITE(3,901)
901  FORMAT('1',T7,'SAMPLE VARGUS SEVEN PATTERNS')
      WRITE(3,911)
911  FORMAT('0',T9,'ID',T21,'PATTERNS',/)
C
C   ENTER CONTROL CARDS AND MATRIX
      READ(2,4) LNTH,INTL,INST,LEAD,NCOL
4    FORMAT(5I5)
      INST = INST + INTL - 1
      DO 10 I=1,NCOL
10   READ(2,5) (P(I,J),J=1,NCOL)
5    FORMAT(9F8.4)
C
C   SET UP POPULATION LOOP
      X=NCOL
      X=1.0/X
      DO 90 NDO = INTL,INST
      DO 20 K=1,40
20   L(K)=0
C
C   CHOOSE STARTING ELEMENT
      MARK = LEAD
      IF(LEAD) 30,30,50
30   CALL RANDY(R)
      DO 40 MARK =1,NCOL
      R = R - X
      IF(R) 50,50,40
40   CONTINUE
      MARK= NCOL
C
C   SET UP INSTANCE LOOP
      DO 80 IDO =1,LNTH
      L(IDO)= MARK
C
C   MOVE ONE STEP
      CALL RANDY(R)
      DO 60 K= 1,NCOL
      R = R - P(MARK,K)
      IF(R) 70,60,60
60   CONTINUE
      K = NCOL
70   MARK = K
80   CONTINUE
C

```

```

C      PRINT RESULTING PATTERN AND ID ON PRINTER
      WRITE(3,9) NDD,(L(I),I=1,LNTH)
9      FORMAT(4X,I7,3X,12(1X,4I1))
90     CONTINUE
      CALL EXIT
      END

```

VARIABLE ALLOCATIONS

P(R)=00A0-0000	R(R)=00A2	X(R)=00A4	L(I)=00E1
INST(I)=00E4	LEAD(I)=00E5	NCOL(I)=00E6	I(I)=00E7
K(I)=00EA	MARK(I)=00EB	IDO(I)=00EC	

STATEMENT ALLOCATIONS

901	=00F5	911	=0108	4	=0115	.5	=0118	9	=0118	10	=0149	20
60	=01DD	70	=01EA	80	=01EE	90	=0210					

FEATURES SUPPORTED

```

NONPROCESS
ONE WORD INTEGERS
IOCS

```

CALLED SUBPROGRAMS

RANDY	FSUB	FSUBX	FDIV	FLD	FSTO	FLOAT	ISTOX	LDFAC	MR
SUBSC	FCHRI	TYPEN	HOLEB	MAGT	PRNTN	EBPRT	CARDN		

REAL CONSTANTS

```

.100000E 01=00EE

```

INTEGER CONSTANTS

3=00F0	2=00F1	1=00F2	40=00F3	0=00F4
--------	--------	--------	---------	--------

CORE REQUIREMENTS FOR ABBA

COMMON	0	INSKEL COMMON	0
VARIABLES	238	PROGRAM	302

END OF COMPILATION

SAMPLE VARGUS SEVEN PATTERNS

ID	PATTERNS
1001	1731 1725 3153 45
1002	7263 1727 2664 53
1003	4531 7726 7645 37
1004	5353 1726 4531 74
1005	3517 2645 3453 17
1006	7264 1717 2617 26
1007	7253 1726 4531 76
1008	2647 2645 1726 45
1009	1726 4574 5726 45
1010	5317 2641 7131 56

I. Identification

1. Program Name: ADHER
2. Category: Pattern Measurement.
3. Purpose: The program measures the degree to which a pattern generated by program VARGUS 7E adheres to the schema rule from which it was generated. See Section III below.
4. Date: May 17, 1971
5. Programmer: Mike Abbamonte

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: As written, the program will measure patterns generated from a seven-element Markov Process. Adjustment in the size of arrays MPS and MX will make the program more general.
4. Input Data:
 - NCØL: The length (number of digits in) an individual pattern.
 - LMPS: The length of the most probable sequence before cycling. Should be seven here.
 - NPAT: The number of patterns to be measured.
 - MPS: A vector containing the schema: i.e., the most probable sequence.

IDPAT: Individual pattern identification number.

IN: The array containing the pattern.

5. Output Data:

IDPAT: Same as above.

IN: Same as above.

PSS: The proportion of schematic steps. See
Section III for details.

6. Subroutines Required:

None.

III. Mathematical Method and References

The proportion of schematic steps (PSS) is the measure used to indicate the degree to which a pattern adheres to its generation rule (also called the MPS, i.e., most probable sequence). PSS is simply that fraction of the number of elements in a pattern which are followed by the most probable successor element.

See also, Evans, S. H. VARGUS 7: Computer patterns from Markov processes. Behavioral Science, 1967, 12, 323-328.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

// JOB    30 JUN 71 13.438 HRS
// * U-2050-003
// FOR ABBA    30 JUN 71 13.438 HRS
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*IOCS (CARD,1443 PRINTER,PLOTTER,DISK,MAGNETIC TAPE,KEYBOARD,TYPEWRITER)
*LIST ALL
C  ADHER
      DIMENSION IN(24),MPS(7),MX(7,7)
      READ(2,2) NCOL, LMPS,NPAT
2     FORMAT(3I5)
      FCOL=NCOL
      READ(2,3) (MPS(I),I=1,LMPS)
3     FORMAT(20I3)
      WRITE(3,88)
      88 FORMAT('1',T3,'ID',T28,'VARGUS SEVEN PATTERNS',T60,'PROPORTION OF
      ISCHEMATIC STEPS',/)
      DO 500 I=1,NPAT
      READ(2,5) IDPAT, (IN(J),J=1,NCOL)
5     FORMAT(I5,5X,20(I1,1X))
      DO 200 K1=1,LMPS
      DO 200 K2=1,LMPS
200    MX(K1,K2)=0
      DO 100 L=1,NCOL
      IR=IN(L)
      IF(L-NCOL) 60,70,60
60     IC=IN(L+1)
      GO TO 100
70     IR=IN(1)
100    MX(IR,IC)=MX(IR,IC)+1
      ICNT=0
      DO 150 L=1,LMPS
      IR=MPS(L)
      IF(L-LMPS) 120,130,120
130    IC=MPS(1)
      GO TO 150
120    IC=MPS(L+1)
150    ICNT=ICNT+MX(IR,IC)
      FCNT=ICNT
      PSS=FCNT/FCOL
      WRITE(3,30) IDPAT, (IN(L),L=1,NCOL), PSS
      30 FORMAT(I5,10X,14I3,12X,F8.4)
500   CONTINUE
      CALL EXIT
      END

```

VARIABLE ALLOCATIONS

FCOL(R)=0000	FCNT(R)=0002	PSS(R)=0004	IN(I)=001D
NCOL(I)=0056	LMPS(I)=0057	NPAT(I)=0058	I(I)=0059
K1(I)=005C	K2(I)=005D	L(I)=005E	IR(I)=005F

STATEMENT ALLOCATIONS

2	=0068	3	=0068	88	=006E	5	=0093	30	=009A	200	=00F0	60
120	=0166	150	=016F	500	=01AB							

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS

FDIV	FLD	FSTO	FLOAT	ISTOX	MRED	MWRT	MCOMP	MIOIX	MI
MAGT	PRNTN	EBPRT	CARDN						

INTEGER CONSTANTS

2=0064	1=0065	3=0066	0=0067
--------	--------	--------	--------

CORE REQUIREMENTS FOR ABBA

COMMON	0	INSKEL COMMON	0
VARIABLES	100	PROGRAM	338

END OF COMPILATION

ID	VARGUS SEVEN PATTERNS													PROPORTION OF SCHEMATIC STEPS												
1001	1	7	3	1	1	7	2	5	3	3	1	5	3	4	5											0.5714
1002	7	2	6	3	1	7	2	7	2	6	6	6	4	5	3											0.7142
1003	4	5	3	1	7	7	2	6	7	6	6	4	5	3	7											0.6428
1004	5	3	5	3	1	7	2	6	4	5	3	3	1	7	4											0.7857
1005	3	5	1	7	2	6	4	5	3	4	5	5	3	1	7											0.7142
1006	7	2	6	4	1	7	1	7	2	6	1	7	7	2	6											0.7142
1007	7	2	5	3	1	7	2	6	4	5	3	3	1	7	6											0.7857
1008	2	6	4	7	2	6	4	5	1	7	2	2	6	4	5											0.7857
1009	1	7	2	5	4	5	7	4	5	7	7	2	6	4	5											0.7142
1010	5	3	1	7	2	6	4	1	7	1	3	3	1	5	6											0.5714

I. Identification

1. Program Name: PAT
2. Category: Pattern Construction
3. Purpose: Punch histograms of VARGUS 7 patterns
on cards by utilizing the punch-column
binary feature of the IBM 1401.
4. Date: September 26, 1968
5. Programmer: Phillip R. Jones

II. Use Information

1. Language: IBM 1401 Autocoder Ver. 1, mod. 13.
2. Machine: IBM 1401, 1402, and 1407 with punch-
column binary feature.
3. Limitations: Punches maximum column height of 7.
4. Input Card: Columns 3-7 identification number.
Columns 14-67 digits specifying
pattern column heights.
5. Output: Columns 1-72 VARGUS 7 histogram.
Columns 76-80 identification number.
6. Operating Instructions: Sense switch "B" off
generates one column per
digit; "B" on generates
two columns per digit.
7. Subroutines Required: None.
8. Error Checks: ID - checks card columns 3-7 for
blank.
Input - checks for invalid characters.

III. Mathematical Method and References

None.

IV. Listing of Source Program

Attached.

V. Flow Chart

Attached.

VI. Sample Run

Not attached.

VII. Remarks

Testing has been extensive and all attempts to punch
incorrect patterns have failed.

CROSS REFERENCE LISTING

ADDRS	LABEL	TAG	SEQUENCE NUMBERS OF INSTRUCTIONS WHERE SYMBOL APPEARS										
00618	BEGIN		0007	0034	0078	0105	0110						
01245	EOJ		0111	0107									
01298	GM		0113	0035	0039	0092	0095	0097	0137	0138			
			0146	0148	0149								
00660	P0112		0017	0022	0057								
00698	P0117		0023	0018									
00790	P0209		0035	0023									
00801	P0211		0037	0024									
00812	P0213		0039	0025									
00823	P0215		0041	0025									
00834	P0217		0043	0027									
00845	P0219		0045	0028									
00856	P0301		0047	0029									
00867	P0303		0049	0030									
00878	P0305		0051	0031									
00889	P0307		0053	0032									
00896	P0308		0054	0035	0038	0040	0042	0044	0046	0048			
00936	P0314		0060	0057	0102	0104	0155	0158					
00987	P0401		0069	0052									
01063	P0412		0080	0059									
01081	P0415		0083	0070									
01092	P0417		0085	0071									
01103	P0419		0087	0072									
01114	P0501		0089	0073									
01132	P0504		0092	0074									
01150	P0507		0095	0075									
01161	P0509		0097	0075									
01175	P0511		0099	0082	0084	0085	0088	0091	0094	0096			
01213	P0517		0105	0055	0157								
01218	P0519		0107	0007									
01233	P0601		0109	0021	0123	0154							
01299	P0701		0115	0051									
01367	P0711		0125	0115									
01385	P0714		0128	0115									
01403	P0717		0131	0117									
01421	P0720		0134	0118									
01439	P0803		0137	0119									
01457	P0806		0140	0120									
01489	P0811		0145	0121									
01507	P0814		0148	0122									
01535	P0818		0152	0127	0130	0133	0135	0139	0144	0147			
00600	START		0004	0159									
00089	X1		IIII	0013	0019	0055	0058	0066	0099	0152			
00094	X2		IIII	0014	0020	0055	0059	0100	0153				

SEQ	PGLIN	LABEL	OPCD	OPERAND	A/C V1-M13
0001	01020	PT219	JOB	COLUMN BINARY PATTERN BUILDER VER 2 MOD 1	
0002	01023		CTL	4111 S	
0003	01025		ORG	600	
0004	01030	START	SW	1,87	
0005	01040		SW	92	
0006	01050		MLCWA	GM,81	
0007	01060	BEGIN	BLC	P0519	TEST LAST CARD
0008	01061		CW	201	
0009	01070		R		READ A CARD
0010	01071		BIN	**5,B	
0011	01072		B	**5	
0012	01073		SW	201	
0013	01080		SBR	X1,3	
0014	01090		SBR	X2,76	
0015	01100		CS	599	CLEAR BINARY AREA
0016	01110		CS		
0017	01120	P0112	BCE	**5,0+X1,	TEST FOR BLANKS
0018	01125		B	P0117	BRANCH ON GOOD ID
0019	01130		SBR	X1,1+X1	
0020	01140		SBR	X2,1+X2	
0021	01150		BCE	P0601,089,8	TEST FOR ID LENGTH ERROR
0022	01160		B	P0112	
0023	01170	P0117	BCE	P0209,0+X1,0	TEST FOR ZERO
0024	01180		BCE	P0211,0+X1,1	ONE
0025	01190		BCE	P0213,0+X1,2	TWO
0026	01200		BCE	P0215,0+X1,3	THREE
0027	02010		BCE	P0217,0+X1,4	FOUR
0028	02020		BCE	P0219,0+X1,5	FIVE
0029	02030		BCE	P0301,0+X1,6	SIX
0030	02040		BCE	P0303,0+X1,7	SEVEN
0031	02050		BCE	P0305,0+X1,8	EIGHT
0032	02060		BCE	P0307,0+X1,9	NINE
0033	02070		WCP	1	
0034	02080		B	BEGIN	
0035	02090	P0209	MLNS	'8',400+X2	BUILD COLUMN BINARY ZERO
0036	02100		B	P0308	
0037	02110	P0211	MLNS	'4',400+X2	ONE
0038	02120		B	P0308	
0039	02130	P0213	MLNS	'2',400+X2	TWO
0040	02140		B	P0308	
0041	02150	P0215	MLNS	'1',400+X2	THREE
0042	02160		B	P0308	
0043	02170	P0217	MLZS	'J',500+X2	FOUR
0044	02180		B	P0308	
0045	02190	P0219	MLZS	'Z',500+X2	FIVE
0046	02200		B	P0308	
0047	03010	P0301	MLNS	'8',500+X2	SIX
0048	03020		B	P0308	
0049	03030	P0303	MLNS	'4',500+X2	SEVEN
0050	03040		B	P0308	
0051	03050	P0305	MLNS	'2',500+X2	EIGHT
0052	03060		B	P0308	

SEQ	PGLIN	LABEL	OPCD	OPERAND	A/C V1-M13
0053	03070	P0307	MLNS	'1',500+X2	NINE
0054	03080	P0308	BCE	*+19,89,7	
0055	03081		SBR	X1,1+X1	
0056	03082		SBR	X2,1+X2	
0057	03090		B	P0112	
0058	03120		SBR	X1,14	REINITIALIZE X1=014
0059	03130		SBR	X2,1	X2=001
0060	03140	P0314	BCE	*+13,0+X1,	TEST FOR BLANK DATA COLUMN
0061	03145		BW	P0701,201	TEST FOR DOJBLE COLUMN
0062	03147		B	P0401	
0063	03150		BCE	*+5,88,6	TEST FOR LAST COLUMN
0064	03160		B	*+9	
0065	03170		BCE	P0517,89,7	
0066	03180		SBR	X1,1+X1	INCREMENT NEXT COLUMN
0067	03190		B	P0314	
0068	03200	*			COLUMN BINARY PATTERN BUILD
0069	04010	P0401	BCE	P0412,0+X1,0	TEST FOR ZERO
0070	04020		BCE	P0415,0+X1,1	ONE
0071	04030		BCE	P0417,0+X1,2	TWO
0072	04040		BCE	P0419,0+X1,3	THREE
0073	04050		BCE	P0501,0+X1,4	FOUR
0074	04060		BCE	P0504,0+X1,5	FIVE
0075	04070		BCE	P0507,0+X1,6	SIX
0076	04080		BCE	P0509,0+X1,7	SEVEN
0077	04090		WCP	1	PATTERN ERROR
0078	04100		B	BEGIN	
0079	04110	*			BUILD BINARY COLUMN - SINGLE
0080	04120	P0412	MLC	' ',400+X2	ZERO
0081	04130		MLC	' ',500+X2	
0082	04140		B	P0511	
0083	04150	P0415	MLNS	'1',500+X2	ONE (1-BIT)
0084	04160		B	P0511	
0085	04170	P0417	MLNS	'3',500+X2	TWO (1,2-BITS)
0086	04180		B	P0511	
0087	04190	P0419	MLNS	'7',500+X2	THREE(1,2,4-BITS)
0088	04200		B	P0511	
0089	05010	P0501	MLC	GM,500+X2	FOUR (1,2,4,8-BITS)
0090	05020		MLZS	' ',500+X2	
0091	05030		B	P0511	
0092	05040	P0504	MLC	GM,500+X2	FIVE (1,2,4,8,A-BITS)
0093	05050		MLZS	'Z',500+X2	
0094	05060		B	P0511	
0095	05070	P0507	MLC	GM,500+X2	SIX (1,2,4,8,A,B-BITS)
0096	05080		B	P0511	
0097	05090	P0509	MLC	GM,500+X2	SEVEN(1,2,4,8,A,B,1-BITS)
0098	05100		MLNS	'1',400+X2	
0099	05110	P0511	SBR	X1,1+X1	INCREMENT TO NEXT COLUMN
0100	05120		SBR	X2,1+X2	
0101	05130		BCE	*+5,88,6	TEST FOR END OF PATTERN
0102	05140		B	P0314	
0103	05150		BCE	*+5,89,8	
0104	05160		B	P0314	

SEQ	PGLIN	LABEL	OPCD	OPERAND	A/C V1-M13
0105	05170	P0517	PCB	BEGIN	PUNCH COLUMN BINARY
0106	05180	*			EOJ HALT
0107	05190	P0519	WCP	EOJ	
0108	05200		H	999,999	
0109	06010	P0601	WCP	1	DATA CARD ERROR
0110	06020		B	BEGIN	
0111	06030	EOJ	DCW	'END OF JOB -- REMOVE ANY CORRECTIONS TYPED ON'	
0112	06040			' CONSOLE'	
0113	06050	GM	DCW	' '	
0114	07000	*			DOUBLED ROW OPTION
0115	07010	P0701	BCE	P0711,0+X1,0	TEST FOR ZERO
0116	07020		BCE	P0714,0+X1,1	ONE
0117	07030		BCE	P0717,0+X1,2	TWO
0118	07040		BCE	P0720,0+X1,3	THREE
0119	07050		BCE	P0803,0+X1,4	FOUR
0120	07060		BCE	P0806,0+X1,5	FIVE
0121	07070		BCE	P0811,0+X1,6	SIX
0122	07080		BCE	P0814,0+X1,7	SEVEN
0123	07090		B	P0601	DATA ERROR
0124	07100	*			BUILD DOUBLED BINARY COLUMN
0125	07110	P0711	MLC	' ',401+X2	ZERO (NO BITS)
0126	07120		MLC	' ',501+X2	
0127	07130		B	P0818	
0128	07140	P0714	MLNS	'1',500+X2	ONE (1-BIT)
0129	07150		MLNS	'1',501+X2	
0130	07160		B	P0818	
0131	07170	P0717	MLNS	'3',500+X2	TWO (1,2-BITS)
0132	07180		MLNS	'3',501+X2	
0133	07190		B	P0818	
0134	07200	P0720	MLNS	'7',500+X2	THREE (1,2,4-BITS)
0135	08010		MLNS	'7',501+X2	
0136	08020		B	P0818	
0137	08030	P0803	MLNS	GM,500+X2	FOUR (1,2,4,8-BITS)
0138	08040		MLNS	GM,501+X2	
0139	08050		B	P0818	
0140	08060	P0806	MLNS	GM,500+X2	FIVE (1,2,4,8,A-BITS)
0141	08070		MLZS	'Z',500+X2	
0142	08080		MLNS	GM,501+X2	
0143	08090		MLZS	'Z',501+X2	
0144	08100		B	P0818	
0145	08110	P0811	MLC	GM,500+X2	SIX (1,2,4,8,A,8-BITS)
0146	08120		MLC	GM,501+X2	
0147	08130		B	P0818	
0148	08140	P0814	MLC	GM,500+X2	SEVEN (1,2,4,8,A,8,1-BITS)
0149	08150		MLC	GM,501+X2	
0150	08160		MLNS	'1',400+X2	
0151	08170		MLNS	'1',401+X2	
0152	08180	P0818	SBR	X1,1+X1	
0153	08190		SBR	X2,2+X2	
0154	08200		BCE	P0601,93,8	
0155	08210		BCE	*+5,88,6	
0156	08220		B	P0314	

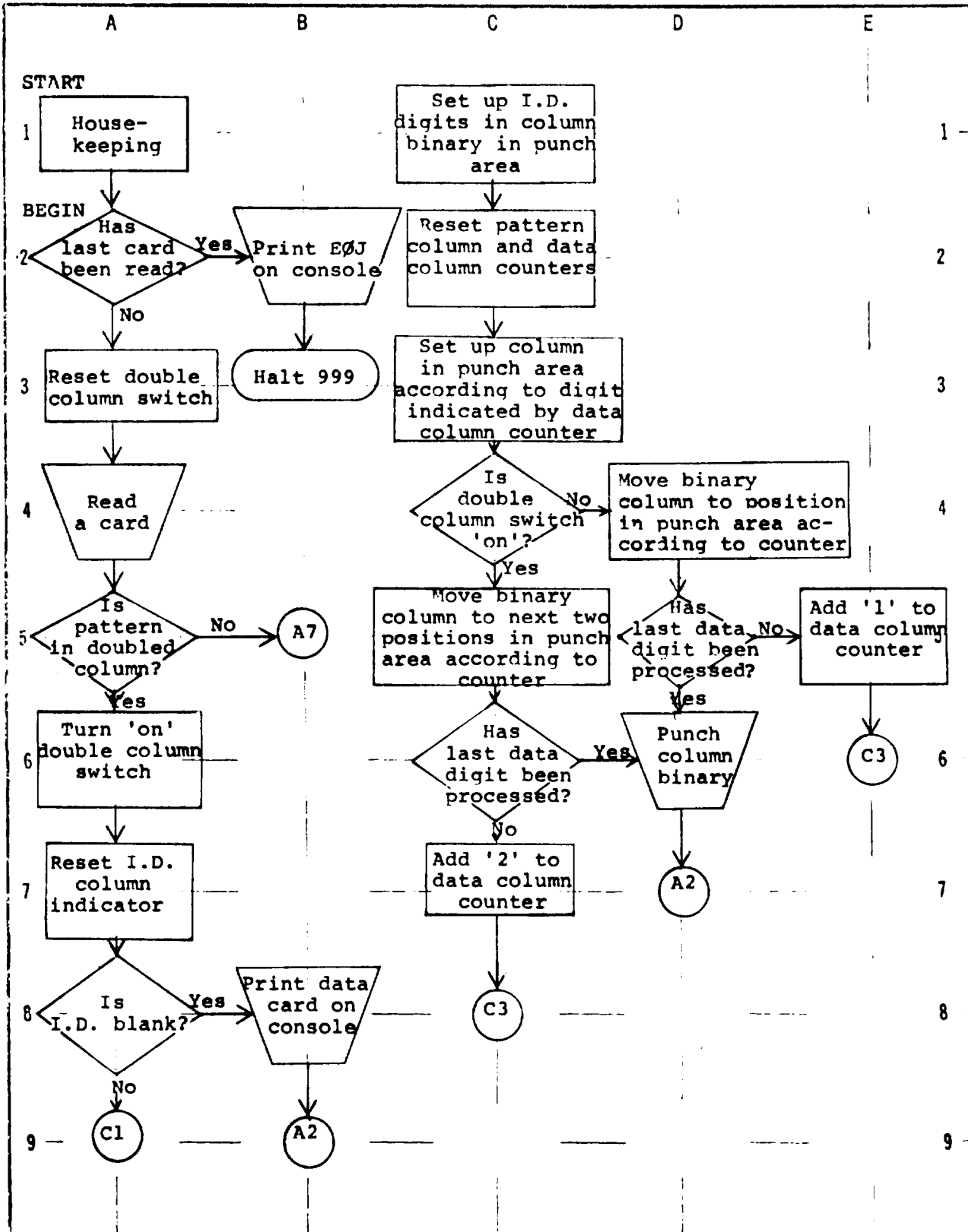
SEQ	PGLIN LABEL	OPCD	OPERAND	A/C V1-M13
0157	08230	BCE	P0517,89,8	
0158	09010	B	P0314	
0159	09020	END	START	
0160		LTRL	'8'	
0161		LTRL	'4'	
0162		LTRL	'2'	
0163		LTRL	'1'	
0164		LTRL	'J'	
0165		LTRL	'Z'	
0166		LTRL	' '	
0167		LTRL	'3'	
0168		LTRL	'7'	
0169		LTRL	' '	

END OF LISTING

THEO CHRISTIAN UNIVERSITY
COMPUTER CENTER

DATE September 26 PROGRAM PAT
ROUTINE _____

PAGE 1 of 1



I. Identification

1. Program Name: VARGUS 9
2. Category: Pattern Generation
3. Purpose: To compute patterns by producing
 probabilistic variations on a prototype.
4. Date: January, 1966
5. Programmer: Selby Evans

II. Use Information

1. Language: 1800 Basic FØRTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Up to 32 columns in a pattern.
4. Input data:

The initial call to RANDY causes a START card to be read by the subroutine. See RANDY documentation for details.

NCOL: Number of columns in a pattern, NCØL less than or equal to 32.

NSCHM: The schema identification number.
Typically it is 3 or 4 digits ending in 00.

NINST: The number of patterns to be generated.
NINST less than or equal to 99.

MAX: A vector the first NCØL elements of which form the schema.

P: The probability matrix the I-th column of which controls the frequency distribution (dispersion) of the I-th elements of the patterns about the I-th schema element. NCØL such probability vectors are read in; i.e., one for each element of the schema.

5. Output Data:

NPAT: The individual pattern identification number.

INST: The pattern itself.

PR: The cumulative product of the probability of each actual selection in constructing a pattern.

FMS: The mean squared deviation of each pattern from its schema (elsewhere called pattern variance).

6. Subroutines Required:

RANDY: A pseudo-random number generator.

III. Mathematical Method and References

Evans, S. H., and Mueller, M. VARGUS 9: Computed patterns for schema research. Psychonomic Science, 1966, 12, 511-512.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

// JOB                11000    30 JUN 71 13.369 HRS
// * U-2050-003
// FOR ABBA    30 JUN 71 13.370 HRS
*NONPROCESS PROGRAM
*IOCS (CARD,1443 PRINTER,PLOTTER,DISK,MAGNETIC TAPE,KEYBOARD,TYPEWRITER)
*ONE WORD INTEGERS
*LIST ALL
C    VARGUS 9
    DIMENSION P(9,32),MAX(32),INST(32)
    CALL RANDY(R)
    WRITE(3,901)
  901 FORMAT('1',T19,'SAMPLE VARGUS NINE PATTERNS')
    WRITE(3,911)
  911 FORMAT('0',T6,'ID',T26,'PATTERN',T48,'PROB.',T55,'PATTERN VARIANCE
    1',/)
C
C    READ PARAMETERS
    READ(2,1) NCOL,NSCHM,NINST
  1    FORMAT(5I5)
    DO 101 IN=1,32
      MAX(IN)=0
  101  INST(IN)=0
C
C    READ PROBABILITY MATRIX
    DO 55 J=1,NCOL
  55  READ(2,2) MAX(J),(P(I,J),I=1,9)
  2    FORMAT(15,3X,9F8.4)
C
    FNCOL =NCOL
C
C    OUTPUT SCHEMA
    WRITE(3,5) NSCHM,(MAX(I),I=1,NCOL)
  5    FORMAT(3X,15,8X,32I2)
C
C    SET UP INSTANCE LOOP
    DO 99 IN=1,NINST
      FMS=0
      PR=1.0
      SS=0
C
C    SET UP COLUMN LOOP
    DO 89 ICOL=1, NCOL
      CALL RANDY(R)
C    CHOOSE ELEMENT
      DO 79 L=1,9
        R = R - P(L,ICOL)
        IF(R)80,79,79
  79    CONTINUE
        L=9
  80    CONTINUE
        PR=PR* P(L,ICOL)
        T= MAX(ICOL)-L
        SS = SS + T*T
  89    INST(ICOL)=L

```

```

C      END INSTANCE LOOP
      NPAT= NSCH!! + IN
      FMS = SS/F*COL
      WRITE(3,15) NPAT,(INST(I),I=1,NCOL),PR,FMS
15     FORMAT(3X,15,8X,14I2,2F8.4)
99     CONTINUE
      CALL EXIT
      END

```

VARIABLE ALLOCATIONS

P(R)=023E-0000	R(R)=0240	FNCOL(R)=0242	FMS(R)=0244
T(R)=024A	MAX(I)=026B-024G	INST(I)=025B-026C	NCOL(I)=028C
IN(I)=028F	J(I)=0290	I(I)=0291	ICOL(I)=0292

STATEMENT ALLOCATIONS

901	=02A0	911	=02B3	1	=02CF	2	=02D2	5	=02D7	15	=02DD	10
89	=03BE	99	=03F6									

FEATURES SUPPORTED

NONPROCESS
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS

RANDY	FADD	FSUBX	FMPY	FMPYX	FDIV	FLD	FSTO	FLOAT	IS
MIOIX	MIOF	MIOI	SUBSC	FCHRI	TYPEN	HOLEB	MAGT	PRNTN	EBI

REAL CONSTANTS

.100000E 01=0298

INTEGER CONSTANTS

3=029A	2=029B	1=029C	32=029D	0=029E	9=029F
--------	--------	--------	---------	--------	--------

CORE REQUIREMENTS FOR ABBA

COMMON	0	INSKEL COMMON	0
VARIABLES	664	PROGRAM	362

END OF COMPILATION'

SAMPLE VARGUS NINE PATTERNS

ID	PATTERN	PROB.	PATTERN VARIANCE
2000	4 3 5 7 4 6 5 4 5 6 5 5 7 5		
2001	4 3 5 7 4 6 5 4 5 6 5 5 5 3	0.0000	0.5714
2002	4 3 5 7 2 6 5 4 5 7 5 5 7 5	0.0001	0.3571
2003	2 3 5 7 4 6 5 3 5 6 5 4 7 5	0.0000	0.4285
2004	4 3 5 7 4 6 5 4 5 6 6 5 7 5	0.0055	0.0714
2005	4 3 6 8 4 6 5 4 5 6 5 5 9 3	0.0000	0.7142
2006	4 3 5 7 4 6 5 4 5 6 5 5 7 5	0.0870	0.0000
2007	4 3 5 7 3 6 5 4 5 6 5 5 7 5	0.0055	0.0714
2008	4 3 5 7 5 6 4 4 5 6 5 5 7 5	0.0003	0.1428
2009	4 3 4 7 6 6 5 4 6 6 5 5 7 5	0.0000	0.4285
2010	4 3 5 7 4 6 5 4 5 6 5 5 7 4	0.0055	0.0714

I. Identification

1. Program Name: VARG 9
2. Category: Pattern Generation
3. Purpose: Compute VARGUS 9 patterns and print dots that can be connected to form a serriform VARGUS 9 pattern.
4. Date: April, 1970
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Maximum of 32 element patterns.
4. Input Data:

First call to subroutine RANDY causes START card to be read. See RANDY documentation for details.

IMØ: Month.

IDS: Day.

IYR: Year.

USER: User information; e.g., name.

NCØL: Number of digits per pattern.

NSHEM: Identification number of schema.

Should be 3 digits at most.

NINST: The number of patterns desired.

MAX: A vector the first NCØL elements of which contain the schema.

PROB: A probability matrix the I-th column of which contains probabilities which control the frequency distribution of values of the I-th element in the patterns produced. For example if MAX(I) is a 5, then the column PROB (1,I) to PROB (32,I) might be 0,0,0.1, 0.1, 0.6, 0.1, 0.1, 0,0,0, etc.

5. Output Data:

ARY: A matrix containing alphameric blanks and points. The points may be connected to form a serriform pattern.

NPAT: Pattern identification number.

INST: The vector of digits which constitute the basic pattern.

PR: A cumulative product of probabilities of the elements actually chosen.

FMS: The mean square deviations of the pattern from its prototype. Also called pattern variance.

All output to logical unit 8 is to produce a tape (in binary) utilized by program DVAR.

6. Subroutines Required:

RANDY: A pseudo-random number generator.

III. Mathematical Methods and References

Evans, S. H. and Mueller, M. VARGUS 9: Computed stimuli for schema research. Psychonomic Science, 1966, 12, 511-512.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

// JOB                11000    30 JUN 71 13.512 HRS
// * U-2050-003
// FOR VARG          30 JUN 71 13.513 HRS
*LIST ALL
*IOCS (CARD,1443 PRINTER, DISK,PLOTTER,MAGNETIC TAPE)
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
C  VARG 9
    DIMENSION PROB (9,32), MAX(32), INST(32), USER(3)
    DIMENSION ARY(20,32)
    DATA POINT, BLANK /'.', ' '/
C
    RR = 1.
    CALL RANDY(RR)
    1 FORMAT (3I5)
    2 FORMAT (I5,3X ,9F8.5)
    3 FORMAT (3I3,3A4)
    4 FORMAT (9H1VARGUS 9, 3I3, 3A4, 4X, 6HSCHEMA, 15, 1H,,15, 5H INST)
    5 FORMAT (6H0 VR9 ,      14, 1H, 8(1H ,4I1))
    6 FORMAT (6H0 VR9 ,      14, 1H, 8(1H ,4I1)/ E14.8, F12.5)
    7 FORMAT (1H1)
    8 FORMAT (50X, 14A3//)
    9 FORMAT (/////////)
    REWIND 8
C
C  READ DATA AND USER
8000 READ (2,3) IMD, IDA, IYR, (USER(I), I=1,3)
C  READ PARAMETERS
    READ (2,1) NCOL, NSHEM, NINST
    DO 101 IN=1,32
    MAX(IN)=0
101 INST(IN)=0
C
C  READ PROBABILITY MATRIX
    READ (2,2) (MAX(J), (PROB(I,J), I=1,9), J=1,NCOL)
C
    FNCOL=NCOL
C  OUTPUT IDENTIFICATION
    WRITE (3,4) IMD, IDA, IYR, (USER(I), I=1,3), NSHEM, NINST
    WRITE (8) IMD, IDA, IYR, USER
    WRITE (8) NCOL, NSHEM, NINST
C  OUTPUT SCHEMA
    WRITE (3,5) NSHEM, (MAX(I), I=1,NCOL)
    WRITE (8) NSHEM, (MAX(I), I=1,NCOL)
    DO 10 I=1,20
    DO 10 J=1,32
10 ARY(I,J)=BLANK
    IP=10
    ARY(IP,1)=POINT
    ARY(IP,NCOL)=POINT
    DO 15 I=1,NCOL
    II=10-MAX(I)
15 ARY(II,I)=POINT
    WRITE (3,9)

```

```

        WRITE (3,8) ((ARY(I,J), J=1,14), I=1,IP)
        WRITE (3,7)
C      SET INSTANCE LOOP
        DO 99 IN=1,NINST
          PR=1.0
          SS=0.0
          DO 20 I=1,20
            DO 20 J=1,32
              20 ARY(I,J)=BLANK
                ARY(IP,NCOL)=POINT
                ARY(IP,1)=POINT
C
C      SET COLUMN LOOP
          DO 89 ICOL=1,NCOL
            CALL RANDY(RR)
            R = RR
C      CHOSE ELEMENT
            DO 79 L=1,9
              R=R-PROB(L,ICOL)
              IF(R)80,79,79
              79 CONTINUE
                L=9
              80 CONTINUE
                PR=PR*PROB(L,ICOL)
                T=MAX(ICOL)-L
                SS=SS+(T*T)
              89 INST(ICOL)=L
                NPAT = NSHEM+IN
                FMS=SS/FNCOL
                WRITE (3,6) NPAT, (INST(I), I=1,32 ), PR, FMS
                WRITE (8) NPAT, (INST(I), I=1,NCOL), PR, FMS
                DO 30 J=1,NCOL
                  II=10-INST(J)
                  30 ARY(II,J)=POINT
                WRITE (3,9)
                WRITE (3,8) ((ARY(I,J), J=1,14), I=1,IP)
                WRITE (3,7)
              99 CONTINUE
C      END INSTANCE LOOP
        CALL EXIT
        END

```

VARIABLE ALLOCATIONS

PROB(R)=023E-0000	USER(R)=0244-0240	ARY(R)=0744-0246	RR(R)=0746
POINT(R)=074C	PR(R)=074E	SS(R)=0750	R(R)=0752
MAX(I)=0777-0758	INST(I)=0797-0778	IMO(I)=0798	IDA(I)=0799
NCOL(I)=079C	NSHEM(I)=079D	NINST(I)=079E	IN(I)=079F
II(I)=07A2	ICOL(I)=07A3	L(I)=07A4	NPAT(I)=07A5

UNREFERENCED STATEMENTS

8000

STATEMENT ALLOCATIONS

1	=07B6	2	=07B9	3	=07BE	4	=07C3	5	=07DB	6	=07E9	7
101	=0848	10	=08F7	15	=093B	20	=0990	79	=09E2	80	=09EF	89

FEATURES SUPPORTED

NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS

RANDY	FADD	FSUBX	FMPY	FMPYX	FDIV	FLD	FSTD	FSTOX	FLI
MWRT	MCOMP	MIOFX	MIOIX	MIOF	MIOI	SUBSC	REWND	UIOAF	UII
PRNTN	EBPRT	CARDN							

REAL CONSTANTS

.100000E 01=07A8 .000000E 00=07AA

INTEGER CONSTANTS

8=07AC 2=07AD 1=07AE 3=07AF 32=0780 0=07B1

CORE REQUIREMENTS FOR VARG

COMMON 0 INSKEL COMMON 0
VARIABLES 1960 PROGRAM 792

END OF COMPILATION

VARGUS 9 5 26 71 I.S.C.S.

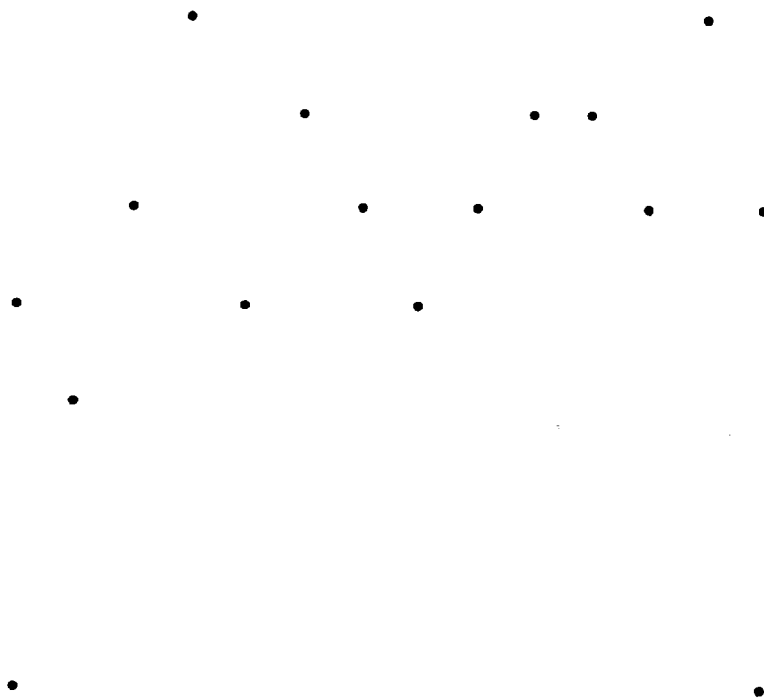
SCHEMA 2000, 30 INST

VR9 2000, 4357 4654 5655 75



VR9 2004, 4357 4654 5665 7500 0000 0000 0000 0000

.55253123E-02 0.07142



I. Identification

1. Program Name: DVAR
2. Category: Pattern Measurement and Description
3. Purpose: Finds Pythagorean distances of a set of patterns from a given pattern.
4. Date: April, 1970
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: The distance of each pattern in a set may be found from up to 4 defined patterns.

4. Input data:

Tape produced by program VARG 9.

TITLE: Distance values against which to compare computed distances and thereby form a histogram.

NPRØT: Number of schema patterns: up to four.

5. Output data:

NPAT: Pattern identification number.

DIST: Distance of a pattern from the set from NPRØT given patterns.

AVER: Average pattern variance; i.e., average of FMS values.

ARY: Histogram of dots.

TITLE: Abscissa of the histogram.

IFREQ: A frequency count of the dots printed
over each abscissa value.

NØS: Histogram corresponding to ARY but
constructed from actual pattern numbers
not dots.

6. Subroutines Required:

None.

III. Mathematical Methods and References

Evans, S. H. and Mueller, M. VARGUS 9: Computed stimuli
for schema research. Psychonomic Science, 1966, 12,
511-512.

Rankin, W. C., Markley, R. P., and Evans, S. H.
Pythagorean distance and the judged similarity of
schematic stimuli. Perception and Psychophysics,
1970, 7, 103-107.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

// JOB    30 JUN 71 13.650 HRS
// * U-2050-003
// FOR DVAR    30 JUN 71 13.650 HRS
*LIST ALL
*IOCS (CARD,1443 PRINTER, DISK,PLOTTER,MAGNETIC TAPE)
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
C  DVAR
    DIMENSION USER(3), MAX(32), INST(32), ARY(24,16), IFREQ(24),
    1 TITLE(24), NOS(24,16), NPONT(24), NPROT(14,4), DIST(4)
    DATA POINT, BLANK /' ', ' ' /
    1 FORMAT(9H1VARGUS 9, 3I3,3A4,4X,6HSCHEMA,15,1H,,15,5H INST)
    2 FORMAT(13H AVERAGE PV = ,F12.5)
    3 FORMAT(24A5)
    4 FORMAT(1H0,14,23I5)
    5 FORMAT(1X,F4.2,23F5.2//)
    6 FORMAT(8F10.5)
    7  FORMAT(5X,4(10X,14I1))
    8  FORMAT(15,4(10X,F14.5))
    9  FORMAT(4(5X,14I1))
    10 FORMAT(15,4F15.5)
    REWIND 8
C
    READ(2,6) TITLE
    READ(2,9) NPROT
    READ(8) IMO,IDA,IYR,USER
    READ(8) NCOL,NSHEM,NINST
    WRITE(3,1) IMO,IDA,IYR,(USER(I),I=1,3),NSHEM,NINST
    WRITE(3,7) NPROT
    READ(8) NSHEM,(MAX(I),I=1,NCOL)
    DO 14 L=1,4
        TEMP=0.
        DO 13 KK=1,NCOL
            TEM=NPROT(KK,L)-MAX(KK)
        13 TEMP=TEMP+TEM*TEM
    14 DIST(L)=SQRT(TEMP)
    WRITE(3,8) NSHEM,DIST
C
    DO 20 I=1,24
        IFREQ(I)=0
        NPONT(I)=1
        DO 20 J=1,16
            NOS(I,J)=0
    20 ARY(I,J)=BLANK
C
    SUM=0.
    DO 50 I=1,NINST
        READ(3) NPAT,(INST(K),K=1,NCOL),PR,FMS
        SUM=SUM+FMS
        DO 16 L=1,4
            TEMP=0.
            DO 15 KK=1,NCOL
                TEM=NPROT(KK,L)-INST(KK)
            15 TEMP=TEMP+TEM*TEM

```

```

16 DIST(L)=SORT(TEMP)
   WRITE(3,8) NPAT,DIST
C
   DO 40 J=1,24
   IF (FMS-TITLE(J)) 30,30,40
30  KK=NPONT(J)
   NOS(J,KK)=NPAT
   NPONT(J)=NPONT(J)+1
   IFREQ(J)=IFREQ(J)+1
   GO TO 50
40  CONTINUE
50  CONTINUE
   AVER=SUM/FLOAT(NINST)
   WRITE(3,2) AVER
   DO 70 I=1,24
   DO 70 J=1,16
   IF (IFREQ(I)-J) 70,60,60
60  ARY(I,J)=POINT
70  CONTINUE
   WRITE(3,3) ARY
   WRITE(3,5) TITLE
   WRITE(3,4) IFREQ
   IMAX=1
   DO 53 IB=1,24
   IF (IFREQ(IB)-IMAX) 53,52,52
52  IMAX=IFREQ(IB)
53  CONTINUE
   WRITE(3,865)
865 FORMAT(1H1)
   DO 86 I=1,IMAX
   II=IMAX-I+1
86  WRITE(3,4)(NOS(IX,II),IX=1,24)
   WRITE(3,5) TITLE
   WRITE(3,4) IFREQ
   CALL EXIT
   END

```

VARIABLE ALLOCATIONS

USER(R)=0004-0000	ARY(R)=0304-0006	TITLE(R)=0334-0306	DIST(R)=033C-
BLANK(R)=0342	SUM(R)=0344	PR(R)=0346	FMS(R)=0348
MAX(I)=036D-034E	INST(I)=038D-036E	IFREQ(I)=03A5-038E	NOS(I)=0525-
IMO(I)=0576	IDN(I)=0577	IYR(I)=0578	NCOL(I)=0579
I(I)=057C	L(I)=057D	KK(I)=057E	J(I)=057F
IMAX(I)=0582	IB(I)=0583	II(I)=0584	IX(I)=0585

UNREFERENCED STATEMENTS

10

STATEMENT ALLOCATIONS

1	=0592	2	=05AA	3	=0584	4	=0587	5	=058D	6	=05C4	7
---	-------	---	-------	---	-------	---	-------	---	-------	---	-------	---

865	=05DE	13	=0669	14	=067A	20	=06BA	15	=0724	16	=0735	30
70	=07C6	52	=0800	53	=0809	86	=0822					

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS

FSQRT	FADD	FSUBX	FMPY	FLD	FSTO	FSTOX	FDVR	FLOAT	IS
MCMP	MIOAI	MIOAF	MIOFX	MIOIX	MIOF	MIOI	SUBSC	REWND	UIO
MAGT	PRNTN	CBPRT	CARDN						

REAL CONSTANTS

.000000E 00=0598

INTEGER CONSTANTS

8=058A	2=058B	3=059C	1=058D	4=058E	24=058F
--------	--------	--------	--------	--------	---------

CORE REQUIREMENTS FOR DVAR

COMMON	0	INSKEL COMMON	0
VARIABLES	1416	PROGRAM	716

END OF COMPILATION

VARGUS 9	5	26	71	I.S.C.S.	SCHEMA 2000.	30	INST	45735334744365	47457436366455	45743737437635
2000				43574654565575				7.21110	7.21110	7.93725
2001				0.00000				7.48331	7.21110	7.41619
2002				2.82942				8.06225	8.30662	8.36660
2003				2.23606				7.34846	7.74596	8.77496
2004				2.44949				7.41619	7.14142	7.74596
2005				1.00000				8.36660	8.71779	9.53939
2006				3.16227				7.21110	7.21110	7.93725
2007				0.00000				7.41619	7.68114	7.87400
2008				1.00000				6.92820	6.63325	7.93725
2009				1.41421				7.34846	6.92820	8.88819
2010				2.44949				7.28011	7.28011	8.00000
2011				1.00000				8.18535	8.66025	8.60232
2012				2.64575				7.21110	6.92820	8.18535
2013				1.41421				7.74596	9.38083	10.34408
2014				3.46410				7.21110	7.21110	7.93725
2015				0.00000				6.85565	7.14142	8.48528
2016				1.73205				8.36660	7.87400	9.32737
2017				2.44949				7.21110	7.21110	7.93725
2018				0.00000				7.61577	7.21110	7.81025
2019				2.00000				8.00000	7.74596	7.93725
2020				1.00000				6.70820	7.00000	7.61577
2021				1.73205				6.24499	6.85565	7.07106
2022				2.44949				7.48331	6.32455	7.28011
2023				1.73205				7.68114	7.00000	8.48528
2024				0.00000				7.21110	7.21110	7.93725
2025				2.00000				7.21110	6.63325	8.42615
2026				1.00000				6.70820	7.00000	7.61577
2027				0.00000				7.21110	7.21110	7.93725
2028				1.00000				7.00000	7.28011	7.61577
2029				1.41421				7.34846	7.48331	8.42615
2030				2.64575				7.00000	7.81025	8.71779
AVERAGE	PV =	0.25238								
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

0.00	0.07	0.14	0.21	0.28	0.35	0.42	0.50	0.57	0.64	0.71	0.78	0.85	0.92	1.00
5	0	6	3	3	3	1	6	0	1	0	1	0	1	0

0	0	2028	0	0	0	0	2030	0	0	0	0	0	0	0
2027	0	2026	0	0	0	0	2022	0	0	0	0	0	0	0
2024	0	2020	0	0	0	0	2016	0	0	0	0	0	0	0
2017	0	2010	2029	2023	2025	0	2011	0	0	0	0	0	0	0
2014	0	2007	2012	2021	2019	0	2009	0	0	0	0	0	0	0
2006	0	2004	2008	2015	2018	2002	2003	0	2001	0	2005	0	2013	0
0.00	0.07	0.14	0.21	0.28	0.35	0.42	0.50	0.57	0.64	0.71	0.78	0.85	0.92	1.00
5	0	6	3	3	3	1	6	0	1	0	1	0	1	0

I. Identification

1. Program Name: VARGUS 10A
2. Category: Pattern Generator
3. Purpose: Input basic parameters, identification information, and matrices to be used in the generation of VARGUS 10-type patterns. See Subroutine JØIN for details of generation.
4. Date: July, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FØRTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: The final pattern matrix is a 48 x 48 matrix. Each of the corner and side submatrices are effectively 16 x 16, but each is represented internally by a string of subscript pairs not by a complete matrix. The first subscript of each pair is considered to be a row index and the second a column index. All corner matrices begin at (16, 10) and end at (10, 16). All side matrices begin at (10, 1) and end at (10, 16).

4. Input Data:

- a. A random-number generator START card under format (17I4,4I1). See subroutine RANDY documentation for further details about the START card.
- b. IDA, NØPAT (4I3). Day, month and year in numbers, followed by the number of patterns to be generated.
- c. NH, NL (2I3). Number of side matrices and number of corner matrices, respectively.
- d. MATH (25I3). This matrix contains in each row the side submatrices in subscript form. Four cards are required for each of the NH side submatrices. See Table in Remarks.
- e. MATL (25I3). This matrix contains in each row the corner submatrices in subscript form. Four cards are required for each of the NL corner submatrices.

5. Output Data: (To printer). See subroutine PATØT for details.

6. Subroutines Used: RANDY: A pseudo-random number generator.

JØIN: Selects, positions and rotates side and corner matrices.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

The following information is a list of the 40 side submatrices and 34 corner submatrices in subscript form. The side submatrices are identified by a leading H in the identification code and the corner submatrices by a leading L.

H1: (10,1) (9,1) (8,1) (7,2) (6,2) (5,3) (4,3) (3,3) (4,4)
(4,5) (5,6) (5,7) (6,7) (6,8) (7,9) (7,10) (8,11) (8,12)
(9,13) (9,14) (10,15) (10,16)

H2: (10,1) (9,1) (8,1) (7,2) (6,2) (5,3) (4,4) (3,4) (4,5)
(5,5) (6,6) (7,6) (8,7) (9,7) (10,8) (11,8) (12,9) (13,9)
(14,10) (15,11) (16,11) (16,12) (15,12) (15,11) (14,12)
(13,13) (13,14) (12,14) (12,15) (11,15) (10,16)

H3: (10,1) (11,1) (11,2) (12,3) (12,4) (13,5) (14,5) (14,6)
(15,7) (16,8) (16,9) (15,10) (15,11) (14,11) (14,12)
(13,13) (13,14) (12,15) (11,16) (10,16)

H4: (10,1) (9,2) (9,3) (8,3) (8,4) (7,5) (6,6) (6,7) (5,8)
(5,9) (4,9) (4,10) (3,11) (4,12) (5,13) (6,13) (7,14)
(8,15) (9,15) (9,16) (10,16)

H5: (10,1) (10,2) (9,2) (8,2) (7,3) (6,4) (5,5) (4,6) (4,7)
(3,8) (3,9) (3,10) (4,11) (4,12) (5,13) (6,13) (7,13)
(8,13) (9,13) (10,14) (10,15) (10,16)

H6: (10,1) (10,2) (10,3) (11,4) (12,5) (12,6) (12,7) (12,8)
(11,8) (10,8) (9,8) (9,9) (9,10) (9,11) (9,12) (9,13)
(9,14) (9,15) (9,16) (10,16)

H7: (10,1) (11,1) (12,2) (13,3) (14,4) (14,5) (15,6) (15,7)
 (15,8) (15,9) (15,13) (15,11) (15,12) (14,13) (13,14)
 (12,15) (11,16) (10,16) (15,10)

 H8: (10,1) (11,1) (12,2) (12,3) (13,3) (13,4) (14,5) (14,6)
 (15,7) (15,8) (15,9) (15,10) (15,11) (15,12) (15,13)
 (15,14) (14,15) (13,15) (13,16) (12,16) (11,16) (10,16)

 H9: (10,1) (9,1) (8,1) (7,1) (6,2) (5,2) (5,3) (4,3) (4,4)
 (3,5) (3,6) (3,7) (3,8) (3,9) (4,10) (4,11) (5,12) (6,13)
 (7,14) (8,15) (9,15) (9,16) (10,16)

 H10: (10,1) (9,1) (9,2) (8,3) (7,4) (7,5) (6,5) (6,6) (5,7)
 (5,8) (4,9) (4,10) (4,11) (4,12) (5,12) (6,12) (7,12)
 (8,12) (9,12) (10,11) (10,12) (10,13) (10,14) (10,15)
 (10,16)

 H11: (10,1) (9,1) (9,2) (8,3) (8,4) (8,5) (7,6) (7,7) (7,8)
 (7,9) (7,10) (8,11) (8,12) (9,12) (9,13) (10,13) (11,14)
 (12,14) (13,14) (14,14) (13,15) (12,15) (12,16) (11,16)
 (10,16)

 H12: (10,1) (9,2) (9,3) (9,4) (9,5) (9,6) (9,7) (8,8) (8,9)
 (8,10) (8,11) (8,12) (8,13) (8,14) (8,15) (9,15) (10,15)
 (10,16)

 H13: (10,1) (9,2) (8,3) (8,4) (7,4) (7,5) (6,6) (6,7) (5,7)
 (5,8) (5,9) (5,10) (6,10) (7,11) (8,11) (8,12) (8,13)
 (9,13) (9,14) (10,14) (10,15) (10,16)

 H14: (10,1) (11,2) (12,3) (13,4) (14,4) (15,5) (15,6) (15,7)
 (15,8) (15,9) (15,10) (15,11) (14,12) (14,13) (14,14)
 (13,14) (12,14) (12,15) (11,15) (10,16)

 H15: (10,1) (10,2) (10,3) (10,4) (10,5) (10,6) (10,7) (10,8)
 (10,9) (10,10) (10,11) (10,12) (10,13) (10,14) (10,15)
 (10,16)

 H16: (10,2) (11,2) (12,3) (13,4) (13,5) (14,6) (14,7) (15,8)
 (15,9) (15,10) (15,11) (14,12) (14,13) (13,14) (12,14)
 (11,15) (10,15) (10,16) (10,1)

 H17: (10,1) (10,2) (11,2) (12,2) (13,3) (14,4) (15,5) (16,6)
 (15,7) (14,8) (13,9) (13,10) (12,11) (12,12) (11,13)
 (11,14) (10,15) (10,16)

 H18: (10,1) (10,2) (9,2) (8,2) (7,3) (7,4) (6,5) (6,6) (6,7)
 (6,8) (6,9) (6,10) (6,11) (6,12) (7,13) (8,14) (9,15)
 (10,15) (10,16)

H19: (10,1) (10,2) (11,2) (12,3) (13,4) (13,5) (14,6) (14,7)
(14,8) (14,9) (13,10) (12,11) (11,11) (10,12) (10,13)
(10,14) (10,15) (10,16)

H20: (10,1) (10,2) (9,2) (8,2) (7,3) (7,4) (7,5) (7,6) (7,7)
(7,8) (8,9) (8,10) (9,11) (9,12) (10,13) (10,14) (10,15)
(10,16)

H21: (10,1) (10,2) (10,3) (10,4) (10,5) (10,6) (9,6) (8,6)
(7,7) (6,7) (6,8) (6,9) (6,10) (6,11) (6,12) (6,13)
(6,14) (7,14) (8,14) (9,14) (10,14) (10,15) (10,16)

H22: (10,1) (10,2) (11,2) (12,2) (13,2) (14,2) (15,3) (16,4)
(16,5) (16,6) (16,7) (16,8) (16,9) (16,10) (16,11)
(16,12) (15,13) (15,14) (14,15) (13,15) (12,15) (11,15)
(10,15) (10,16)

H23: (10,1) (10,2) (9,3) (9,4) (8,5) (7,5) (6,5) (5,5) (4,5)
(4,6) (4,7) (4,8) (4,9) (4,10) (3,11) (3,12) (3,13)
(4,13) (5,13) (6,13) (7,13) (8,13) (9,13) (10,13) (10,14)
(10,15) (10,16)

H24: (10,1) (10,2) (9,2) (8,2) (7,3) (6,3) (5,4) (5,5) (4,6)
(4,7) (4,8) (4,9) (4,10) (4,11) (6,12) (7,12) (8,11)
(5,12) (9,11) (9,12) (10,13) (10,14) (10,15) (10,16)

H25: (10,1) (10,2) (9,2) (9,3) (9,4) (9,5) (9,6) (9,7) (10,8)
(10,9) (11,10) (12,11) (12,13) (12,12) (11,14) (10,15)
(10,16)

H26: (10,1) (10,2) (9,3) (9,4) (9,5) (8,6) (8,7) (8,8) (7,9)
(7,10) (6,11) (6,12) (6,13) (6,14) (6,15) (7,15) (8,15)
(9,15) (10,15) (10,16)

H27: (10,1) (10,2) (9,2) (8,2) (7,2) (6,2) (5,3) (5,4) (6,5)
(6,6) (6,7) (7,8) (7,9) (6,10) (6,11) (6,12) (6,13)
(6,14) (7,15) (8,15) (9,14) (10,14) (10,15) (10,16)

H28: (10,1) (10,2) (9,2) (8,2) (7,2) (6,2) (5,3) (4,4) (5,5)
(6,6) (7,7) (8,7) (9,8) (10,9) (9,10) (9,11) (8,12)
(8,13) (8,14) (9,15) (10,15) (10,16)

H29: (10,1) (10,2) (11,2) (12,2) (13,2) (14,2) (15,3) (15,4)
(14,5) (14,6) (13,7) (13,8) (12,9) (12,10) (11,11) (11,12)
(10,13) (10,14) (9,15) (10,15) (10,16)

H30: (10,1) (10,2) (9,2) (8,2) (7,2) (7,3) (6,3) (5,4) (4,5)
(5,6) (5,7) (6,8) (6,9) (7,10) (8,11) (8,12) (9,13) (9,14)
(9,15) (10,15) (10,16)

H31: (10,1) (10,2) (9,2) (8,2) (7,2) (6,2) (5,2) (4,3) (4,4)
(4,5) (4,6) (3,7) (3,8) (3,9) (3,10) (3,11) (3,12) (3,13)
(3,14) (3,15) (4,15) (5,15) (6,15) (7,15) (8,15) (9,15)
(10,15) (10,16)

H32: (10,1) (10,2) (11,2) (11,3) (12,4) (12,5) (13,6) (13,7)
(14,8) (14,9) (14,10) (15,11) (15,12) (14,12) (13,12)
(12,12) (11,13) (10,13) (9,13) (9,14) (10,15) (10,16)

H33: (10,1) (10,2) (9,3) (8,4) (10,5) (10,6) (10,7) (11,8)
(12,9) (13,10) (14,11) (15,12) (14,13) (13,13) (12,14)
(11,14) (10,15) (10,16) (9,4)

H34: (10,1) (10,2) (9,2) (8,2) (7,2) (6,3) (5,4) (5,5) (4,6)
(4,7) (4,8) (4,9) (5,10) (5,11) (6,12) (7,13) (7,14)
(8,14) (8,15) (9,15) (10,15) (10,16)

H35: (10,1) (10,2) (11,2) (12,3) (12,4) (13,5) (13,6) (14,7)
(15,8) (15,9) (14,10) (13,11) (12,12) (11,12) (10,13)
(10,14) (10,15) (10,16)

H36: (10,1) (10,2) (9,3) (9,4) (10,4) (11,4) (12,4) (13,4)
(14,4) (15,4) (15,5) (15,6) (15,7) (15,8) (15,9) (15,10)
(15,11) (15,12) (14,12) (13,12) (12,12) (11,12) (10,12)
(10,13) (10,14) (10,15) (10,16)

H37: (10,1) (10,2) (11,2) (12,2) (13,2) (14,2) (13,3) (14,3)
(15,4) (15,5) (15,6) (15,7) (15,8) (15,9) (15,10) (15,11)
(15,12) (15,13) (15,14) (15,15) (14,15) (13,15) (12,15)
(11,15) (10,15) (10,16)

H38: (10,1) (10,2) (9,3) (9,4) (9,5) (8,6) (8,7) (8,8) (8,9)
(7,10) (7,11) (7,12) (7,13) (7,14) (6,15) (7,15) (8,15)
(9,15) (10,15) (10,16)

H39: (10,1) (10,2) (9,3) (9,4) (9,5) (9,6) (9,7) (9,8) (9,9)
(9,10) (9,11) (9,12) (9,13) (9,14) (9,15) (10,15) (10,16)

H40: (10,1) (10,2) (9,2) (8,2) (7,3) (6,3) (5,4) (5,5) (6,6)
(7,7) (8,8) (9,8) (10,8) (9,9) (8,10) (7,11) (7,12) (8,13)
(9,13) (10,14) (11,14) (10,15) (10,16)

L1: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10)
 (9,10) (8,10) (7,10) (6,10) (5,11) (4,12) (4,13) (4,14)
 (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)

L2: (16,10) (16,9) (15,8) (15,7) (14,6) (13,5) (12,4) (11,4)
 (10,3) (9,3) (8,3) (7,3) (6,4) (5,4) (4,5) (4,6) (3,7)
 (3,8) (3,9) (3,10) (3,11) (4,12) (5,13) (6,14) (7,15)
 (8,16) (9,16) (10,16)

L3: (16,10) (15,11) (14,11) (13,11) (12,10) (12,9) (12,8)
 (12,7) (12,6) (12,5) (12,4) (11,4) (10,4) (9,4) (8,4)
 (7,4) (6,4) (5,4) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9)
 (4,10) (4,11) (4,12) (4,13) (4,14) (4,15) (5,15) (6,15)
 (7,15) (8,15) (9,15) (10,15) (10,16)

L4: (16,10) (16,9) (15,8) (15,7) (15,6) (15,5) (15,4) (14,4)
 (13,4) (12,4) (11,4) (10,4) (9,3) (8,3) (7,3) (6,3)
 (5,3) (4,3) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9) (4,10)
 (4,11) (4,12) (4,13) (4,14) (4,15) (5,15) (6,15) (7,15)
 (8,15) (9,16) (10,16)

L5: (16,10) (16,11) (16,12) (16,13) (15,13) (15,14) (14,15)
 (13,15) (13,16) (12,16) (11,16) (10,16)

L6: (16,10) (16,9) (15,8) (14,7) (13,6) (12,5) (12,4) (11,3)
 (10,4) (9,5) (8,6) (7,6) (6,7) (5,8) (4,9) (3,10) (4,11)
 (5,12) (6,13) (6,14) (7,15) (8,16) (9,16) (10,16)

L7: (16,10) (15,9) (15,10) (15,8) (15,7) (15,6) (14,5) (13,5)
 (12,5) (11,5) (10,5) (9,5) (9,6) (8,6) (7,6) (7,7) (6,7)
 (5,8) (5,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15)
 (5,15) (5,16) (6,16) (7,16) (8,16) (9,16) (10,16)

L8: (16,10) (16,9) (16,8) (16,7) (15,6) (15,5) (15,4) (14,4)
 (13,3) (12,3) (11,3) (10,3) (9,3) (8,3) (7,3) (6,3)
 (5,4) (4,4) (4,5) (4,6) (3,7) (3,8) (3,9) (3,10) (3,11)
 (3,12) (3,13) (3,14) (4,15) (5,15) (6,15) (7,15) (8,15)
 (9,15) (10,16)

L9: (16,10) (15,11) (14,11) (13,12) (12,12) (11,13) (10,14)
 (10,15) (10,16)

L10: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (11,11)
 (11,12) (10,13) (10,14) (10,15) (10,16)

L11: (16,10) (15,10) (14,11) (13,11) (12,12) (11,13) (11,14)
 (11,15) (10,16)

L12: (16,10) (15,9) (14,9) (13,9) (12,9) (11,8) (10,8) (9,8)
 (8,7) (7,7) (7,8) (7,9) (6,10) (6,11) (6,12) (6,13)
 (6,14) (6,15) (6,16) (7,16) (8,16) (9,16) (10,16)

L13: (16,10) (15,10) (14,10) (13,10) (12,10) (12,9) (12,8)
 (12,7) (12,6) (12,5) (12,4) (11,4) (10,4) (9,4) (8,4)
 (7,4) (6,4) (5,4) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9)
 (4,10) (4,11) (4,12) (4,13) (4,14) (6,15) (5,15) (7,14)
 (8,14) (9,14) (10,14) (10,15) (10,16)

 L14: (16,10) (15,11) (14,12) (13,12) (12,13) (11,14) (11,15)
 (10,16)

 L15: (16,10) (16,11) (15,12) (14,12) (13,13) (12,13) (11,13)
 (10,13) (9,13) (9,14) (9,15) (10,16)

 L16: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10)
 (9,10) (8,10) (7,10) (6,10) (5,11) (5,12) (5,13) (4,14)
 (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)

 L17: (16,10) (15,10) (15,9) (15,8) (15,7) (15,6) (15,5) (15,4)
 (14,3) (13,3) (12,3) (11,3) (10,3) (9,3) (8,3) (7,3)
 (6,3) (5,3) (4,4) (4,5) (4,6) (4,7) (5,8) (5,9) (5,10)
 (5,11) (5,12) (6,12) (7,12) (8,12) (9,12) (10,12) (11,12)
 (11,13) (11,14) (10,15) (10,16)

 L18: (16,10) (16,11) (15,11) (15,12) (14,13) (14,14) (14,15)
 (13,15) (12,15) (11,14) (10,14) (10,15) (10,16)

 L19: (16,10) (16,9) (16,8) (15,7) (14,6) (13,5) (12,5) (11,4)
 (10,4) (9,5) (8,5) (7,6) (6,7) (6,8) (5,9) (5,10) (5,11)
 (6,12) (6,13) (7,14) (8,15) (9,16) (10,16)

 L20: (16,10) (15,9) (15,8) (14,8) (13,8) (12,8) (11,8) (10,8)
 (9,9) (8,10) (8,11) (8,12) (8,13) (8,14) (9,14) (10,14)
 (10,15) (10,16)

 L21: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10)
 (9,11) (9,12) (9,13) (9,14) (10,15) (10,16)

 L22: (16,10) (16,11) (16,12) (15,13) (15,14) (14,14) (13,14)
 (12,15) (11,15) (10,16)

 L23: (16,10) (15,9) (14,8) (13,7) (12,7) (11,6) (10,6) (9,6)
 (8,6) (7,7) (7,8) (7,9) (7,10) (7,11) (7,12) (7,13)
 (8,14) (9,15) (10,16)

 L24: (16,10) (15,11) (14,12) (14,13) (13,14) (13,15) (12,15)
 (11,16) (10,16)

 L25: (16,10) (16,9) (15,8) (14,7) (13,6) (12,6) (11,6) (10,6)
 (9,6) (8,7) (7,7) (6,8) (6,9) (5,10) (5,11) (5,12) (5,13)
 (6,14) (7,15) (8,16) (9,16) (10,16)

L26: (16,10) (15,9) (14,9) (13,8) (12,8) (11,8) (10,7) (9,7)
 (8,8) (7,9) (6,10) (6,11) (5,12) (5,13) (6,14) (7,15)
 (8,16) (9,16) (10,16)

 L27: (16,10) (15,10) (14,10) (13,9) (12,9) (11,9) (10,9) (9,9)
 (8,9) (7,10) (7,11) (7,12) (6,13) (6,14) (7,15) (8,16)
 (9,16) (10,16)

 L28: (16,10) (15,10) (14,10) (13,11) (13,12) (11,12) (11,13)
 (10,14) (10,15) (10,16) (12,12)

 L29: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10)
 (10,9) (10,8) (10,7) (10,6) (10,5) (10,4) (10,3) (9,3)
 (8,3) (7,3) (6,3) (6,4) (6,5) (6,6) (6,7) (5,7) (4,7)
 (4,8) (4,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15)
 (4,16) (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)

 L30: (16,10) (16,9) (16,8) (15,7) (14,6) (13,5) (12,5) (11,4)
 (10,4) (9,4) (8,4) (7,5) (6,5) (5,6) (5,7) (5,8) (5,9)
 (6,10) (7,11) (8,11) (9,12) (10,12) (11,12) (9,13) (8,14)
 (9,15) (10,16)

 L31: (16,10) (15,9) (14,9) (13,8) (12,8) (11,8) (10,7) (9,7)
 (8,7) (7,8) (6,9) (7,10) (8,11) (9,11) (9,12) (9,13)
 (9,14) (9,15) (10,16)

 L32: (16,10) (16,9) (15,8) (14,7) (13,8) (13,9) (12,10) (11,10)
 (10,9) (9,8) (8,7) (7,8) (6,9) (5,10) (4,11) (5,12) (6,12)
 (7,14) (8,15) (9,16) (10,16) (7,13)

 L33: (16,10) (15,10) (14,11) (13,11) (12,12) (11,12) (10,12)
 (9,12) (8,13) (9,14) (9,15) (10,16)

 L34: (16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (11,11)
 (11,12) (11,13) (11,14) (11,15) (10,16)

```

// JOB                11000    30 JUN 71 13.695 HRS
// * U-2050-003
// FOR VAR10    30 JUN 71 13.696 HRS
*NON PROCESS PROGRAM
*IOCS(CARD,1443 PRINTER)
*ONE WORD INTEGERS
*LIST ALL
C   VARGUS 10-A
C
C   CHIP BRUCE
C
C   INSTITUTE FOR THE STUDY OF COGNITIVE SYSTEMS
C
C   VARGUS 10 A
C   AUGUST 1963
C
C   IBM 1800 MPX
C   1130 / 1800 BASIC FORTRAN IV
C   WITH SUBROUTINES RANDY, JOIN, AND PATOT
C
C   ACCEPTS NH H MATRICES AND NL L MATRICES
C   AND POSITIONS FOUR OF EACH TYPE, AFTER APPROPRIATE ROTATION,
C   IN A STANDARD 48 X 48 MATRIX, SO THAT A CONTINUOUS CLOSED FIGURE
C   IS FORMED
C
C   INTEGER HORL(8)
C   DIMENSION MATS (48, 48), MATH (50, 100), MATL (50, 100), IDA (3)
C
C   CALL RANDY (R)
C
C   READ DATE AND NUMBER OF PATTERNS
C   READ (2, 10) IDA, NOPAT
10  FORMAT (4I3)
C
C   READ PARAMETERS
C   NH IS THE NUMBER OF H MATRICES
C   NL IS THE NUMBER OF L MATRICES
C   READ (2, 20) NH, NL
20  FORMAT (2I3)
C
C   READ H AND L TYPE SUBMATRICES
C   READ (2, 30) ((MATH (I, J), J = 1, 100), I = 1, NH)
C   READ (2, 30) ((MATL (I, J), J = 1, 100), I = 1, NL)
30  FORMAT (25I3)
C
C   START LOOP OVER PATTERNS
C   DO 150 NO = 1, NOPAT
C
C   PRINT HEADING
C   WRITE (3, 120) NO, IDA
120  FORMAT (11H1VARGUS 10A, 10X, 14HPATTERN NUMBER, I3, 10X, 3I3)
C
C   CALL JOIN (MATS, MATH, MATL, NH, NL, HORL)
C   WRITE(3,145)(HORL(IXX),IXX=1,8)

```

```

145 FORMAT ('0 L (1, 1) H (1, 2) L (1, 3) H (2, 3) L (3, 3) H (3,
1 2) L (3, 1) H (2, 1)' / 17, 7110)
C
C PRINT PATTERN
CALL PATOT (MATS, 48, 48)
150 CONTINUE
C
CALL EXIT
END

VARIABLE ALLOCATIONS
R(R )=0000 MATS(I )=0901-0002 MATH(I )=1C89-0902 MATL(I )=3011-
NOPAT(I )=301D NH(I )=301E NL(I )=301F I(I )=3020
IXX(I )=3023

STATEMENT ALLOCATIONS
10 =302C 20 =302F 30 =3032 120 =3035 145 =304A 150 =3107

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS
RANDY JOIN PATOT HRED MWRT MCOMP MIOAI MIOIX MIOI SUE

INTEGER CONSTANTS
2=3026 1=3027 100=3028 3=3029 8=302A 48=302B

CORE REQUIREMENTS FOR VAR10
COMMON 0 INSKEL COMMON 0
VARIABLES 12326 PROGRAM 236

```

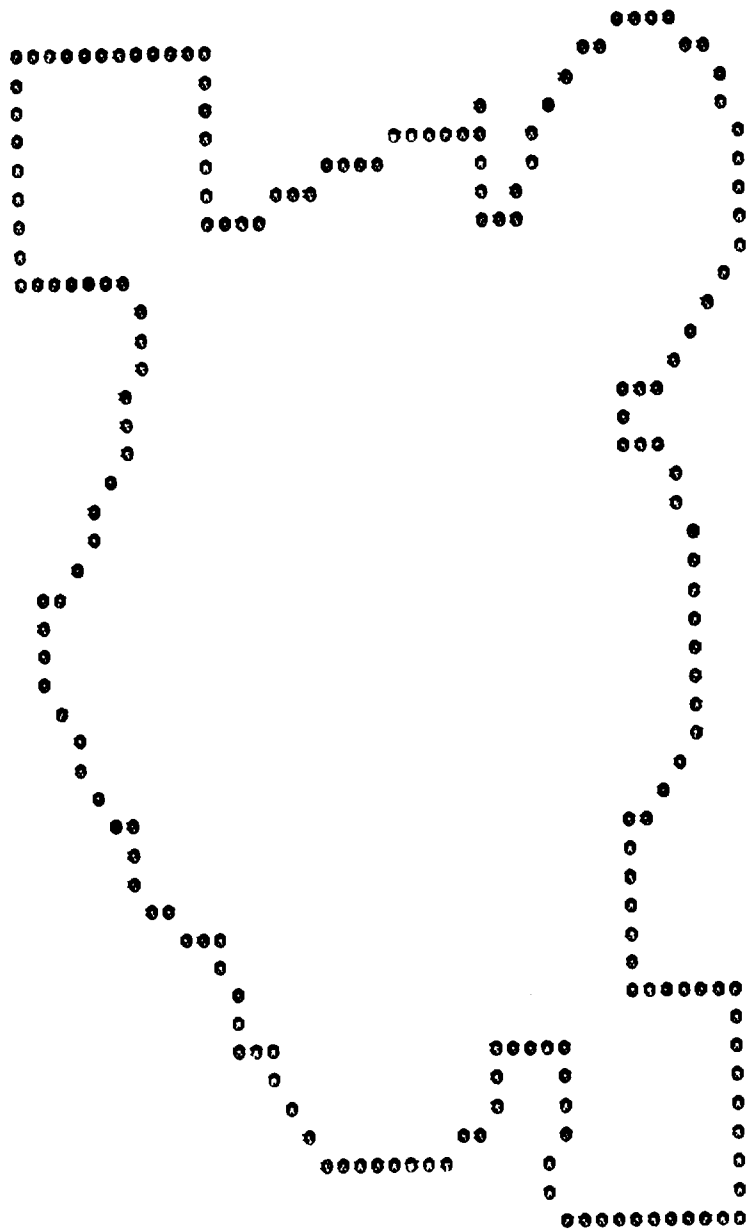
END OF COMPILATION

VARGUS 10A

PATTERN NUMBER 1

1 6 71

L (1, 1)	H (1, 2)	L (1, 3)	H (2, 3)	L (3, 3)	H (3, 2)	L (3, 1)	H (2, 1)
26	22	27	11	4	11	10	9



I. Identification

1. Subroutine Name: JØIN
2. Category: Pattern Generation
3. Purpose: Create pattern matrices by randomly selecting four corner submatrices and four side submatrices. These eight submatrices are joined together into a closed figure by performing the appropriate rotation and positioning and storing into the pattern matrix.
4. Date: August, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: See VARGUS 10A documentation.
4. Usage: CALL JØIN (MATS, MATH, MATL, NH, NL).
5. Subroutine Parameters:
 - MATS: The 48 x 48 output matrix.
 - MATH: The matrix each row of which is a string of subscript pairs for side submatrices.
 - MATL: The matrix each row of which is a string of subscript pairs for corner submatrices.
 - NH: The number of side submatrices.
 - NL: The number of corner submatrices.

6. Output: HØRL: A vector the eight elements of which are numbers corresponding to the corner and side submatrices chosen to form a particular pattern. The elements are in the order beginning at the upper left of the pattern and moving clockwise around the figure.

7. Subroutine Required:

RANDY: A pseudo-random number generator.

RØTAT: The subroutine used to rotate the side and corner submatrices so that they form a continuous closed figure.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

ROTAT
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13.722 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
      SUBROUTINE JOIN (MATS, MATH, MATL, NH, NL, HORL)
      INTEGER SUBMX (16, 16), CHANI(8), CHANJ (8), HORL (8)
      DIMENSION MATS (48, 48), MATH (50, 100), MATL (50, 100)
      DATA CHANI / 0, 0, 0, 16, 32, 32, 32, 16 /
      DATA CHANJ / 0, 16, 32, 32, 32, 16, 0, 0 /

C
C      CLEAR OUTPUT MATRIX MATS
      DO 40 I = 1, 48
      DO 40 J = 1, 48
40 MATS (I, J) = 0

C
      DO 110 IBLOC = 1, 8

C
C      CLEAR SUBMATRIX SUBMX
      DO 50 I = 1, 16
      DO 50 J = 1, 16
50 SUBMX (I, J) = 0

C
      NTURN = (IBLOC - 1) / 2
      CALL RANDY(R)

C
      IF (IBLOC - 2 * (IBLOC / 2)) 60, 60, 80

C
60 I = R * NH + 1
      DO 70 J=1,99,2
      K = MATH (I, J)
      L = MATH (I, J + 1)
      IF(K) 70,100,70
70 SUBMX (K, L) = 1
      GO TO 100

C
80 I = R * NL + 1
      DO 90 J=1,99,2
      K = MATL (I, J)
      L = MATL (I, J + 1)
      IF(K) 90,100,90
90 SUBMX (K, L) = 1
100 CALL ROTAT (SUBMX, 16, 16, NTURN)
      HORL (IBLOC) = I

C
      DO 110 I = 1, 16
      DO 110 J = 1, 16
      II = I + CHANI (IBLOC)
      JJ = J + CHANJ (IBLOC)
110 MATS (II, JJ) = SUBMX (I, J)
      RETURN
      END

```

VARIABLE ALLOCATIONS

R(R)=0000 SUBMX(I)=0103-0004 CHANI(I)=010B-0104 CHANJ(I)=0113-
IBLOC(I)=0116 NTURN(I)=0117 K(I)=0118 L(I)=0119

STATEMENT ALLOCATIONS

40 =0141 50 =016A 60 =01A1 70 =01CE 80 =01E5 90 =0212 100

FEATURES SUPPORTED

NONPROCESS

ONE WORD INTEGERS

CALLED SUBPROGRAMS

RANDY ROTAT FADD FMPY FSTO IFIX FLOAT ISTOX STFAC SBF

INTEGER CONSTANTS

1=0120 48=0121 0=0122 8=0123 16=0124 2=0125

CORE REQUIREMENTS FOR JOIN

COMMON 0 INSKEL COMMON 0
VARIABLES 298 PROGRAM 354

END OF COMPILATION

I. Identification

1. Subroutine Name: RØTAT
2. Category: Pattern Modifier
3. Purpose: Accepts a matrix and rotates it by multiples of 90 degrees.
4. Date: July, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitation: Input matrix must be dimensioned to 16 x 16 in calling routine.
4. Subroutine Usage: CALL RØTAT (MX, NI, NJ, NTURN).
5. Subroutine Parameters:
 - MX: The matrix to be rotated.
 - NI: The number of rows.
 - NJ: The number of columns.
 - NTURN: The number of 90-degree rotations.
6. Subroutines Required:
 - MØDUL: performs modular arithmetic.

III. Mathematical Method and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

MODUL

DMP FUNCTION COMPLETED

// FOR ROTAT 30 JUN 71 13.713 HRS

*ONE WORD INTEGERS

*NONPROCESS PROGRAM

*LIST ALL

 SUBROUTINE ROTAT (MX, NI, NJ, NTURN)
C ACCEPTS MATRIX MX (NI ROWS, NJ COLUMNS)
C AND ROTATES IT BY NTURN TIMES NINETY DEGREES
 DIMENSION MX (16, 16), LX (16, 16)

C
 CALL MODUL (4, NTURN, NTURN)
 NTURN = NTURN + 1
 GO TO (120, 40, 60, 80), NTURN

C
C 90 DEGREE ROTATION
40 DO 50 I = 1, NJ
 DO 50 J = 1, NI
 K = NI - J + 1
50 LX (I, J) = MX (K, I)
 GO TO 100

C
C 180 DEGREE ROTATION
60 DO 70 I = 1, NI
 DO 70 J = 1, NJ
 K = NI - I + 1
 L = NJ - J + 1
70 LX (I, J) = MX (K, L)
 M = NI
 NI = NJ
 NJ = M
 GO TO 100

C
C 270 DEGREE ROTATION
80 DO 90 I = 1, NJ
 DO 90 J = 1, NI
 K = NJ - I + 1
90 LX (I, J) = MX (J, K)

C
C LOAD ROTATED MATRIX
100 DO 110 I = 1, NJ
 DO 110 J = 1, NI
110 MX (I, J) = LX (I, J)

C
120 RETURN
 END

VARIABLE ALLOCATIONS

LX(I)=00FF-0000

I(I)=0100

J(I)=0101

K(I)=0102

STATEMENT ALLOCATIONS

40 =013D 50 =014D 60 =0175 70 =018D 80 =01C1 90 =01D1 100

FEATURES SUPPORTED

NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
MODUL COMGO ISTOX SUBSC SUBIN

INTEGER CONSTANTS
4=0108 1=0109

CORE REQUIREMENTS FOR ROTAT
COMMON 0 INSKEL COMMON 0
VARIABLES 264 PROGRAM 278

END OF COMPILATION

I. Identification

1. Subroutine Name: ~~MODUL~~
2. Category: Computational
3. Purpose: Perform modular arithmetic
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1300 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Operates on one number per call.
4. Subroutine Usage: CALL ~~MODUL~~ (~~MODOP~~, IX, ~~MODX~~).
5. Subroutine Parameters:
 - ~~MODOP~~: The modulus.
 - IX: The input value.
 - MODX: The output value.
6. Subroutines Required: None.

III. Mathematical Method and References

None.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

VAR10
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13.707 HRS
*ONE WORD INTEGERS
*LIST ALL
*NONPROCESS PROGRAM
    SUBROUTINE MODUL (MODOP,IX,MODX)
C  SUBROUTINE FOR DOING MODULAR ARITHMETIC
C  MODOP IS THE MODULAR OPERATOR
C  IX IS THE INPUT VALUE
C  MODX IS THE OUTPUT MOD VALUE
C
    MODX=IX
    GO TO 20
10  MODX=MODX-MODOP
20  IF (MODX-MODOP)30,10,10
30  CONTINUE
    RETURN
    END

```

```

STATEMENT ALLOCATIONS
10  =0013 20  =0019 30  =001F

```

```

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

```

```

CALLED SUBPROGRAMS
SUBIN

```

```

CORE REQUIREMENTS FOR MODUL
COMMON      0  INSKEL COMMON      0
VARIABLES   0  PROGRAM      34

```

```

END OF COMPILATION

```

I. Identification

1. Subroutine Name: CØNG
2. Category: Comparison of Patterns
3. Purpose: Computes an estimate of the similarity of two VARGUS 10 patterns.
4. Date: August, 1969
5. Programmer: Earl H. Greer

II. Use Information

1. Language: 1800 Basic FØRTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Matrices dimensioned to 48 x 48.
4. Subroutine Usage:
CALL CØNG (MX, NI, NJ, LX, MI, MJ, IBLUR, CØNGR).
5. Subroutine Parameters:
MX: Pattern matrix.
NI: Number of rows.
NJ: Number of columns.
LX: Comparison matrix (typically the schema).
MI: Number of rows in comparison matrix.
MJ: Number of columns in comparison matrix.
IBLUR: If IBLUR = 1, LX is blurred as well as MX.
CØNGR: Returns the similarity measure.
6. Subroutines Required:
BLUR: Blurs the input matrix using a 7 x 7 element random walk matrix.

III. Mathematical Method and References

SUM contains a measure of the degree of overlap of the two patterns. SUMZ contains a measure of the maximum possible overlap; i.e., a comparison of LX with itself. CONGR is thus the percentage of maximum possible overlap of LX with LX.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

PATOT
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.606 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
      SUBROUTINE CONG (MX, NI, NJ, LX, MI, MJ, IBLUR, CONGR)
C  SUBROUTINE CONG          EARL M. GREER          AUGUST 1, 196
C  IBM 1800
C  ACCEPTS TWO MATRICES FROM CALLING STATEMENT,
C  BLURS THE FIRST, MULTIPLIES THE TWO MATRICES,
C  AND SUMS THE RESULTING MATRIX.
C
C  IF IBLUR = 1 THE SECOND MATRIX IS ALSO BLURRED ALONG WITH THE FIRST.
C  THIS SECOND MATRIX IS SQUARED TO PROVIDE A STANDARD, THE MAXIMUM
C  POSSIBLE CONGRUENCE.
C
C  THE DIFFERENCE BETWEEN THE STANDARD AND ACTUAL CONGRUENCE IS THE
C  DEVIATION. THE ACTUAL CONGRUENCE IS DIVIDED BY THE STANDARD AND
C  MULTIPLIED BY 100 TO FIND THE PERCENT CONGRUENCE.
C
      DIMENSION MX(48,48), LX(48,48)
      COMMON FX(54,54)
      IF (IBLUR - 1) 60,50,60
50 CALL BLUR(LX, MI, MJ)
60 CALL BLUR(MX, NI, NJ)
      SUM = 0.0
      SUM2 = 0.0
      DO 70 I = 1,NI
      DO 70 J = 1,NJ
      FX(I,J) = MX(I,J) * LX(I,J)
70 SUM = SUM + FX(I,J)
      DO 90 I = 1,MI
      DO 90 J = 1,MJ
      FX(I,J) = LX(I,J) * LX(I,J)
90 SUM2 = SUM2 + FX(I,J)
      DIFF = SUM2 - SUM
      CONGR = (SUM / SUM2) * 100.0
      RETURN
      END

```

VARIABLE ALLOCATIONS
FX(RC)=FFFE-E938 SUM(R)=0000 SUM2(R)=0002 DIFF(R)=0004

STATEMENT ALLOCATIONS
50 =0030 60 =0035 70 =0062 90 =009C

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
BLUR FADDX FSUB FMPY FDIV FLD FSTO FSTOX FLOAT SU

REAL CONSTANTS

.000000E 00=000A .100000E 03=000C

INTEGER CONSTANTS

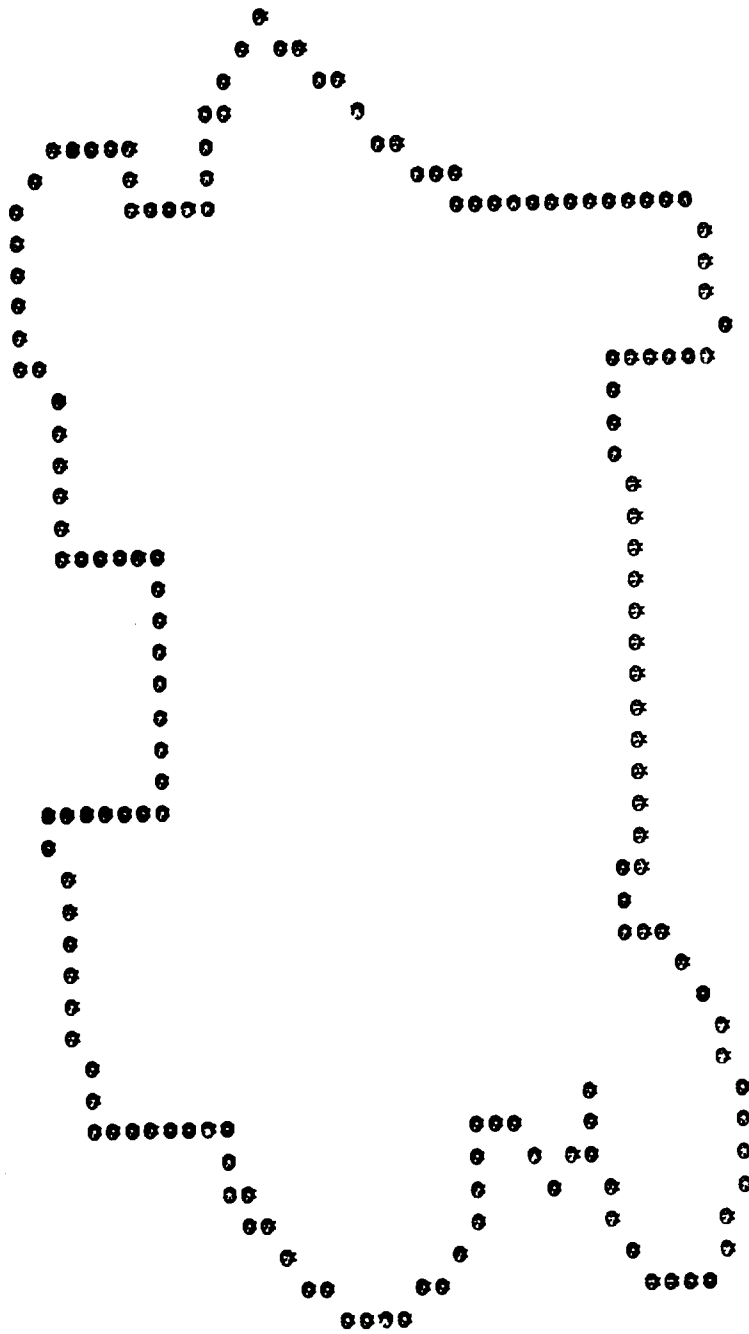
1=000E

CORE REQUIREMENTS FOR CONG

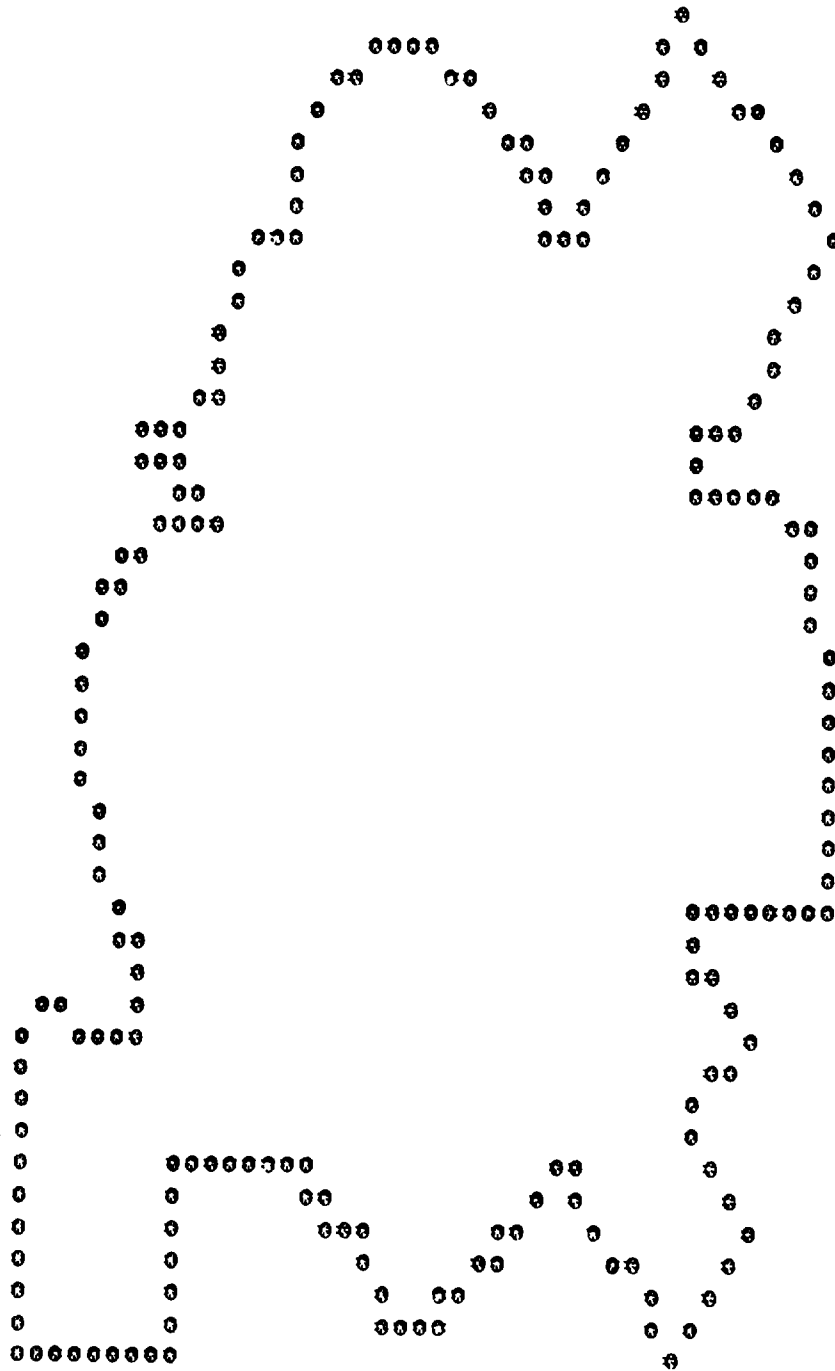
COMMON	5832	INSKEL COMMON	0
VARIABLES	10	PROGRAM	188

END OF COMPILATION

VARGUS 10A PATTERN 2



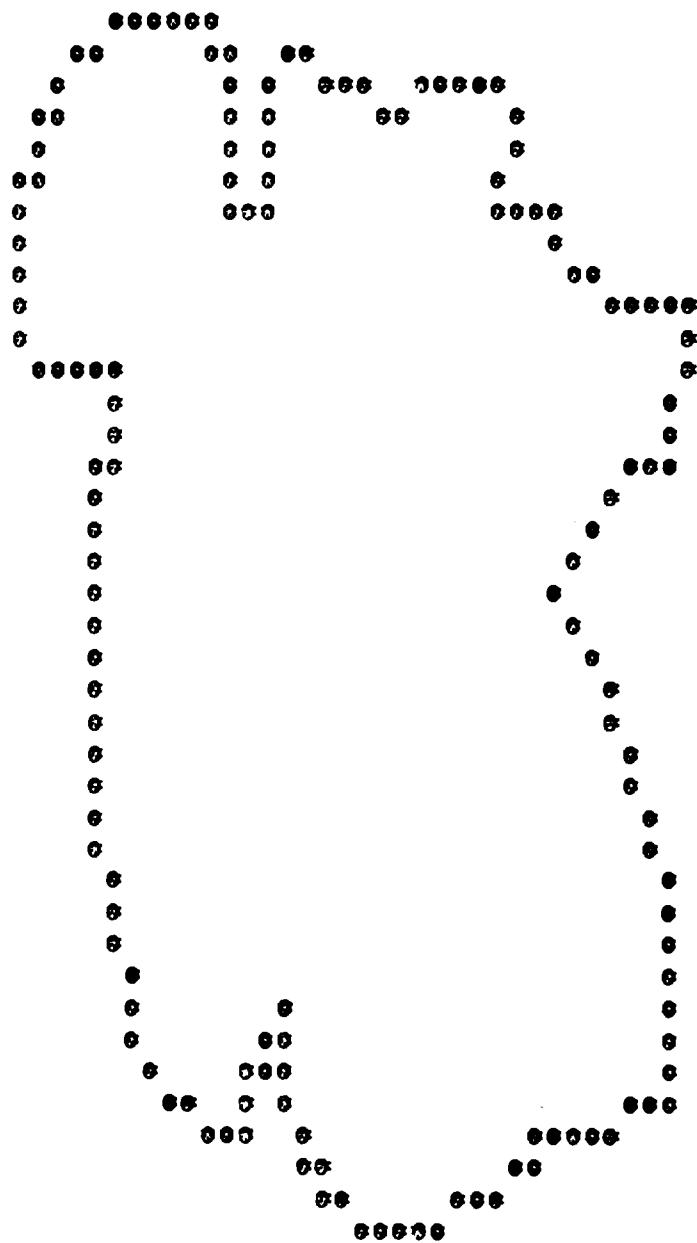
VARGUS 10A PATTERN 3



RESULT OF CONG ON VARGUS 10-A PATTERN 2 AND PATTERN 3 WITH IBLUR = 0
CONG RETURNS 322.63

RESULT OF CONG AS ABOVE WITH IBLUR = 1
CONG RETURNS 77.34

VARGUS 10A PATTERN 4



RESULT OF CONG ON VARGUS 10-A PATTERN 2 AND PATTERN 4 WITH IBLUR = 0
CONG RETURNS 353.69

RESULT OF CONG AS ABOVE WITH IBLUR = 1
CONG RETURNS 77.62

I. Identification

1. Subroutine Name: BLUR
2. Category: Pattern Manipulation
3. Purpose: To accept a binary matrix and simulate all possible 3-step random walks for every "black" cell. Thus a weight or loading is built up on a scale from 0 to 9 representing the probability that a given cell might be black.
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Pattern matrix is a 48 x 48 fixed point matrix.
4. Subroutine Usage: CALL BLUR (MX, NI, NJ).
5. Subroutine Parameters:
 - MX: pattern matrix.
 - NI: number of rows.
 - NJ: number of columns.
6. Subroutines Required:
 - RESCAL: Rescales a matrix such that values are whole numbers from 0 to 9.

III. Mathematical Methods and References

None.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

RSCAL
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.580 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
  SUBROUTINE BLUR(MX,NI, NJ)
C
  REAL IMX (54, 54)
  DIMENSION MX (48, 48), IWALK (49)
  COMMON IMX
  DATA IWALK/1,3,6,7,6,3,1,3,6,10,10,10,6,3,6,10,25,27,25,10,6,7,10,
127,30,27,10,7,6,10,25,27,25,10,6,3,6,10,10,10,6,3,1,3,6,7,6,3,1/
C
  NJJ=NJ+3
  NII=NI+3
  IN=NI+6
  JN=NJ+6
C
C   SET WORK MATRIX TO ZERO
  DO 20 I=1,IN
  DO 20 J=1,JN
20  IMX(I,J)=0
C
C   CHECK INPUT MATRIX FOR SIGNAL
  DO 50 I=1,NI
  DO 50 J=1,NJ
  IF (MX(I,J))50,50,30
30  IOVER=I+6
  JDOWN=J+6
  KOUNT=1
  DO 40 II=I,IOVER
  DO 40 JJ=J,JDOWN
  IMX(II,JJ)=IMX(II,JJ)+IWALK(KOUNT)
40  KOUNT=KOUNT+1
50  CONTINUE
C
C
C   SET BORDER AROUND MATRIX TO ZERO
  DO 100 K=4,NJJ
  IMX(4,K)=0
100  IMX(NII,K)=0
  DO 105 K=4,NII
  IMX(K,4)=0
105  IMX(K,NJJ)=0
  DO 110 K=4,NII
  DO 110 J=4,NJJ
110  MX(K-3,J-3)=IMX(K,J)
  CALL RSCAL (1, IMX, MX, NI, NJ)
  RETURN
  END

```

VARIABLE ALLOCATIONS

IMX(RC)=FFFE-E938	IWALK(I)=0032-0002	NJJ(I)=0033	NII(I)=0034
-------------------	---------------------	--------------	--------------

I(I)=0037
JJ(I)=003D

J(I)=0038
K(I)=003E

IOVER(I)=0039

JDOWN(I)=003A

STATEMENT ALLOCATIONS

20 =0078 30 =00A9 40 =00DE 50 =00F6 100 =0116 105 =0139 11

FEATURES SUPPORTED

NONPROCESS

ONE WORD INTEGERS

CALLED SUBPROGRAMS

RSCAL FADD FLDX FSTO FSTOX IFIX FLOAT ISTOX SUBSC SU

INTEGER CONSTANTS

3=0042 6=0043 1=0044 0=0045 4=0046

CORE REQUIREMENTS FOR BLUP

COMMON 5832 INSKEL COMMON 0

VARIABLES 66 PROGRAM 324

END OF COMPILATION

I. Identification

1. Subroutine Name: RSCAL
2. Category: Computational
3. Purpose: To rescale a matrix such that each cell contains a whole number in the interval 0 to 9.
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Matrix dimensions up to 48 x 48.
Input matrix is either fixed or floating point. Output matrix is always fixed point.
4. Subroutine Usage: CALL RSCAL (MODE, FMX, MX, NI, NJ).
5. Subroutine Parameters:
 - MODE: Indicates whether data matrix is fixed point (MODE = 1), or floating point (MODE = 2).
 - FMX: Floating-point input matrix.
 - MX: Fixed-point input matrix. Returns rescaled matrix.
 - NI: Number of rows.
 - NJ: Number of columns.

6. Subroutines Required:

MIMAX: Find the maximum and minimum values
in a matrix.

III. Mathematical Method and References

Simple scaling procedure.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

CON
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.573 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
    SUBROUTINE RSCAL(MODE,FMX,MX,NI,NJ)
    DIMENSION FMX(54,54),MX(48,48)
    CALL MIMAX(MODE,FMX,MX,NI,NJ,BIG,SMALL)
    DIVID=(BIG-SMALL)/9.9
    DO 5 I=1,NI
    DO 5 J=1,NJ
    TEMP=(FMX(I,J)-SMALL)/DIVID
5  MX(I,J)=TEMP
    RETURN
    END

VARIABLE ALLOCATIONS
    BIG(R )=0000      SMALL(R )=0002      DIVID(R )=0004      TEMP(R )=0006

STATEMENT ALLOCATIONS
5      =0048

FEATURES SUPPORTED
    NONPROCESS
    ONE WORD INTEGERS

CALLED SUBPROGRAMS
    MIMAX  FSUB  FDIV  FLD  FLDX  FSTO  IFIX  ISTOX  SUBSC  SU

REAL CONSTANTS
    .990000E 01=000C

INTEGER CONSTANTS
    1=000E

CORE REQUIREMENTS FOR RSCAL
    COMMON      0  INSKEL COMMON      0
    VARIABLES   12  PROGRAM      92

END OF COMPILATION

```

I. Identification

1. Subroutine Name: MIMAX
2. Category: Computational
3. Purpose: Find the maximum and minimum values
in a matrix.
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Matrix dimensions up to 48 x 48.
4. Subroutine Usage: CALL MIMAX (MODE, FMX, MX, NI,
NJ, BIG, SMALL).
5. Subroutine Parameters:
 - MODE: Indicates whether data matrix is fixed
point (MODE = 1), or floating point
(MODE = 2).
 - FMX: Floating-point input matrix.
 - MX: Fixed-point input matrix.
 - NI: Number of rows.
 - NJ: Number of columns.
 - BIG: Return largest value in matrix to
calling program.
 - SMALL: Returns smallest value in matrix to
calling program.

6. Subroutines Required: None.

III. Mathematical Methods and References

None.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

BLUR
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.590 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
    SUBROUTINE MIMAX(MODE,FMX,MX,NI,NJ,BIG,SMALL)
    DIMENSION FMX(54,54),MX(48,48)
    GO TO (5,10),MODE
    5 DO 7 I=1,NI
      DO 7 J=1,NJ
        7 FMX(I,J)=MX(I,J)
    10 BIG=FMX(1,1)
    SMALL=BIG
    DO 15 I=1,NI
      DO 15 J=1,NJ
        IF(BIG-FMX(I,J)) 11,15,13
    11 BIG=FMX(I,J)
    GO TO 15
    13 IF(SMALL-FMX(I,J)) 15,15,14
    14 SMALL=FMX(I,J)
    15 CONTINUE
    RETURN
    END

VARIABLE ALLOCATIONS
    I(I )=0000          J(I )=0001

STATEMENT ALLOCATIONS
    5      =0028  7      =0030  10     =0057  11     =0079  13     =0086  14     =0094  15

FEATURES SUPPORTED
    NONPROCESS
    ONE WORD INTEGERS

CALLED SUBPROGRAMS
    FSUBX  FLD    FLDX    FSTO    FSTOX    FLOAT    COMGO    LDFAC    SUBSC    SU

INTEGER CONSTANTS
    1=0004

CORE REQUIREMENTS FOR MIMAX
    COMMON      0  INSKEL COMMON      0
    VARIABLES    4  PROGRAM      176

END OF COMPILATION

```

I. Identification

1. Subroutine Name: SØLID
2. Category: Pattern Manipulation
3. Purpose: Accepts a closed figure represented in matrix form and fills in the interior.
4. Date: July, 1969
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: The program is designed to accept a 48 x 48 matrix. The size of IARY and JARY may be reduced to save space but at a cost in running time.
4. Subroutine Usage: CALL SØLID (INARY).
5. Subroutine Parameters:

INARY: a 48 x 48 input matrix containing the pattern. This same matrix returns the solid figure so that the original pattern is destroyed.
6. Subroutines Required: None.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

WIDER
DMP FUNCTION COMPLETED
// FOR SOLID 30 JUN 71 13.893 HRS
*ONE WORD INTEGERS
*LIST ALL
*NON PROCESS PROGRAM
  SUBROUTINE SOLID(INARY)
    DIMENSION INARY(48,48), ARY(54,54), IARY(54), JARY(54)
    COMMON ARY
C
    DO 10 I=1,48
    DO 10 J=1,48
10  ARY(I+2,J+2)= INARY(I,J)
C    THIS PUTS THE ORIGINAL ARRAY IN THE WORKING ARRAY
    M1=52
    M2=51
    M3=50
C
    DO 11 I=1,M1
    ARY(1,I)=-1
    ARY(M1,I)=-1
    ARY(I,1)=-1
11  ARY(I,M1)=-1
C    THIS PUTS -1 AROUND THE BORDER
    DO 15 I=2,M2
    ARY(2,I)=0
    ARY(M2,I)=0
    ARY(I,2)=0
15  ARY(I,M2)=0
C
    IP1=1
    IP2=2
    IARY(1)=2
    JARY(1)=2
    ARY(2,2)= -1.
C    THIS SECTION FILLS IN -1 AROUND THE FIGURE
35  K1=IARY(IP1)
    K2 = JARY(IP1)
C
    IF (ARY(K1+1,K2 )) 50,45,50
45  ARY(K1+1,K2 )=-1.
    IARY(IP2)=K1+1
    JARY(IP2)=K2
    IP2=IP2+1
50  IF (ARY(K1 ,K2-1)) 60,55,60
55  ARY(K1 ,K2-1)=-1.
    IARY(IP2)=K1
    JARY(IP2)=K2-1
    IP2=IP2+1
60  IF (ARY(K1 ,K2+1)) 70,65,70
65  ARY(K1 ,K2+1)=-1.
    IARY(IP2)=K1
    JARY(IP2)=K2+1
    IP2=IP2+1

```

```

70 IF (ARY(K1-1,K2 )) 80,75,80
75 ARY(K1-1,K2 )=-1.
   IARY(IP2)=K1-1
   JARY(IP2)=K2
   IP2=IP2+1
80 IP1=IP1+1
   IF (IP1-IP2) 85,110,85
85 IF (IP2+4-54) 95,90,90
90 IARY(IP2)=-1
   IP2=1
95 IF (IARY(IP1)) 100,100,35
100 IP1=1
   GO TO 35
110 DO 200 I=3,M3
   DO 200 J=3,M3
   IF (ARY(I,J)) 200,150,150
150 INARY(I-2,J-2)=1
200 CONTINUE
   RETURN
   END

```

VARIABLE ALLOCATIONS

```

ARY(RC)=FFFE-E938   IARY(I )=0035-0000   JARY(I )=0068-0036   I(I )=006C
M2(I )=006F         M3(I )=0070         IP1(I )=0071         IP2(I )=0072

```

STATEMENT ALLOCATIONS

```

10  =0094  11  =00EE  15  =0128  35  =0158  45  =0173  50  =0196  55
75  =0200  80  =0223  85  =022F  90  =0237  95  =0245  100 =024E  110

```

FEATURES SUPPORTED

```

NONPROCESS
ONE WORD INTEGERS

```

CALLED SUBPROGRAMS

```

FLD      FLDX      FSTOX      FLOAT      ISTOX      LDFAC      SUBSC      SNR      SUBIN

```

REAL CONSTANTS

```

.100000E 01=007A

```

INTEGER CONSTANTS

```

1=007C      48=007D      52=007E      51=007F      50=0080      2=0081

```

CORE REQUIREMENTS FOR SOLID

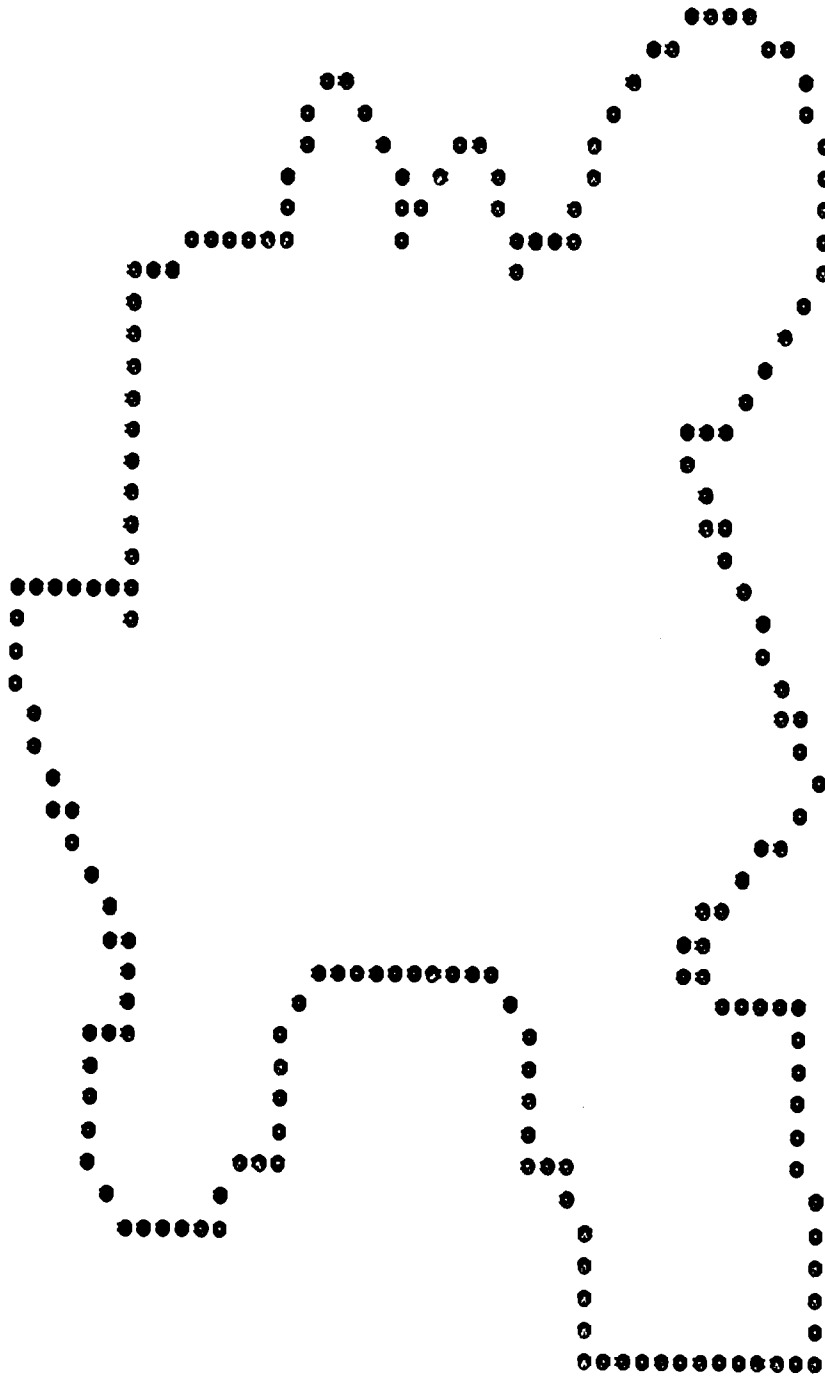
```

COMMON 5832 INSKEL COMMON 0
VARIABLES 122 PROGRAM 526

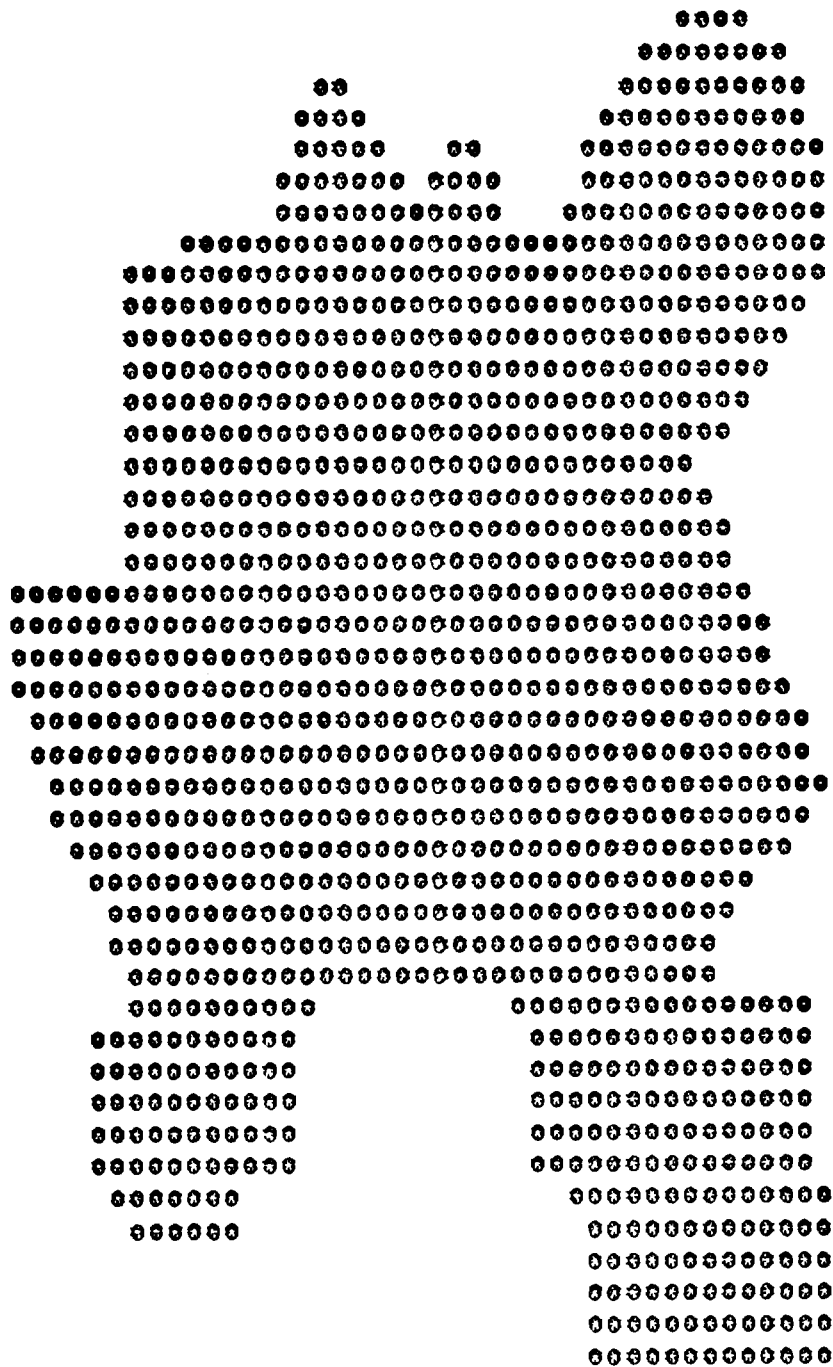
```

END OF COMPILATION

BASIC VARGUS 10A PATTERN



EFFECT OF SOLID ON PATTERN



I. Identification

1. Subroutine Name: WIDEN
2. Category: Pattern Manipulation
3. Purpose: To widen lines of a figure centered in a matrix.
4. Date: September, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitation: Widens line one cell thick to 3 cells thick. Multiple calls to the subroutine will result in still thicker lines.
Matrix should be dimensioned to 48 x 48.
4. Subroutine Usage: CALL WIDEN (MX, NI, NJ).
5. Subroutine Parameters:
MX: The matrix containing the pattern.
NI: The number of rows.
NJ: The number of columns.
6. Subroutines Required: None.

III. Mathematical Method and References

Each positive cell of the matrix is treated as the center of a 3 x 3 submatrix and its eight surrounding cells are set equal to 1.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

```

U
DMP FUNCTION COMPLETED
// FOR WIDEN 30 JUN 71 13.879 HRS
*LIST ALL
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
      SUBROUTINE WIDEN (MX, NI, NJ)
C      ACCEPTS MATRIX MX (NI ROWS, NJ COLUMNS)
C      AND PLACES 1 S IN EVERY POSITION OF A 3 X 3 SUBMATRIX OF MX
C      WHICH IS CENTERED SUCCESSIVELY ON POSITIVE ENTRIES OF MX
      DIMENSION MX (48, 48), LX (48, 48)
      DO 5 I=1,NI
      DO 5 J=1,NJ
5      LX(I,J)=0
      DO 30 I = 1, NI
      DO 30 J = 1, NJ
      IF (MX (I, J)) 30, 30, 10
10 DO 20 K = 1, 3
      DO 20 L = 1, 3
      M = I + K - 2
      N = J + L - 2
20 LX (M, N) = 1
30 CONTINUE
      DO 40 I = 1, NI
      DO 40 J = 1, NJ
      MX (I, J) = LX (I, J)
40 CONTINUE
      RETURN
      END

VARIABLE ALLOCATIONS
      LX(I )=08FF-0000      I(I )=0900      J(I )=0901      K(I )=0902
      N(I )=0905

STATEMENT ALLOCATIONS
5      =0922 10      =0952 20      =096A 30      =0987 40      =09AC

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

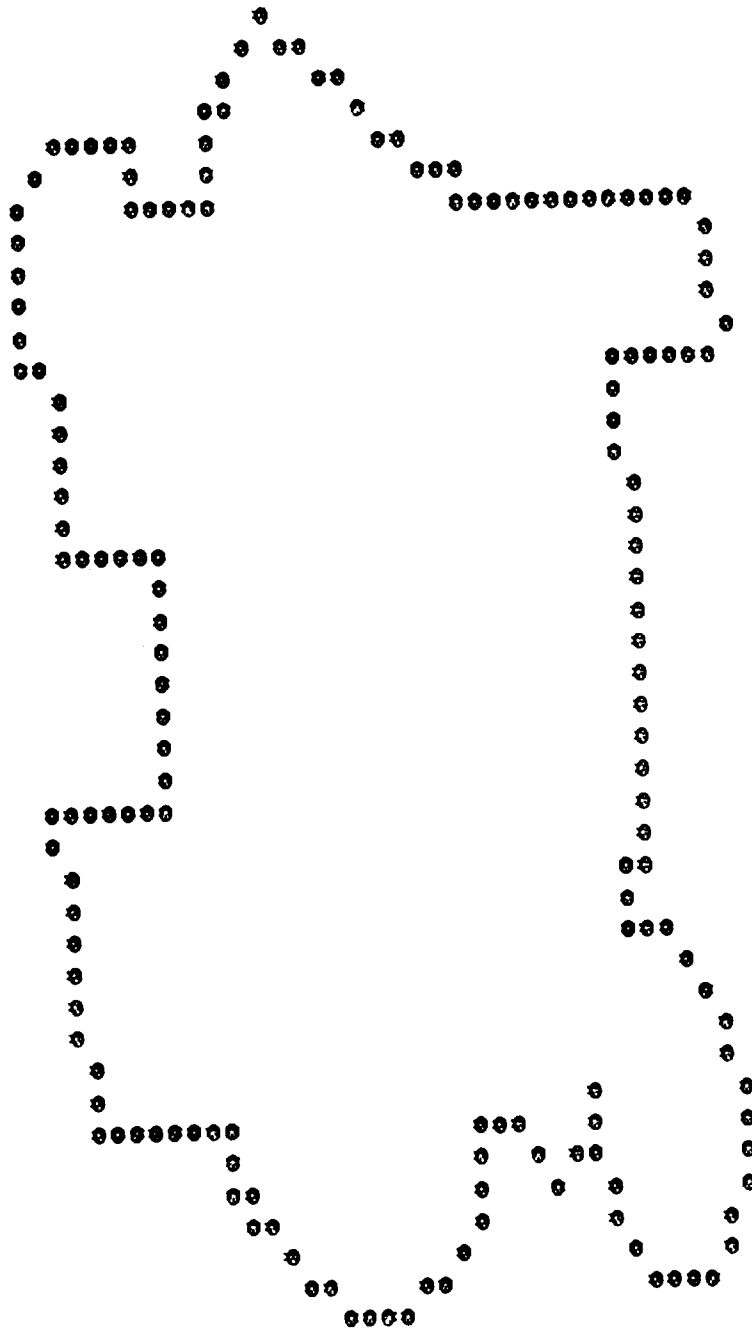
CALLED SUBPROGRAMS
ISTDX SUBSC SUBIN

INTEGER CONSTANTS
      1=0908      0=0909      3=090A      2=090B

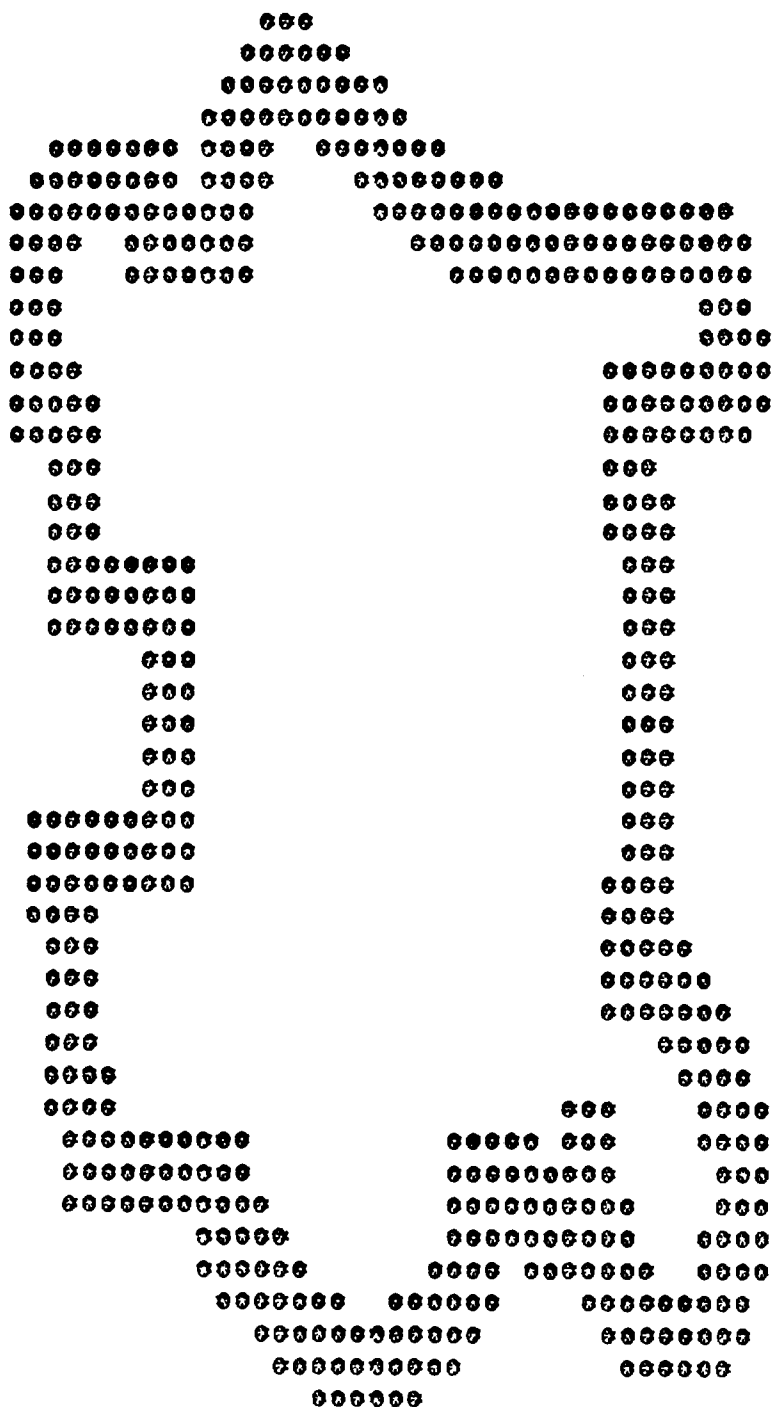
CORE REQUIREMENTS FOR WIDEN
COMMON      0 INSKEL COMMON      0
VARIABLES 2312 PROGRAM 184

```

BASIC VARGUS 10A PATTERN



EFFECT OF WIDEN ON PATTERN



I. Identification

1. Subroutine Name: WIDER
2. Category: Pattern Manipulation
3. Purpose: To widen lines of a closed figure inwardly and retain the original figure as the outer perimeter.
4. Date: September, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: Widens to 3 cells thickness. Matrix should be dimensioned to 48 x 48.
4. Subroutine Usage: CALL WIDER (MX, NI, NJ).
5. Subroutine Parameters:
 - MX: pattern matrix.
 - NI: number of rows.
 - NJ: number of columns.
6. Subroutines Required:
 - SØLID: Fills in closed figure.
 - WIDEN: Thickens lines to width 3.

III. Mathematical Methods and References

None.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

WIDEN
 DMP FUNCTION COMPLETED
 // FOR WIDER 30 JUN 71 13.887 HRS
 *NONPROCESS PROGRAM
 *LIST ALL
 *ONE WORD INTEGERS

```

    SUBROUTINE WIDER (MX,NI,NJ)
    DIMENSION MX(48,48),LX(48,48)
    DO 5 I=1,NI
    DO 5 J=1,NJ
  5  LX(I,J)=MX(I,J)
    CALL SOLID(LX)
    CALL WIDEN(MX,NI,NJ)
    CALL WIDEN(MX,NI,NJ)
    DO 6 I=1,NI
    DO 6 J=1,NJ
  6  MX(I,J)=MX(I,J)*LX(I,J)
    RETURN
  END

```

VARIABLE ALLOCATIONS
 LX(I)=08FF-0000 I(I)=0900 J(I)=0901

STATEMENT ALLOCATIONS
 5 =0920 6 =0952

FEATURES SUPPORTED
 NONPROCESS
 ONE WORD INTEGERS

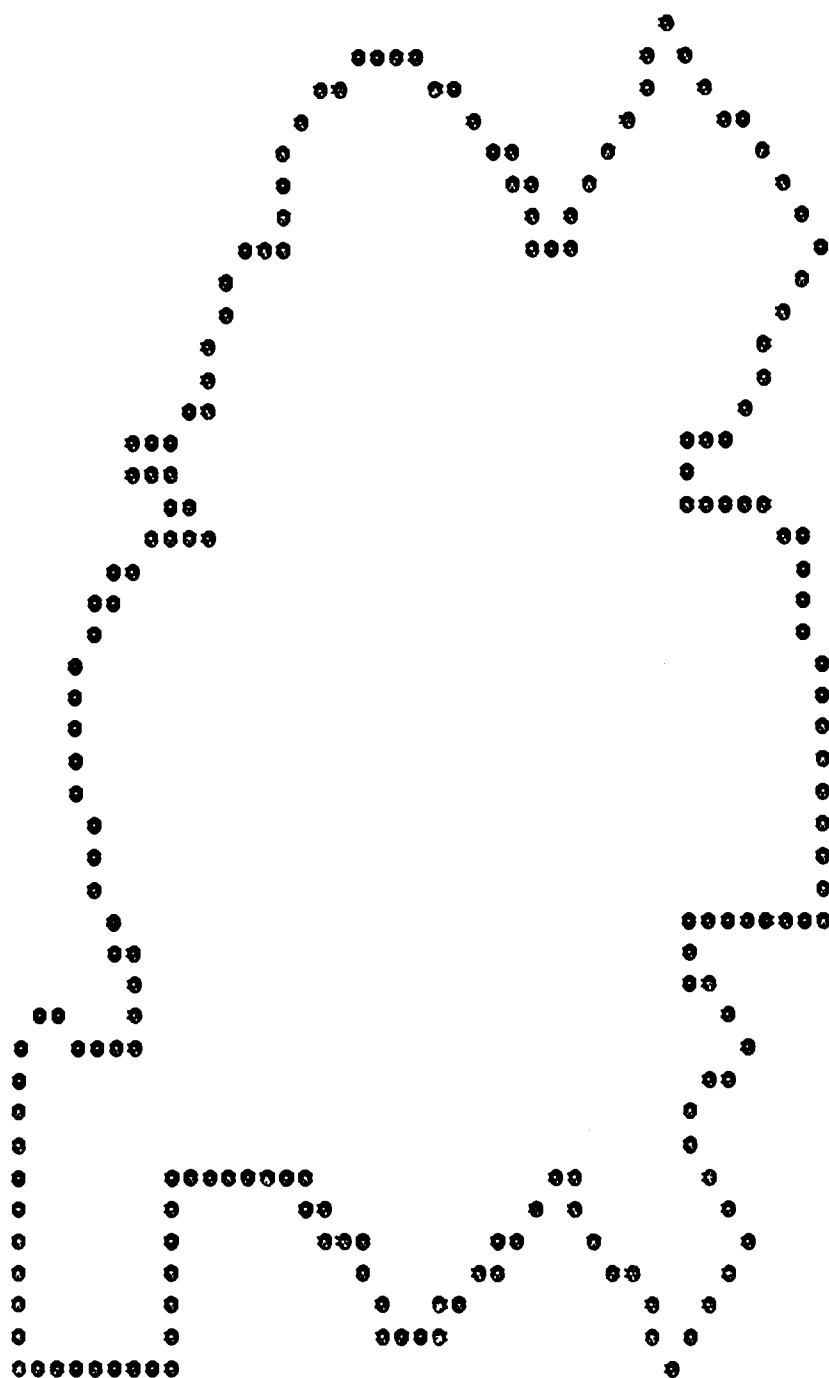
CALLED SUBPROGRAMS
 SOLID WIDEN ISTOX SUBSC SUBIN

INTEGER CONSTANTS
 I=0904

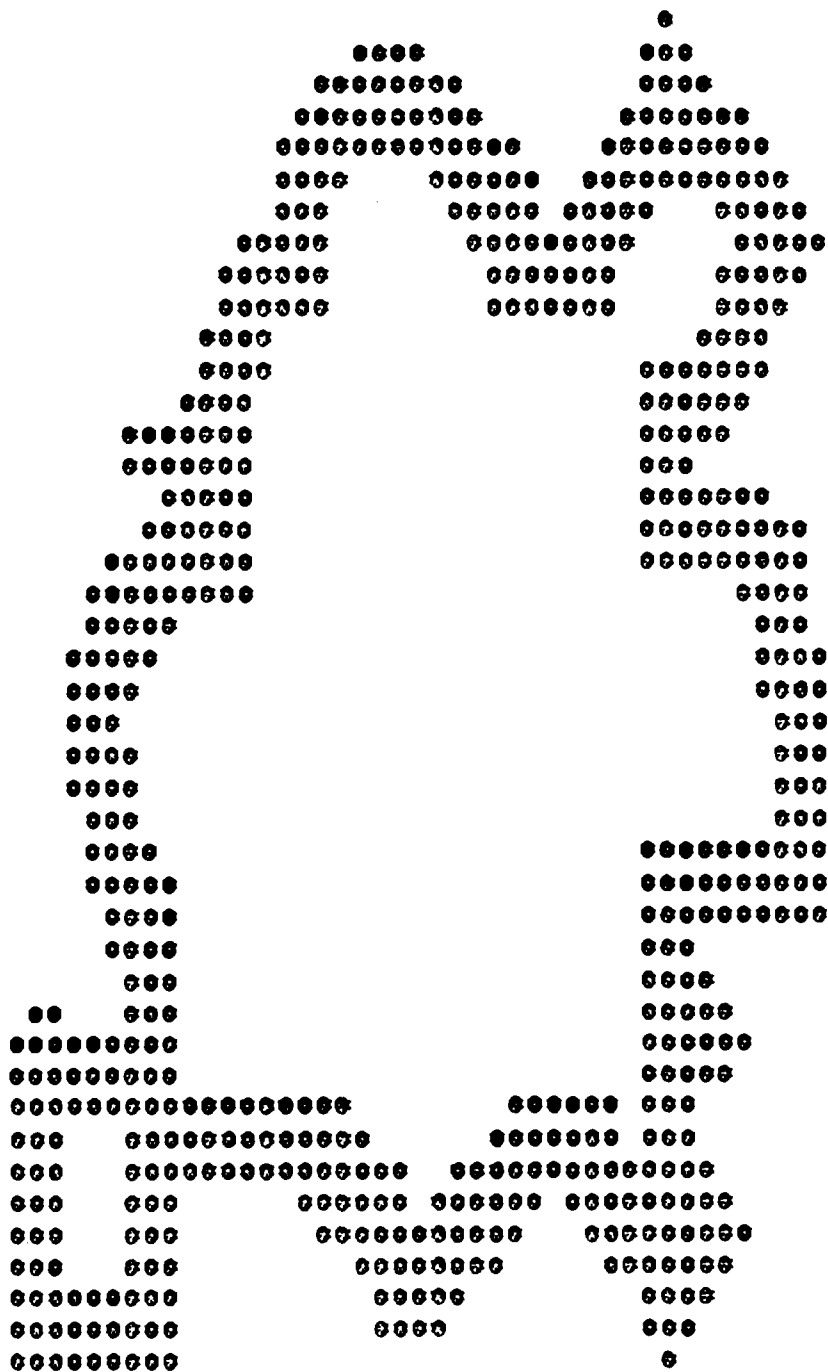
CORE REQUIREMENTS FOR WIDER
 COMMON 0 INSKEL COMMON 0
 VARIABLES 2308 PROGRAM 112

END OF COMPILATION

BASIC VARGUS 10A PATTERN



EFFECT OF WIDER ON PATTERN



I. Identification

1. Subroutine Name: RANDY
2. Category: Pseudo-random Number Generator
3. Purpose: Compute uniformly distributed random floating-point numbers between 0.0 and 1.0.
4. Date: March 25, 1968
5. Programmer: Bill Randol Brown
Statistical tests in Remarks Section performed by Mike Abbamonte, April 20, 1971.

II. Use Information

1. Language: IBM Basic Fortran IV.
2. Machine: IBM 1800, 32K, 16-bit words.
3. Limitations: None.
4. Subroutine Usage: CALL RANDY (R).
5. Input Card: START card.
 - a. The START card contains 68 random digits in columns 1 to 68. Within these 68 digits strings of zeros in excess of two should not occur.
 - b. In columns 69-72 of the card should be four digits sampled at random and with replacement from 1, 3, 7 and 9.
 - c. This START card is read in by the subroutine itself. This read occurs only the first time

the subroutine is called. It is recommended that RANDY be called once early in execution so that the READ statement will be executed at a predictable time and thereby data synchronization problems avoided.

6. Output: The value returned (R) will be a floating-point number between 0.0 and 1.0.

7. Subroutines required: None.

III. Mathematical Methods and References

- a. Seventeen generators each operating under the power residue method are in RANDY. The random number generated on call i is used to select which of the seventeen generators will be used on call i+1.
- b. The cycle of each generator is 8192.
- c. Also see: Evans, S. H. The generation of pseudo-random numbers for the IBM 1800. Working paper #1. Institute for the Study of Cognitive Systems, Texas Christian University, 1968.

IV. Listing of Source Program

Attached.

V. Sample Output

Attached.

VI. Remarks

Using four randomly selected START cards, four Chi square tests were performed, each involving the

generation of 20,000 random numbers and 128 degrees of freedom

Test #1. Chi square/df = .900 ; 22nd percentile

Test #2. Chi square/df = .725 ; 1st percentile

Test #3. Chi square/df = .937 ; 30th percentile

Test #4. Chi square/df = .917 ; 37th percentile

It should be noted that these numbers tend to fall on the low side of the Chi square distribution indicating that the numbers generated by RANDY tend to be more evenly distributed in the unit interval that you would expect when randomly sampling from a rectangular distribution.

RANDY was also tested for cycling and none was found in the first 100,000 numbers. The criterion for cycling was repetition of the second through the eleventh numbers.

```

RAN
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13.478 HRS
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*LIST ALL
  SUBROUTINE RANDY(RR)
    DIMENSION KU(17),KC(17),KL(4)
    DATA NDUM, KC /1,899,835,957,877,915,861,947,925,867,909,883,
1851,941,893,931,845,963/
    GO TO (15,16),NDUM
15 READ(2,19)(KU(I),I=1,17),(KL(J),J=1,4)
19 FORMAT (17I4,4I1)
    J=0
    DO 13 I=1,17
      J=J+1
      KU(I)=KU(I)*10+KL(J)
      IF(KU(I))21,22,22
21 KU(I)=KU(I)+32767+1
22 IF(J-4)13,11,11
11 J=0
13 CONTINUE
    R=KU(1)
    R=R/32768.
    NDUM=NDUM+1
16 K=R*16.99+1.0
    IR=KU(K)*KC(K)
    IF(IR)23,24,24
23 IR=IR+32767+1
24 R=IR
    R=R/32768.
    KU(K)=IR
    RR=R
    RETURN
  END

VARIABLE ALLOCATIONS
  R(R )=0000      KU(I )=0012-0002      KC(I )=0023-0013      KL(I )=0027
  J(I )=002A      K(I )=002B      IR(I )=002C

STATEMENT ALLOCATIONS
19  =003D  15  =004D  21  =00A0  22  =00AD  11  =00B3  13  =00B7  16

FEATURES SUPPORTED
  NONPROCESS
  ONE WORD INTEGERS

CALLED SUBPROGRAMS
  FADD  FMPY  FDIV  FLD  FSTO  IFIX  FLOAT  COMGO  ISTOX  MRI

REAL CONSTANTS
  .327680E 05=0030  .169900E 02=0032  .100000E 01=0034

INTEGER CONSTANTS

```

2=0036

1=0037

17=0038

4=0039

0=003A

10=003B 32

CORE REQUIREMENTS FOR RANDY

COMMON 0 INSKEL COMMON 0

VARIABLES 48 PROGRAM 222

END OF COMPILATION

VALUES RETURNED FROM RANDY

0.46206670
0.97213757
0.60293591
0.33059698
0.09817506
0.82565319
0.71530163
0.06094361
0.42031866
0.22714236
0.80990612
0.48190313
0.71408092
0.34793096
0.52871716
0.10806275
0.96603405
0.62716686
0.91708385
0.30319219
0.22537234
0.28756719
0.87167370
0.94570934
0.96157848
0.99996960
0.97061169
0.69894421
0.26358038
0.58132946
0.74215710
0.40298467
0.22409060
0.19638064
0.22579959
0.02621460
0.07809449
0.63833630
0.78494274
0.33944708

I. Identification

1. Subroutine Name: PATØT
2. Category: Pattern Construction
3. Purpose: Output a 48 x 48 matrix in two symbols.
4. Date: July, 1968
5. Programmers: S. H. Evans, Bob Breckenridge.

II. Use Information

1. Language: 1800 Basic FØRTRAN IV.
2. Machine: IBM 1800, 32K, 16 bit words.
3. Limitations: Accepts fixed point 48 x 48 matrix.
Positive numbers are output by overprinting a zero by an asterisk. Zero or negative numbers are output as a blank space.
4. Subroutine Usage: CALL PATØT (MX, NI, NJ)
5. Subroutine Parameters:
MX: Matrix to be output. Must be dimensioned to 48 x 48.
NI: Number of rows to output.
NJ: Number of columns to output.
6. Subroutines Required: None.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.

JOIN

DMP FUNCTION COMPLETED

// FOR 30 JUN 71 13.732 HRS

*LIST ALL

*ONE WORD INTEGERS

*NONPROCESS PROGRAM

```
      SUBROUTINE PATOT(MX,NI,NJ)
C     ACCEPTS MATRIX MX(NI ROWS, NJ COLUMNS)
C     AND PRINTS ALL POSITIVE ENTRIES AS BLACK SPOTS,
C     BY OVER PRINTING CHA AND CHB FOR BLACK AND CHC FOR WHITE.
      DIMENSION IX(48,48), T(48), S(48)
      DATA CHA,CHB,CHC/'0','*',' '/
      DO 100 I=1, NI
      DO 50 J=1, NJ
      IF(MX(I,J))30, 30, 40
30    T(J)=CHC
      S(J)=CHC
      GO TO 50
40    T(J)=CHA
      S(J)=CHB
50    CONTINUE
      WRITE (3,51)(T(J), J=1, NJ)
51    FORMAT (1H 48A1)
      WRITE (3,52)(S(J), J=1, NJ)
52    FORMAT (1H+48A1)
100   CONTINUE
      RETURN
      END
```

VARIABLE ALLOCATIONS

T(R)=005E-0000

S(R)=008E-0060

CHC(R)=00C0

CHA(R)=00C2

J(I)=00C7

STATEMENT ALLOCATIONS

51 =00CC 52 =00D1 30 =00F4 40 =0105 50 =0114 100 =014B

FEATURES SUPPORTED

NONPROCESS

ONE WORD INTEGERS

CALLED SUBPROGRAMS

FLO FSTUX MWRT MCOMP MIOFX SUBSC SUBIN

INTEGER CONSTANTS

1=00CA

3=00CB

CORE REQUIREMENTS FOR PATOT

COMMON 0 INSKEL COMMON 0

VARIABLES 202 PROGRAM 140

END OF COMPILATION

DISTRIBUTION LIST

CG, USAMC, Wash, D. C.	CO, USACDC Med Svc Agency	CO, Harry Diamond Labs	
AMCDL (Ofc of Dep for Labs)	1 Fort Sam Houston, Texas	Washington, D. C.	
AMCRD (Air Def & Msl Ofc)	1	AMXDO-EDC (B. I. Green)	1
AMCRD (Air Mobility Ofc)	1 CO, USACDC Military Police Agency		
AMCRD (Comm-Elec Ofc)	1 Fort Gordon, Georgia	1 CO, USA Mobility Equip R&D Ctr	
AMCRD-G	1	Fort Belvoir, Va.	
AMCRD (Weapons Ofc)	1 CO, USACDC Supply Agency	Human Factors Engr.	1
AMCRD (Dr. Kaufman)	1 Fort Lee, Va.		
AMCRD (Mr. Crellin)	1	CO, Engr. Topographic Labs.	
	USACDC Experimentation Command	Fort Belvoir, Va.	
Ofc of Chief of Staff, DA, Wash, D. C.	Fort Ord, Calif.	Mr. Sidney Presser	1
CSAVCS-W-TIS	1 Tech Library, Box 22		
		U. S. Army Natick Laboratories	
USA Behavioral Science Rsch Lab	Human Factors Division	Natick, Mass.	
Arlington, Va.	1 G-2/3, USACDCEC	AMSRE-STL, Rsch Library	1
	Fort Ord, Calif.	AMXRE-PRB	1
Dr. J. E. Uhlaner, Dir.		AMXRE-PRBN	1
USA Behavioral Science Rsch Lab	Dir of Graduate Studies & Rsch	AMXRE-PRBE	1
Arlington, Va.	1 USA Command & Gen Staff College		
	Fort Leavenworth, Kansas	Comdt, Army Log Mgmt Ctr	
Behavioral Sciences Division	Behavioral Sciences Rep.	1 Fort Lee, Va.	
Ofc, Chief of R&D, DA		E. F. Neff, Proc Div.	1
Washington, D. C.	1 CO, USA Environ Hygiene Agency		
	Edgewood Arsenal, Md.	USA Gen Equip Test Activity	
Deputy Chief of Staff for Personnel	Librarian, Bldg 2400	2 Methods Engr Dir, HF Div.	
DA, Washington, D. C.		Fort Lee, Va.	1
Personnel Rsch Div.	1 Human Factors Br, Med Rsch Lab		
	Rsch Labs, Edgewood Ars, Md.	1 CG, US CONARC	
CG, USACDC, Fort Belvoir, Va.		Fort Monroe, Va.	1
CDCCD-C	1 CO, USA Edgewood Arsenal, Md.	ATTI-RD-RD	1
CDCMR	1 Psychology Branch		
CDCRE	1	CO, USA Rsch Ofc, Box CM	
	CO, Frankford Arsenal, Phila, Pa.	Duke Station, Durham, N.C.	1
CO, USACDC Air Defense Agency	SMUFA-N/6400/202-4 (HF)		
Fort Bliss, Texas	1 Library (C2500, B1 51-2)	1 Dir Rsch, USA Avn HRU	
		PO Box 428, Fort Rucker, Ala.	
CO, USACDC Armor Agency	CO, Picatinny Ars, Dover, N.J.	Librarian	1
Fort Knox, Kentucky	1 SMUPA-VC1 (Dr. Strauss)		
		USA Bd for Avn Acdt Rsch Lab	
CO, USACDC Artillery Agency	CG, USA Electronics Command	Fort Rucker, Ala.	
Fort Sill, Okla	1 Fort Monmouth, N.J.	Gail Bankston, Bldg 5504	1
	AMSEL-RD-GDA		
CO, USACDC Aviation Agency		1 CG, USASCOM, PO Box 209	
Fort Rucker, Alabama	1 Dir, Military Psychol & Ldrship	St. Louis, Missouri	
	USMA, West Point, N.Y.	1 AMSAV-R-F (S. Moreland)	1
CO, USACDC CBR Agency			
Fort McClellan, Alabama	1 CO, Watervliet Arsenal, N.Y.	CG, USA Missile Command	
	SWEWV-RDT	1 Redstone Arsenal, Alabama	
CG, USACDC Combat Arms Group		AMSMI-RBLD	1
Fort Leavenworth, Kansas	1 CO, USA Med Equip R&D Lab	AMSMI-RLH (Chaikin)	1
	Fort Totten, Flushing, L.I., N.Y.		
CG, USACDC Combat Svc Spt Group		President, USA Infantry Board	
Fort Lee, Va.	1 CO, USA Rsch Inst of Envir Med	Fort Benning, Georgia	1
	Natick, Mass.		
CO, USACDC Comm-Elec Agency	MEDRI-CL (Dr. Dusek)	1 President, USA Maintenance Bd.	
Fort Monmouth, N. J.		Fort Knox, Kentucky	
	CG, USA Medical R&D Command	Adjutant	1
CO, USACDC Engineer Agency	Main Navy Bldg, Wash, D. C.		
Fort Belvoir, Va.	1 Behavioral Sciences Rsch Br	1 USA Armor, HRU, Fort Knox, Ky.	
		Library	1
CO, USACDC	Dir, Walter Reed Army Inst Rsch		
Institute of Strategic & Stability Opns	Washington, D. C.	CO, USA Med Rsch Lab	
Fort Bragg, N.C.	1 Neuropsychiatry Division	1 Fort Knox, Kentucky	1

CG, USA Weapons Command, RI, Ill.		Dir, Naval Research Laboratory	Rsch Sec, Psychology Service
AMSWE-RDT	1	Washington, D. C.	VA Hospital, Irving Ave & Univ Pl
AMSWE-SMM-P	1	Code 5143A	1 Syracuse, N.Y.
SWERI-RDD-PD	2		Dr. Harvey A. Taub
		Code 455 Ofc of Naval Research	
CG, USA Tank-Automotive Command		Washington, D. C.	2 Federal Aviation Administration
Warren, Michigan		Engr Psychol (Dr. Tolcott)	1 800 Independence Ave, S.W.
AMSTA-R	1		Washington, D. C.
AMSTA-RHFL	2	Code 458 Ofc of Naval Research	Admin Stds Div (MS-110)
AMSTA-RKAE	1	Washington, D. C.	
		Personnel & Tng (Dr. Farr)	1 USPO Dept, Bureau R&E, HF Br.
Cleveland Army Tank-Auto Plant			Washington, D. C.
Cleveland, Ohio			Mr. D. Cornog
HF Engr.	1		
			US Dept of Commerce, NTIS
Director of Research			Springfield, Va.
HumRRO Div. No. 5 (Air Defense)			
PO Box 6021, Fort Bliss, Texas	1	USN Submarine Med Ctr, Libr	Defense Documentation Center
		Box 600, USN Sub Base	Cameron Station, Alexandria, Va.
		Groton, Conn.	1
Comdt, USA Artillery & Msl School			Lib, George Washington Univ.
Fort Sill, Oklahoma		CO & Dir, Naval Tng Dev Ctr	HumRRO, Alexandria, Va.
USAAMS Tech Library	1	Orlando, Florida	
		Technical Library	1 American Institute for Rsch
CG, White Sands Msl Range, N.M.			8555 16th St, Silver Spring, Md.
Technical Library	1		Library
STEWs-TE-Q (Mr. Courtney)	1	US Navy Electronics Laboratory	
		San Diego, Calif.	1 American Institute for Rsch
CG, USA Elec Proving Ground		Ch, Human Factors Div.	1 135 North Bellefield, Pgh, Pa.
Fort Huachuca, Ariz.		Code 3400 (Wardell B. Welch)	Library
Mr. Abraham, Test Dir.	1		
		Dept of Operations Analysis	
CO, Ft Huachuca Spt Comd, US Army		Naval Postgraduate School	American Institute for Rsch
Fort Huachuca, Ariz.		Monterey, Calif.	PO Box 1113, Palo Alto, Calif.
Tech Ref Div	1	Prof. James K. Arima	1 Library
CO, Yuma Proving Ground		RADC (EMEDI)	Center for Rsch in Social Systems
Yuma, Ariz.		Griffiss AFB, N.Y.	1 American Institutes for Rsch
Technical Library	1		10605 Concord St, Kensington, Md.
		Hq, ESD (ESTI)	ISB
CO, USA Tropic Test Center		L. G. Hanscom Field	
PO Drwr 942, Fort Clayton, CZ		Bedford, Mass.	1 The Franklin Institute Rsch Labs
Behavioral Scientist	2		Phila, Pa.
		Wright-Patterson AFB, Ohio	Tech Reports Library
CO, USA Arctic Test Center		6570 AMRL (MRHE)	1
APO Seattle, Wash.		6570 AMRL (MRHE/Bates)	1 Institute for Defense Analyses
STEAC-IT	1	6570 AMRL (MRHE/Warrick)	1 Arlington, Va.
		AFFDL-FDCR (CDIC)	1 Dr. J. Orlansky
USA Standardization Group, UK			1 Serials Unit, Purdue University
Box 65, FPO New York		AMD (AMRH) Brooks AFB, Texas	Lafayette, Ind.
Rsch/Gen Materiel Rep.	1		
		Hq, 4442D Combat Crew Tng Wing	
USATECOM, Bldg 314, APG	1	(TAC) Little Rock AFB	Dept Psychol, Univ of Maryland
		Jacksonville, Ark.	1 College Park, Md.
USACDC Liaison Ofc, Bldg 314, APG	1		
		CO, USACDC Infantry Agency	Mr. R. K. Brome, Govt Pub Sec
CO, USACDCMA, Bldg 305, APG	1	Fort Benning, Ga.	JFK Memorial Library
		Central Files	1 Calif State College/Los Angeles
US Marine Liaison Ofc, Bldg 314	1		Los Angeles, Calif.
Tech Library, Bldg 305, APG	1	Civil Aeromedical Institute	Dr. R. G. Pearson
		Fed Avn Agency Aero Center	Dept of Ind Engineering
		PO Box 25082, Okla City, Okla.	North Carolina State Univ.
		Psychol Br, AC-118	1 Raleigh, N.C.

Dr. F. Loren Smith Dept Psychol, Univ of Delaware Newark, Delaware	1	Dr. Herbert J. Bauer GM Rsch Labs, GM Tech Ctr Warren, Mich.	1	Prof. Richard C. Dubes Michigan State University East Lansing, Michigan	1
Dr. H. W. Stoudt Harvard Univ, Boston, Mass.	1	Dr. Edwin Cohen Link Group, Gen Precision Sys Inc. Binghamton, New York	1	Dr. Bill R. Brown University of Louisville Louisville, Kentucky	1
Dr. Leonard Uhr Computer Sciences Dept Univ of Wisconsin Madison, Wisconsin	1	Mr. Henry E. Guttman Sandia Corp, Albuquerque, N.M.	1	Mr. John H. Duddy, Dept 62-40 Bldg 151, Lockheed, PO Box 504 Sunnyvale, Calif.	1
Dr. R. A. Wunderlich Psychol Dept, Catholic Univ. Washington, D. C.	1	Dr. M. I. Kurke Human Sciences Rsch Inc. McLean, Va.	1	Dr. Arthur S. Kamlet Bell Telephone Labs (1B-125) Whippany, N. J.	1
Psychological Abstracts 1200 17th Street, NW Washington, D. C.	1	Mr. James Moreland Westinghouse Elec Corp, R&D Ctr Pittsburgh, Pa.	1	US Dept of Commerce National Bureau of Standards Washington, D. C.	1
AC Electronics Div, GMC Milwaukee, Wisconsin J. S. Inserra, HF Tech Library, Dept 32-55 2A	1	Mr. Robert F. Roser Resors Rsch Association Upland, Calif.	1	Dr. Arthur Rubin	1
Libr, Chrysler Def Engr Detroit, Michigan	1	Dr. S. Seidenstein, Org 55-60 Bldg 151, Lockheed, PO Box 504 Sunnyvale, Calif.	1		
Grumman Aircraft Engr Corp Bethpage, L.I., N.Y. L. Bricker, Life Sci, Plant 5	1	Mr. Wesley E. Woodson MAN Factors, Inc. San Diego, Calif.	1		
Hughes Aircraft Co. Culver City, Calif. Co. Tech Doc Ctr, E/110	1	Mr. C. E. Righter Airesearch Mfg Co Los Angeles, Calif.	1		
Itek Corp, Lexington, Mass.	1	Dr. Charles Abrams Human Factors Rsch Goleta, Calif.	1		
Mgr, Behavioral Sciences Litton Sci Spt Lab Fort Ord, Calif.	1	Dr. Corwin A. Bennett Kansas State University Manhattan, Kansas	1		
Dr. Lauritz S. Larsen Univ of SC, College of Engr Traffic & Transportation Ctr Columbia, S. C.	1	The University of Wyoming Laramie, Wyoming Documents Library	1		
Dr. Irwin Pollack Univ of Mich, Ann Arbor, Mich.	1	Dr. Lawrence C. Perlmutter Bowdoin College, Brunswick, Maine	1		
Doc Libr, Wilson Library Univ of Minnesota Minneapolis, Minn.	1	Dr. Alexis M. Anikeeff Univ of Akron, Akron, Ohio	1		
Rsch Analysis Corp, McLean, Va. Document Library	1	The Boeing Co, Vertol Division Philadelphia, Pa. Mr. Walter Jablonski	1		
Ritchie, Inc, Dayton, Ohio	1	Mr. Gerald J. Fox Grumman Aerospace Corp Bethpage, N.Y.	1		
Dir, HF Engr, Mil Veh Org GMC Tech Ctr, Warren, Mich.	1	BioTechnology, Inc. Falls Church, Va. Librarian	1		
Sprint Human Factors MP 537 Martin Co, Orlando, Florida	1				

Security Classification		DOCUMENT CONTROL DATA - R & D	
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)			
1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Institute for the Study of Cognitive Systems Texas Christian University Fort Worth, Texas 76129		Unclassified	
3. REPORT TITLE		2b. GROUP	
VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name)			
Michael Abbamonte Selby H. Evans			
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS	
November 1971	132	43	
8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)		
DAAD05-68-C-0176	Technical Memorandum 21-71		
b. PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
1T061102B81A			
c. Program Element 61102A			
d. THEMIS Proposal No. 367			
10. DISTRIBUTION STATEMENT			
Approved for public release and sale; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Human Engineering Laboratories USA Aberdeen Research & Development Center Aberdeen Proving Ground, Maryland 21005	
13. ABSTRACT			
This document describes a collection of computer programs developed for use in research on human pattern perception. The overall orientation which guided the development of the VARGUS (Variable Generator of Unfamiliar Stimuli) pattern-generation programs and the historical backgrounds of each category (VARGUS 7, 9 and 10) are related in the first section. The second section provides documentation, sample output and summary for each program and subroutine.			

14.		LINK A		LINK B		LINK C	
KEY WORDS		ROLE	WT	ROLE	WT	ROLE	WT
Pattern Recognition		8	3				
Photointerpretation		8	3				
Image Interpretation		8	3				
Man-Machine Decision Making		8	3				
Feature Space		8	3				
Feature Selection		8	3				
Personnel Selection		8	3				