AD-A248 181

DTIC
ELECTE
APR 1 1992
S C D

ANTIFYING THE IMPACT OF THE ACQUISITION PROFESSIONAL DEVELOPMENT PROGRAM CERTIFICATION REQUIREMENTS ON THE AIR FORCE INSTITUTE OF TECHNOLOGY

THESIS

Paul A. Nicholson
Captain, USAF

AFIT/GOR/ENS/92M-22

DISTRI... [illegible] ...ENT A
Approv... [illegible] public release;
Distribution Unlimited

92-08141

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

92 3 31 096

MISSING PAGES WILL BE INSERTED AT AN LATER DATE
AS ERRATA(S)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
March 1992

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Quantifying The Impact of The Acquisition Professional Development Program Certification Requirements on The Air Force Institute of Technology

5. FUNDING NUMBERS

6. AUTHOR(S)
Paul A. Nicholson, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/92M-22

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The purpose of this research is to determine the impact of implementing the Acquisition Professional Development Program on the Air Force Institute of Technology. APDP is part of a DOD-wide effort to decrease inefficiency in the procurement process. The program encompasses a tri-level certification process based on experience, training, and educational standards. APDP serves as a guideline for professional development of acquisition personnel from entry to senior level management. Many of the courses required for certification are offered through AFIT School of Systems and Logistics. Integrating the new educational requirements as a standard for career development is expected to produce a tremendous increase in AFIT course demands. It is the purpose of this research to analyze the acquisition workforce using the standards of APDP and determine the probable course demand facing AFIT School of Systems and Logistics.

14. SUBJECT TERMS
Manpower, Acquisition Process, Acquisition Workforce, Training Analysis, Manpower Simulation, Acquisition Professional Development Program

15. NUMBER OF PAGES
354

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT/GOR/ENS/92M-22

Accession For

Avail and/or
Special

A-1

QUALITY INSPECTED 3

QUANTIFYING THE IMPACT OF THE ACQUISITION PROFESSIONAL DEVELOPMENT PROGRAM CERTIFICATION REQUIREMENTS ON THE AIR FORCE INSTITUTE OF TECHNOLOGY

THESIS

Paul A. Nicholson
Captain, USAF

AFIT/GOR/ENS/92M-22

Approved for public release; distribution unlimited

AFIT/GOR/ENS/92M-22

QUANTIFYING THE IMPACT OF THE ACQUISITION PROFESSIONAL DEVELOPMENT PROGRAM CERTIFICATION REQUIRMENTS ON THE AIR FORCE INSTITUTE OF TECHNOLOGY

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Masters of Science in Operations Research

Paul A. Nicholson, B.S., M.S.

Captain, USAF

March 1992

Approved for public release; distribution unlimited

THESIS APPROVAL

STUDENT: Capt Paul A. Nicholson CLASS: GOR-92M

THESIS TITLE: Quantifying the Impact of the Acquisition Professional Development Program Certification Requirements on the Air Force Institute of Technology.

DEFENSE DATE: 6 March 1992

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Maj Kenneth W. Bauer/ENS	Kenneth W. Bauer Jr.
Reader	Col Thomas F. Schuppe/ENS	Thomas F. Schuppe
Reader	Lt Col William Schneider/LSY	William L. Schneider

Preface

The purpose of this thesis was to develop a decision support tool to analyze the impact of the new training and educational requirements established by the Acquisition Professional Development Program on the Air Force Institute of Technology. The immediate needs of the simulation model contained in this research is to forecast the course demand of APDP courses taught by AFIT School of Systems and Logistics. This information will provide AFIT the necessary flexibility to adjust appropriately during the DOD-wide movement to increase the proficiency of the acquisition workforce.

Because of the genial approach used in emulating the APDP process, the model can easily be adapted to support acquisition manpower analysis efforts throughout the Air Force.

In performing the experimentation and writing this thesis I have had a great deal of help from others. I am deeply indebted to my thesis advisor, Major Kenneth Bauer and co-advisor, Lieutenant Colonel William Schneider. I could not have completed this research without invaluable lectures on SLAM offered by Major Bauer. Lt Col Schneider provided indispensable information to increase my understanding of the underlying policy and operations of APDP.

I would also like to thank Major Aaron Glover and the rest of the Head Quarters Air Force Material Command provisional staff for consistently providing timely data and other critical APDP information.

Finally, I would like to thank my wonderful wife, Wendy, for her patience, prayers, and understanding throughout my graduate school experience. Without her tireless support, this endeavor would have been much more difficult and empty.

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this research is to determine the impact of implementing the Acquisition Professional Development Program on the Air Force Institute of Technology. APDP is part of a DOD-wide effort to decrease inefficiency in the procurement process. The program encompasses a tri-level certification process based on experience, training, and educational standards. APDP serves as a guideline for professional development of acquisition personnel from entry to senior level management. Many of the courses required for certification are offered through AFIT School of Systems and Logistics. Integrating the new educational requirements as a standard for career development is expected to produce a tremendous increase in AFIT course demands. It is the purpose of this research to analyze the acquisition workforce using the standards of APDP and determine the probable course demand facing AFIT School of Systems and Logistics.

QUANTIFYING THE IMPACT OF THE ACQUISITION PROFESSIONAL DEVELOPMENT PROGRAM CERTIFICATION REQUIREMENTS ON THE AIR FORCE INSTITUTE OF TECHNOLOGY

I. Introduction

This research examines some of the impacts of congressional initiatives to enhance the efficiency of the procurement process. In particular, this thesis will examine those initiatives on improving the acquisition workforce and determine the impact of the newly imposed training and educational requirements for acquisition personnel on the Air Force Institute of Technology (AFIT).

Background

Throughout the evolution of military system's technology, it has been common place to the field of defense systems acquisition for some defense contractors to provide vital technology to government organizations at an unreasonably high cost. Discoveries of spare-parts, overpricing, poor quality, overruns, and excessive specifications of programs have fueled an already intense scrutiny and corresponding distrust of the defense acquisition community. Strict budget cuts in the Department of Defense (DoD), and the exponential growth of these scandals in both occurrence and cost, directed the attention of policy makers to the defense procurement process.

In 1989, the Secretary of Defense, Richard Cheney, was directed by the President to develop a plan to improve the procurement process. In Cheney's study of the acquisition workforce, he made the following observation:

> The defense acquisition workforce mingles civilian and military expertise in many disciplines for management and staffing of the world's largest procurement organization. Each year billions of dollars are spent more or less efficiently, based on the competence and experience of these personnel. Yet, compared to its industry counterparts, this workforce is undertrained, underpaid, and inexperienced. Whatever other changes may be made, it is vitally important to enhance the quality of the defense acquisition workforce -- both by attracting qualified new personnel and by improving the training and motivation of current personnel. (4:13)

To meet the requirement for a more proficient defense acquisition workforce, Secretary Cheney established the Acquisition Professional Development Program (APDP). This idea is an evolutionary process extending from the Acquisition Management Professional Development Program (AMPDP) which focused on the educational requirements of program managers. The purpose of the APDP is to maximize job proficiency of personnel in the acquisition career field by implementing a distinct education and training certification plan throughout total career development. The program establishes minimum education, training, and experience requirements that are important for an individual's successful pursuit of senior responsibilities (5:7). The certification process divides into three levels corresponding to junior, middle, and senior level management respectively. In accordance with the Defense Acquisition Workforce Improvement Act of 1990, the Acquisition Professional Development Program began implementation on 1 October 1991. Full scale implementation is not expected until after 1 October 1993. The program requires all personnel seeking new positions to be appropriately certified through the APDP process. Personnel occupying positions prior to program implementation will be allowed to maintain their current level of certification until a position change is required. (20:Sec. 1121-1301).

General Issue

As the technological needs of the Air Force continue to grow, so does the complexity of the contracts negotiated to obtain this technology. To help prevent government organizations from falling prey to profiteers and prevent the purchase of systems not capable of meeting the needs of the Air Force, the competence of the acquisition workforce must be increased. The career development approach of APDP strives to produce a workforce of proficient personnel that is equivalent to its industrial counterparts. Although civilian institutes will be instrumental in meeting some educational and training requirements necessary of the acquisition workforce, most of the responsibility rests upon the Air Force and other DoD educational institutions.

Problem Statement

The career development requirements of the acquisition workforce set by APDP demand a range of educational and training components such as professional military education, acquisition specialty courses, and advanced academic degrees (6:16). These certification requirements for existing personnel as well as new accessions to the acquisition career field will heavily tax DOD educational institutions in terms of resources required to increase student throughput and subsequent certification.

Research Objective

Although the multi-level certification process will impact many DoD educational and training organizations, it is the purpose of this thesis to quantify the impact of the APDP certification requirement for the Acquisition Logistics career specialty on the Air Force Institute of Technology (AFIT). In particular, this thesis will identify the amount of increase in training requirements necessary to certify the Acquisition Logistics personnel during the initial backlog and future steady state condition. Additionally, yearly training projections and a workforce profile will be accomplished using a twenty-eight year aging process. The aging process will employ a combination of life-cycle events such as promotions, attrition, and accessions and APDP events such as AFIT training courses, certification screening, and position updates using the certification criteria. All life-cycle and APDP events will occur within the span of a calendar year and replicated over 28 years.

Sub-objectives

To solve this research problem, several key objectives were to be identified. Each of these objectives are explained in greater detail in the methodology sections of chapter 3.

Data Collection. Identify and sample the necessary data elements to model the process of APDP certification.

Acquisition Workforce. Identify the acquisition workforce in terms of position requirements and the population of people currently filling those positions.

Population Status. Identify the current level of certification for each person based on the certification level field in the individual's records and the associated educational history.

Backlog Training Requirements. Identify those persons in the acquisition workforce who are serving in positions which require a higher level of certification than that obtained by the occupant.

Current Capability. Identify all AFIT resources currently used to certify acquisition personnel. Identify the procedures used by AFIT to certify acquisition personnel at each level of management and the associated number of students AFIT currently certifies through each process.

AFIT APDP Training Contribution. Identify the frequency and media of those courses taught by AFIT in support of APDP certification.

New Training Requirement. Identify the relevant range of the percent of the acquisition workforce expected to utilize AFIT resources for training. Assuming a uniform distribution of attendance over the given population percentage, identify the new training level dictated to AFIT based on student demand.

Steady State Conditions. The total impact of APDP certification requirements on AFIT must be computed under two conditions, backlog and steady state conditions. The backlog training conditions emerge from the population of people serving in positions above their certification level as of 1 October 1991. The long range impact or steady state condition is identified by finding the time line in which backlog requirements no longer dictate training levels and requirements are based solely upon personnel moving into positions based on accessions, promotions, and sufficient training.

Research Approach

The approach taken in this research effort was to undertake an extensive literature review that included the structure of the acquisition workforce, laws and regulations forming the Acquisition Professional Development Program and a dissection of the efforts undertaken by Air Staff and other command head quarters to meet the requirements of APDP. The information gained in this review was scoped down to represent AFIT's role in improving the acquisition workforce. Once the problem had been totally defined and appropriately scoped, an analysis of AFIT's APDP role was accomplished using a discrete simulation model in Simulation Language for Alternative Modelling (SLAM). The model was constructed to emulate the demands placed on AFIT by the acquisition workforce due to the policies of APDP. The model encompassed twenty-eight years of replication to provide a confident projection of the backlog and steady state training requirements.

II. Literature Review

Introduction

This chapter provides a review of the literature on the educational and training problems arising from the acquisition professional development program certification requirements. The emphasis of this review is on an overview of the acquisition career field, the origin of the acquisition workforce certification requirement, and the different simulation approaches that can be used to analyze the problems arising from the certification requirement.

Overview of the Air Force Acquisition Career Field

Most narrowly defined, the acquisition workforce is comprised of only those who negotiate and administer contracts for major weapon systems. Broader definitions include activities occurring outside the contract process, such as documenting the need for a new weapon, testing systems under development, maintaining systems in the field, and disposing of outmoded or unnecessary equipment. A still more comprehensive perspective would encompass all those who procure the ordinary goods, such as office supplies and delivery vehicles, needed to support any large organization within DoD (12:104).

Air Force Regulation (AFR) 36-1 governs the officer specialty classification system. The Air Force classifies the types of primary officer duties according to "utilization fields" or "specialties." Each officer job type is assigned a four-digit Duty Air Force Specialty Code (DAFSC). The first two digits represent the utilization field to which the officer is assigned. The third digit determines the officer's specialty while the fourth digit signifies the rank limitation (12).

Acquisition is defined by several DoD directives in the 5000 series. Pertinent to the research contained in this study, information will be extracted from *Systems Acquisition Management Careers* (5000.23), *Defense Acquisition Education, Training, and Career Development Program* (5000.52), and *Reporting Functional and Training-Related Data on DoD Military and Civilian Acquisition Personnel* (5000.55). Many utilization fields are associated with acquisition management. The fields include Program Director (0029), Program Management (27XX), Cost/Budget (67XX), Test Pilots/Navigators (286X/287X), Development Engineering (28XX), Communications-Computers Systems (49XX), Contracting/Manufacturing (65XX), and Logistics Plans and Programs (66XX). All of these personnel combine to structure the acquisition workforce. However, when referencing "acquisition manager," the term generally applies to officers in the 27XX career fields. Because of their critical role in the procurement process, acquisition managers have been targeted as the prime source of efficiency improvement in the procurement process (11:1-81).

Origin of Certification Requirement

Outrageous prices for simple items of hardware are only the most obvious symptoms of deeper problems in management of military procurement. The Strategic Air command paid $982 for a nylon stool cap worth eighteen cents and only began to investigate the cost when it rose to $1,118 (3:12). General Dynamics billed the Air Force $9,606 for a twelve cent allen wrench. The Naval Department relieved an officer of duty because he contracted custom design ashtrays for $659 (20:14). Beyond contractor fraud permitted by contract negligence in the procurement process, budgetary waste also stems from contract over-specification and lack of quality control.

Aircraft refrigerators priced at $17,000 and coffee brewers for $7,600 appear not only because the items are produced in very small, inefficient lots, but also because the military requires that even such common equipment be built according to customized designs and with very high military specifications. Airline beverage-makers are available for about $3,000. Rather than settle for commercial equipment, the Air Force paid $7,600 for a coffee-brewer on the C-5A air transport. The brewer was the first of its kind, designed to work even if the plane lost all cabin pressure, and able to withstand forty gravities of acceleration, although the loss of cabin pressure would force the plane down and forty times the force of gravity would kill all the passengers. In 1984, after spending $1.8 billion to develop the Division Air Defense Gun (DIVAD), the Pentagon belatedly canceled it because of test failures and poor reliability. (1:24). These examples of contract over specifications and lack of quality control are prime examples of the willingness of DoD officials to accept high technological risk in systems procurement at the expense of reliability, maintainability and affordability.

After completing an investigation of fraud against major defense contractors, "General Joseph Sherick revealed to a Congressional committee that forty-five of the nation's top 100 defense contractors were under some kind of criminal investigation for fraud. Later he supplied a list that revealed that eight of the top ten defense contractors were among those under investigation." After several years of adjudication, four fraud cases have been proven, two dismissed and two still pending(12:105).

The estimates are controversial, but they suggest that over the past four decades the nation may have lost tens, even hundreds, of billions of defense dollars to waste and inefficiency. Those losses are significant, relative to a total military budget now approaching $300 billion annually. Contractor fraud, contract over-specification, and lack of contract quality control are all prime contributors to the vast losses suffered by the DoD budget (22:11). A common factor to each of these elements is the acquisition workforce.

In 1986, President Ronald Reagan assigned David Packard, former deputy secretary of defense 1969 - 1971 and president of Hewlett-Packard Company, to chair the Blue Ribbon Commission on Defense Management that became known as the Packard Commission. The purpose of the commission was to develop a plan to improve the defense procurement process and management of the Pentagon. In the report to the President, the Packard Commission defined two major issues on improving the procurement process. First, the defense acquisition system was identified as "a major contributor to the long delays in getting new technology in the field that builds formidable barriers to exploiting technology developed in the civilian sector" (4:26). The report further explained that while congress did not intend the system to be slow, cumbersome, and inefficient, laws passed to foster goals other than efficient procurement made it so (4:26). Secondly, the report pointed out that the acquisition workforce spends billions of dollars each year more or less efficiently, based on their competence and experience (4:12). The report concluded that despite what other congressional actions may be necessary to improve the procurement process, streamlining the acquisition process to delete cumbersome bureaucracy and improving the competence of the acquisition workforce would be a prerequisite for improvement.

In accordance with the recommendations made by the Packard Commission, the 1987 Defense Authorization Act provided the Secretary of Defense the authority to realign agencies responsible for all stages of the procurement process (21:Sec. 1202). Continuing in his effort to enhance the procurement process, the Defense Authorization Act of 1991 called for the creation of the Defense Acquisition Workforce Improvement Act that established all the critical acquisition positions and programs necessary for producing a proficient workforce. To ensure that DoD-wide training, education, and career development policies concerning civilian and military acquisition personnel are developed and implemented effectively, the Acquisition Professional Development Program (APDP) was developed.

APDP Implementation **.** This program seeks to maximize the professional development and mission capability of the acquisition workforce by setting forth a definitive and viable professional development management plan that produces broad-based managers capable of assuming middle and senior management roles in support of acquisition (5:7). Intrisinc to the philosophy of the APDP is the fact that the Air Force acquisition system needs trained and experienced professional leaders at all levels to ensure mission success. The APDP must produce acquisition managers with broad experience, yet with a common core of experience, training, education, and professional development. This mixture of qualifications must be achievable and is accomplished in part by providing opportunities for high-potential officers in related fields to transition into acquisition and encouraging operational broadening assignments for those officers who begin their careers in acquisition (5:5-14). This program became functional in October 1991, and set forth the requirement for all acquisition personnel to be certified according to their functional duties by October 1993 (21:Sec. 1735).

The certification function consists of an individual record screening process against education, training, and experience standards. The screening process provides a three level certification identifier which channels promising officers into jobs of high responsibility such as program director duty. Each level has specific requirements to ensure proper levels of academic and military education, specialty training, and acquisition-related experience. The three certification levels corresponding to basic, intermediate, and senior level management. The basic level is designed to establish fundamental qualifications and expertise in the individual's job series or career field. When individuals enter into a level one positions, they are expected to gain level one certification within eighteen months of their entry into the job. For all other job levels, personnel are expected to obtain the necessary certification prior to entry into the position. Level one lays the foundation for career progression and is designed to prepare qualified, motivated personnel for positions of increasing responsibility (12:2-1). At this level, personnel are expected to gain widespread exposure to the different acquisition career fields. Hence, vast occurrences of crossover among people in different career stalls is expected of the population of level one personnel. The expected grade of personnel at this level range from second lieutenant to major. At level two, intermediate management, specialization is emphasized. Occurrences of crossover is minimized and tour length in individual career stalls is increased. The expected grade of personnel at this level range from captain to lieutenant colonel. At level three, senior management, career progression is highly specialized and advanced acquisition education and training become essential. By the time an individual reaches the senior levels of acquisition management, he or she should have completed all the mandatory training and education requirements to that level. Career advancement should have provided a pattern which maximized the individuals exposure to his functional area while still allowing for a breadth of knowledge across the entire acquisition process. The expected grade at this level range from lieutenant colonel to high ranking generals (12:2.1-2.9).

The formal selection process of APDP permits the screening of certified acquisition managers and the identification of those with the greatest potential to assume senior acquisition duties. Four distinct screening points exists, one for each level of certification and one for entry into the acquisition corps. The acquisition corps consist of those level two and three officers capable of filling critical acquisition positions. Since each level of certification has both educational and experience requirements, job selection as well as educational accomplishments is a decisive element of career advancement. The building block structure of education, training, and experience requirements is the key element for the APDP goal to improve the proficiency of the acquisition workforce.

The APDP requirements on the DoD level will be implemented in accordance to the guidelines presented in DoD manual *Career Development Program for Acquisition Personnel* (5000.52M), and DoD Instruction *Reporting Management Information on DoD Military and Civilian Acquisition Personnel and Positions* (5000.55). Current Air Force guidelines for implementation of the requirements of APDP are expressed in AFR 36-27. DoD guidelines serve as a baseline requirement for APDP implementation.

Combined, 5000.52M and 5000.55 instruction provide all the necessary personnel and manpower information necessary for implementing the APDP. Instruction 5000.55 provides the necessary information within the DoD to establish a management information system capable of providing standardized information on acquisition positions and persons serving in acquisition positions. This document details each position and personnel record required to maintain information on the acquisition workforce. It contains the necessary instructions for submitting requests for APDP certification at each of the different levels, certification waiver request, and other forms necessary for the standardized flow of information on the acquisition workforce across the DoD community (11:2-13).

Manual 5000.52M establishes the guidelines for career development of all personnel serving in acquisition positions in the DoD. It also contains the educational, training, and experience standards for specific acquisition workforce position categories and career fields. Additionally, the manual provides guidelines for APDP certification along each of the distinct acquisition career paths (12:1.2-1.3).

Problem. The career development requirements of the acquisition workforce set by APDP demand a range of educational and training components such as professional military education, acquisition specialty courses, and advanced academic degrees (6:16). These certification requirements for existing personnel as well as new accessions to the acquisition career field will heavily tax DOD educational institutions in terms of resources required to increase student throughput and subsequent certification. The problem is to determine the impact of backlog and increased steady state educational and training requirements arising from implementation of the APDP on AFIT.

Modelling the APDP Process

Problems similar to the APDP problem facing AFIT have been analyzed using decision support models such as, continuous simulation models with emphasis on finite capacity flow, discrete simulation models, and pattern simulation models using relational data bases. The relevance of each of these approaches to the problem defined in this research is discussed below.

Continuous Simulation Model. Simulation of this problem is analogous to a pipeline-flow problem (20:238). Envision an oil tanker dispensing fuel through a large pipe. The large pipe leads to several smaller pipe. The smaller pipes lead to a group of even smaller pipes. Each pipe in the system has a finite capacity flow. The existing configuration of pipes and valves is sufficient for unloading a single ship, but a new dock has been completed to enable two ships to unload simultaneously. The problem now facing the station manager is to decide which portions of the pipeline structure constrain the station from meeting the demand of unloading an extra ship. Like the volume the new ship adds to the unloading station, the new certification requirements enforced by APDP creates a greater demand for courses taught by AFIT for certification at each level of management. Similar to the different areas of the pipeline, the maximum capacity of students AFIT can certify at each level of management is fixed by the institution current level of resources and methods of instruction. Using these fixed levels of input and output, the simulation model can find the resulting bottlenecks in the system and provide a time schedule on the amount of backlog that will result. Fixing the appropriate distributions for new accessions and promotions in the acquisition career field and dynamically adjusting the number of people AFIT can certify for each level of management will enable the simulation model to provide critical information on how much AFIT must adjust certification training at each level of management to account for the new demand. Using the required adjustments and the fixed cost of resources used during certification, the total impact on AFIT can be computed in dollar amounts. There are several major advantages to the above simulation approach to this problem. The first is the additional information provided on the distinct bottlenecks of certification at each level of management. Secondly, the simulation output provides the capability of dynamic sensitivity analysis without repeating several experimental runs. Thirdly, time lines and histograms monitoring the flow of personnel through the AFIT certification process are available. Therefore without any further experimental runs, backlog and steady state training requirements can be analyzed. The final and most important advantage to this approach is the simplicity of the simulation model. Since the different levels of certification can be simulated and analyzed independently, changes to one section of the model during sensitivity analysis do not require changes to the entire flow environment of the model (2:904). The major disadvantage to this approach is that the APDP process is a collection of discrete events taking place throughout the year. The completion of each course at AFIT may cause an instantaneous jump in certification for a class of people. Hence, statistics on backlog depletion, APDP course demand and steady state training requirements could be inaccurate if the process were modelled continuously.

Discrete Simulation Approach. Because of the discrete events occurring in the APDP certification process, discrete event simulation appears to be a viable option. After close analysis of the APDP process, the problem can be formulated as events to analyze the initial population, conduct training during each year and APDP certification screening at the end of each year. Using discrete simulation, each APDP process can be identified as a school event or end-of-year screening event. This approach would allow accurate reporting of the discrete statistics unavailable in the continuous simulation approach. Since some statistics are continuous, the use of time persistent functions can be used to accurately collect this information. Although many simulation languages provide the capability to accomplish discrete event modelling, only SLAM will be considered for this research. This is due solely to the placement of the SLAM course early in the Masters of Operation Research curriculum which negated any unnecessary learning curves during model building.

Pattern Simulation Approach. Compared to the finite capacity flow approach, pattern simulation using relational data bases is an advanced and more complex approach to the problem. Simply stated, this approach simulates the pattern of mandatory education and training required for a specific job from entry to the highest level of proficiency. This approach uses the hierarchical relationship capability of data base software to describe required training patterns for personnel in a given career field. The data base links the appropriate probabilities and statistics necessary to describe the movement of a person from the unqualified state of career entry to job proficiency at the highest level of management. This data base serves as an interactive input module to a simulation model to emulate personnel moving through career development (22:5). In their recent efforts to evaluate the impact of policy changes in manpower, personnel, and training (MPT), the Air Force Manpower and Personnel Center (AFMPC) employed the concept of pattern simulation in their new training system. The new system, Training Decision System (TDS), is a computer-based decision support system used by AFMPC decision makers to model the flow of airmen through jobs in an Air Force occupation (8:4-11). TDS provides procedures for building data bases concerning the dynamic flow of people through jobs for both formal and on-the-job training (OJT). Furthermore, the TDS includes modeling and optimization capabilities that provide estimates of training quantities, costs, and capacities for both formal and OJT. Such capabilities allow the TDS to go beyond simply describing the current situation in a specialty (22:1). The TDS can model alternative scenarios which reflect possible policy options. This characteristic of TDS is called "what-If" scenario modeling (22:15). This modeling technique can be used to decide the necessary trade-off between formal, OJT, or correspondence training.

Using pattern simulation to solve the acquisition certification problem, it is necessary to build a data base that describes the educational and training path for personnel in the acquisition career field. This training pattern will serve as the control baseline for career progression in the acquisition specialty. Incorporating the appropriate changes in career progression due to the implementation of APDP will provide the experimental what-if scenario. Running the TDS model for this alternate career pattern will analyze personnel career progression and provide estimates on training quantities, costs, and capacities associated with the career pattern change (22:1-6). Using the same cost model from the previous approach, the total impact on AFIT can be computed in terms of dollar amounts. Figure 2.1 illustrates the TDS operation-phase process. As may be seen, the TDS modeling process involves two major steps. First, inputs are made for the baseline and alternate utilization and training (U&T) patterns to be modeled. This involves data about the jobs and training courses--in particular, the transition probabilities for moving from job to job and from job to training course (22:3-4).

The second step involves estimating training resource requirements, costs, and capacities associated with the U&T pattern simulation completed in stage one. First, Training resource quantities are estimated to support the training quantities from the U&T pattern model; these include both labor (student and instructor) and nonlabor (e.g., training equipment) hour requirements. The training resource requirement estimates are the basis for both cost estimation and capacity analysis. Costs are estimated by applying cost

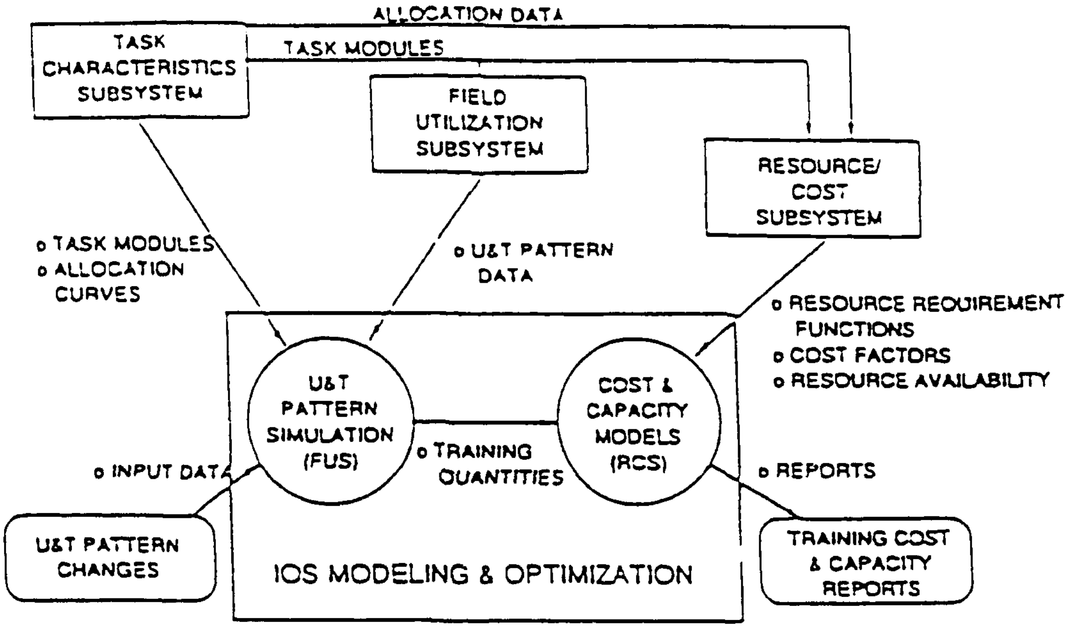


Figure 2.1 Training Decisions System Modeling Process

factors (e.g., salaries) to resource requirement estimates. Training capacities are analyzed by comparing resource quantities required to those available at representative training sites. The training capacity analysis provides an overall capacity estimate of the numbers of trainees who can be trained at a site. The capacity analysis also provides resource-by-resource statistics (22:4).

This approach clearly has many advantages over the previous approach. The major advantage is the capability to impose policy and regulation changes throughout career progression using a single experimental run. The capability to evaluate total career changes makes this approach more adaptable to the APDP certification problem. The second advantage of this approach is the flexibility provided by what-if modeling. This comprehensive model can be modified to numerous policy changes over many career fields. The grave disadvantage to this approach is the complexity of programming required to accomplish an analysis of the problem. The learning curves associated with reprogramming the TDS model, building the required relational database of acquisition personnel, and incorporating the new APDP policy are prohibitive.

Summary

The literature review detailed a history of USAF procurement lined with horror stories of unnecessary budgetary waste totalling billions of dollars. The scrutiny of the weapon system acquisition process arising from these horror stories generated initiatives targeted at providing a seasoned, well-trained acquisition cadre. Such efforts are embodied in the work of the Packard Commission which concluded that to prevent inefficient government spending, the proficiency of all personnel involved in the procurement process would have to be certified according to their managerial duties. DoD adopted the recommendations of the Packard Commission and The Acquisition Professional Development Program was established to meet the necessity to improve the proficiency of the acquisition workforce. The APDP called for a tri-level certification process based on increased experience, training and education requirements for all people in the acquisition workforce. These certification requirements for existing personnel as well as new accessions to the acquisition career field will greatly increase the demand for educational requirements from DOD educational institutions in terms of resources required to increase student throughput and subsequent certification. In particular, the demand placed on courses taught by AFIT will be substantially increased. The impact, if any, caused by this new demand for courses to support APDP needs to be determined. Simulation models with emphasis on flow capacities and employing data base training patterns were analyzed as decision support systems to aide decision makers in finding the impact of the certification requirement on AFIT. Analyzing the problem through the use of continuous simulation provides a simplistic approach but does not provide a comprehensive view of total career progression. Although Pattern Simulations using relational data bases provide all the necessary tools for analyzing the problem, the complexity of the model may reduce its utility. Further analysis of the discrete process undertaken in APDP certification process identified discrete simulation modelling using SLAM as the most viable option for analyzing the problem.

III. Methodology

Introduction

This chapter presents the methodology used in completing the objectives outlined in Chapter I. The first section discusses the exploratory study done to understand the Acquisition Professional Development Program (APDP) and the role AFIT plays in the certification process. The second section explains data collection and the development of each research objective necessary to construct a simulation model. Since information contained in this section will be augmented with Chapter IV, Model Formulation, model discussion in this chapter will be limited to the construction of the research objectives as a simulation model. Model assumptions and functionality will be discussed in the next chapter.

Exploratory Study

The problem of determining the impact of APDP requirements on AFIT originated in AFIT School of Systems and Logistics. The School of Systems and Logistics is responsible for many of the core and specialty courses necessary for APDP certification. Prior to the implementation of the current APDP, specialized education and training requirements were limited to only a few of the acquisition career fields operating in the procurement process. The APDP, designed to increase proficiency of the entire acquisition workforce, set minimal educational standards across the entire spectrum of acquisition career fields. Realizing that resources required for student throughput was already nearing its maximum, Lieutenant Colonel William Schneider spearheaded the effort to determine the impact of these new requirements on AFIT's operations. He solicited support in analyzing this problem from AFIT School of Engineering, Operations Research Department. Under his guidance, a series of trips were made to scope this DoD-wide movement down to the role of AFIT.

Because of their experience with the original certification program, Acquisition Management Professional Development Program (AMPDP) and their recent efforts to predict APDP impact on systems command, a trip was made to Headquarters Systems Command, Andrews Air Force Base. Captain Daniel Gerrig was the action officer for assistance in the research efforts. Captain Gerrig presented a briefing on the processes of APDP and provided vital liaison to manpower analysis offices at the Air Staff level. Captain Gerrig had recently completed a spreadsheet model used to identify APDP training requirements, which provide rough estimates of the course demand for Introduction to Acquisition Management (SYS100), Acquisition Planning and Analysis (SYS200), and Advanced Program Management (SYS400). The population of this model was limited to systems command officers in Duty Air Force Specialty Codes (DAFSC) of 26XX, 27XX, and 28XX. The model employed rates for training, promotion, retention, and accession to age the population and predict training demand. The model predicted that the backlog training for SYS100 in systems command would diminish by 1991 and steady state training capability would be equal to demand by 1992. The model also predicted the demand for SYS200 and SYS400 in systems command could not be adequately supported by the current training capability. Hence identifying these courses as the "choke points" of APDP certification in the future years. In the briefings presented on his model, he identified the need to validate the results of this model and build an improved model that would extend his efforts to a total Air Force level. In addition to other requirements, this research is designed to encompass that goal.

Following the guidance of Captain Gerrig, Captains John Garstka and Michael Grant of the Pentagon USAF/DPXA manpower analysis directorate, were contacted. Captain Grant provided briefings on the process of manpower modelling and force analysis. He demonstrated several Simscript models and pointed out some key elements of manpower simulations. These elements will be discussed in detail in the next chapter on model formulation. Captain Garstka is the project officer for an eighteen month contract to SRA corporation to assess the feasibility of developing a policy management model to support Air Force acquisition force career development under the terms of the Defense Acquisition Workforce Improvement Act (DAWIA) and APDP. Additionally, the contract employs SRA to develop a Decision Support System to be used by the Secretary of the Air Force Acquisition, (SAF/AQ) in managing the Air Force acquisition workforce. The contract is scheduled to be complete early 1993. Because of the experience he gained from working APDP through the SRA contract, Captain Garstka was instrumental in identifying some of the key data elements necessary for an APDP simulation model. He explained the process in five stages: procurement, education, training, utilization, and the retirement/separation stages. Figure 3.1 demonstrates the relationship of these stages and several common factors of each stage that must be considered when modelling manpower and the influences of policy. AFIT's role is captured in the educational and training blocks of this diagram.

The procurement stage represent the management of new accessions and existing personnel in the acquisition workforce. There are three major goals here. The first is to identify the existing population of people in the acquisition workforce. Next, the current certification status and educational deficiencies of these people must be determined. Finally, population rates such as promotion, training, accessions, and attrition must be identified. This stage identifies a "snap shot" of the acquisition workforce and

Modeling APDP Processes

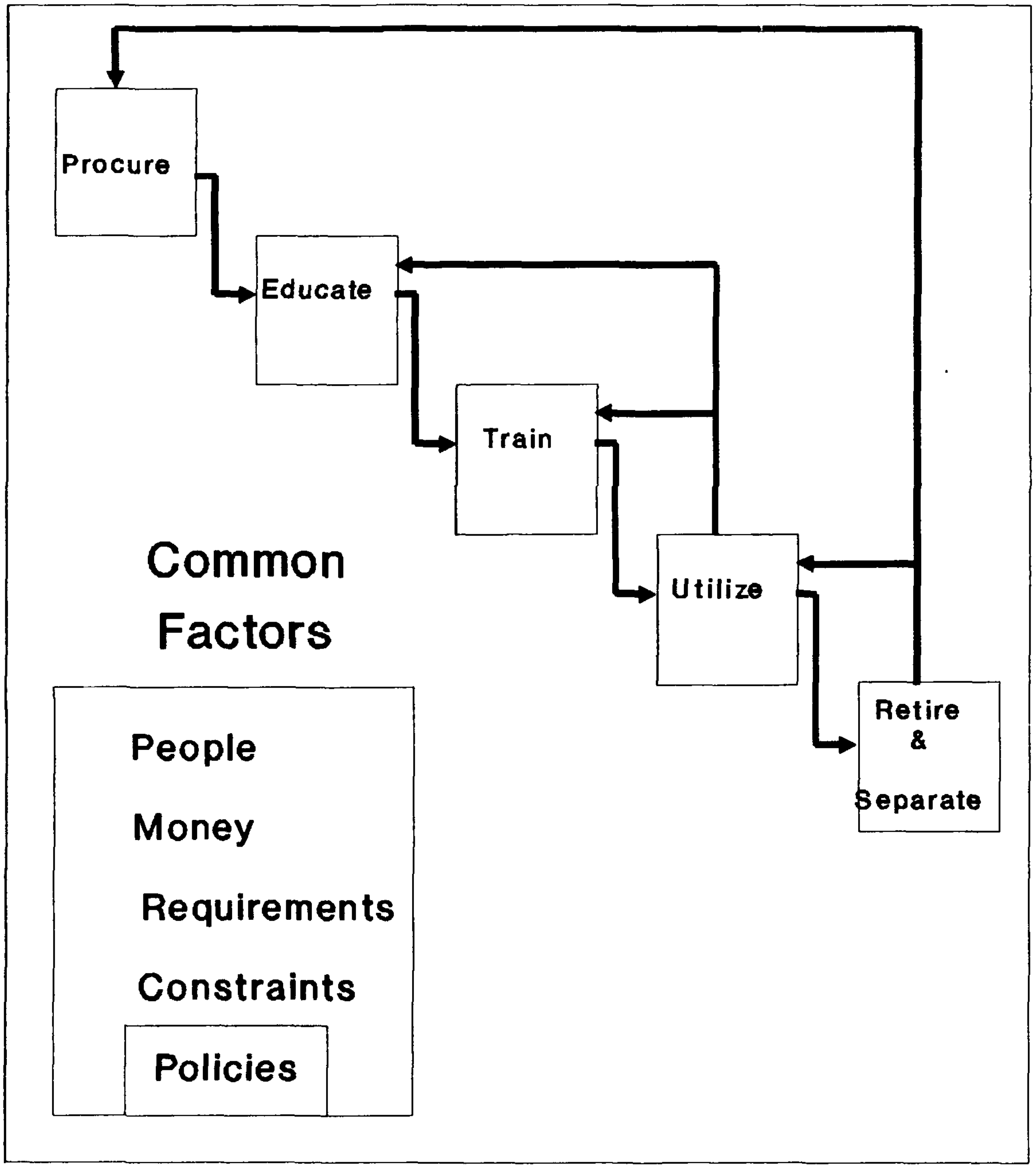


Figure 3.1

prepares it for the aging process during simulation.

The next two stages capture AFIT's primary role in the APDP process. AFIT's APDP function is to provide acquisition personnel the necessary educational and training courses required for job proficiency. At a minimum, this task entails determining the needs of the acquisition community and maintaining course materials which reflect those needs.

The utilization stage represents the cycle of personnel being educated, trained, and utilized at each level of certification throughout career development. The final stage, retire and separate, represent two cycles of events in the acquisition community. The first cycle refers to the employment of retired or separated military personnel as civilian employees. When this occurs, knowledge and experience is retained in the acquisition workforce. The second cycle represents the opening of jobs for forward progression opportunity and thus new accessions when people retire or separate. This cycle is represented by the line in Figure 3.1 from the retire/separation block to procurement block. Although each of the common factors listed in the diagram can be associated with a particular stage in the acquisition personnel life-cycle, the influences of these factors are actually common to each stage. The common factors of people and requirements can be combined to represent the student population requiring AFIT courses for APDP certification. The common factors of money and constraints can be combined here to represent the necessary resources required by AFIT to provide the requirement of APDP courses to the acquisition workforce. Policy represents the influence of DAWIA and APDP on all acquisition personnel. It also captures AFIT's policy on course frequency, course content, and other Air Force regulations governing the training and educational process. The information gained from Captains Michael Grant and John Garstka was critical in diagramming the high level processes of APDP as a simulation model. The next step was collecting the data which enabled modelling each of the stages listed above.

Development of Research Objectives

Data Collection. Since this research is centered around the structure of the acquisition workforce in terms of positions and personnel, a trip to the military and civilian manpower and personnel centers was necessary to accomplish this objective. Although the simulation model constructed in this research is limited to military personnel certified according to the standards of the Acquisition Logistics career field, the development of the research model encompassed all acquisition career fields and both the military and civilian populations. At the Air Force Manpower and Personnel Center manpower analysis shop (AFMPC/DPMYA), Major Glenn Bailey acted as the action officer for all research simulation efforts.

Major Bailey is currently a rated force analyst and performs manpower analysis on problems similar to that of the APDP certification problem. Mrs. Barbara Smith, Chief of System Developmental Analysis (AFMPC/DPMRS), was also heavily involved in the research. Mrs. Smith served as a focal point for identifying the location of all APDP data elements in the military manpower and personnel files. A listing of the required data elements and their associated simulation attributes can be found in Appendix A. Using their combined expertise, several extractions were made from the manpower and personnel databases to obtain the necessary APDP data used for the simulation model. A sample of this data can be found in Appendix B. Since both the military and civilian population compete for the same course quotas at AFIT, the population size and certification demands of the civilian population affect the rate of certification for military personnel. To accurately model the military training process, the reduction in training rate due to the influence of civilian personnel had to be included. The Air Force Civilian Personnel Management Center systems requirement shop (AFCPMC/DPCIR) provided all critical statistics on the civilian population.

Acquisition Workforce. This objective requires a file containing the attributes of all military acquisition workforce positions and the people filling those positions. Additionally, population totals and critical rates must be collected for the civilian acquisition population. This objective requires dissecting the workforce into its various population components. There are two populations, the military and civilian populations. Each population has two components, positions and personnel, which are maintained in separate data base files. Position files will be referred to as the manpower database and personnel files will be referred to as the personnel database. Each position component refers to a mission requirement and has attributes (characteristics) of the job necessary to support that mission requirement. Examples of position attributes are rank, acquisition position type, and acquisition career field type. The personnel component refer to an individual and consist of attributes listing the individual's qualifications and accomplishments. Such examples of personnel attributes are rank, years of service, and educational history. The military and civilian centers have unique methods of tracking the manpower and personnel attributes of the acquisition workforce. As a result of APDP implementation, DoD Instruction 5000.55 provides new directions governing the standardized tracking of all acquisition manpower and personnel information. Since this research must be completed prior to the necessary software changes to implement the standards of 5000.55, the interim methods of obtaining this data will be discussed. The military acquisition workforce identifies its position omponent using a combination of two methods. The first method of identification is to augment each acquisition position with a "H" prefix. Since some positions already have designated prefixes, a second method had to be utilized. The second method used the Required Language (RLA) field of each position to code the acquisition position type and acquisition career stall. After all military positions were identified, similar attributes in the personnel file were use to identify the population of people. Two files containing all records of positions and personnel were compiled. Now that the positions and personnel were identified, the task at hand was to match the person to the appropriate position. Using the position number, which is identical in both files, a fortran sort program was employed to match people to their associated position. Since civilian manpower and personnel files can be extracted simultaneously, statistics on the position also provides statistics on the personnel. The civilian personnel system employ the functional job class code to track members of the acquisition workforce. This data entry is equivalent to the information contained in the RLA field of the military system. All statistics concerning the civilian acquisition workforce were be obtained by a database query using this field. No specific data records were extracted for civilian personnel, only statistics on population totals and population rates such as attrition and accession.

Population Status. The objective here is to determine the certification status of each person in the military acquisition workforce. This status will be used to compute the population of people certified at the different levels of certification. Unless an individual has obtained a level three certification, a list of course deficiencies necessary to get level three certification will be maintained. This information will be used to compute the initial and expected course demands for each year during the aging process. This objective contains two parts. Part one involves determining the current certification level of each person and part two generates an evaluated certification level for each person in the military acquisition workforce. Since each personnel record has a data field which reports the current certification level, part one consists only of reading each file and the associated certification level field. Under the APDP program, certification consists of training, experience, and education. Relative to AFIT's role to provide education to the acquisition workforce, certification consists only of completing the required courses. Hence, each person's educational history is searched for acquisition courses completed. A temporary file of credit for each course completed towards certification at the different level is maintained. After the search is complete and all credits given, an evaluated certification level is assigned based solely upon courses taken. For example, if an individual is certified at level one and has taken the necessary courses to be certified at level three, but lacks the necessary experience for the certification, his records will report only a level one. The evaluated level of certification will be level three. This assumption is valid since this person no longer require APDP courses at AFIT, and should not be consider a part of the demand population. One of the critical elements of determining the evaluated certification is compiling a master list of acquisition courses that could possibly appear in an individual's educational history. Once this list has been constructed, the appropriate credit must be applied under the new APDP program. In essence, the equivalence of the historic courses must be determined in order to provide accurate credit under the new certification system. This milestone requires a detailed review of historic acquisition courses occurring in Air Force Regulation (AFR) 50-5. This regulation contains a listing of the course codes for all credited courses that could be maintained in a person's educational history.

Backlog Training Requirements. The objective here is to divide the acquisition workforce into two populations. One population consist of individuals requiring on-time certification training. The remaining portion is the backlog population. There are two ways to become a member of the backlog population. The first is those people who, as of 1 October 1991, were in positions which required a certification level higher than the one currently obtained by the individual in the position. This portion of the backlog population only decreases over the aging process. The second way to become a member of the backlog population is through accessions. This only applies to those lieutenants who were accessed into positions requiring a level one certification and after eighteen months, the certification has not been obtained. This entry criteria could allow personnel to be introduced into the backlog population throughout the simulation process. Since APDP regulations only allow people to move into positions in which they are already certified to hold (new lieutenants are the only exceptions), once a member leaves the backlog population, return is not possible. A member leaves backlog status by obtaining the certification level required by his position. Continuous monitoring of the backlog population is a critical statistic in determining the onset of steady state training conditions.

AFIT APDP Training Contribution. To determine the impact APDP has on AFIT, the contributions AFIT makes to the program must first be determined. Through a review of the course schedules of the School of Systems and Logistics in conjunction with a review of APDP certification requirements in AFR 36-27, it was found that AFIT offers over 75 different courses that could be credited toward certification. Additionally, AFIT's SYS100, SYS200, SYS225 and SYS400 exists as one the primary sequences of achieving levels one and two certification. The wide spectrum of specialty courses taught by AFIT in support of APDP also increase the probability of acquisition workforce personnel attending AFIT.

Current Capability. This requirement is to identify the different modes in which an individual could be credited a course offered by AFIT. Some courses are offered by several modes such as correspondence, in-residence, total course credit examination, and/or on-site facilitator. In addition to determining the different modes, the expected number of students per course offering for a given mode is also needed. Both portions of this objective were estimated using historical data provided by the School of Systems and Logistics administrative office, AFIT/LSA. This research assumes yearly replication of training modes per course. Capacity per mode is an input variable to the simulation and will be varied during output analysis.

New Training Requirement. The purpose of this objective is to identify a percentage range of the acquisition population that will utilize AFIT for APDP certification. Many of the courses required for certification have equivalencies which are offered at different DoD institutions. Since there is no true way to find the exact percentage of the population which will attend AFIT, minimum and maximum bounds on the population are calculated. The maximum bound assumes that every person seeking certification during the aging process will utilize AFIT resources. The minimum bound assumes an equal probability of attendance between the different institutions which offer a particular course. To calculate the minimum bound, the number of course equivalencies must be established. This information was obtained for the APDP Educational and Training subpanel. The Educational and Training subpanel is responsible for all issues concerning educating the acquisition workforce. Using the information provided by this subpanel, a uniform distribution is used for each person requiring a particular course to channel AFIT its portion of the course population. For example, if 100 people required SYS100, and there are 5 agencies which offer equivalent courses, 20 students would be channeled to AFIT. After the initial calculations using the minimum and maximum populations have been computed, sensitivity analysis will be conducted to monitor the course demand behavior between the minimum and maximum bounds.

Steady State Conditions. The purpose of this objective is to determine the time frame in which the initial backlog training demand ends and steady state training begin. At the end of each simulation year, a simulation event emulating the certification screening process is performed. A check is made for each individual to determine if that person has obtained the required certification of his position. If so, the person is remove from the backlog population. Similar checks are made after eighteen months for new accessions. Closely monitoring the backlog population will provide the time line for the onset of steady state on-time training. This assumes that steady state can be reached with AFIT's current methods of instruction and level of resources. If steady state conditions can not be obtained, this will also be identified in the analysis of the backlog population.

Summary

This research encompasses a DoD-wide movement to increase the job proficiency of the acquisition workforce. The focus of the movement is education, training, and experience. One of AFIT's many roles in support of the APDP is to provide educational support to the acquisition community. Because of the numerous agencies involved in implementing the APDP, several trips had to be made to formulate and quantify the objectives necessary to complete this research. Agencies such as HQ USAF, HQ AFMPC, HQ AFSC, and others were all integral parts in formulating this research. Now that the methods used to achieve each critical milestone of this research have been explained, the simulation model used to analyze the APDP process will now be discussed.

IV. Model Formulation

Introduction

This chapter outlines the construction of the SLAM simulation model used to analyze the impact of APDP certification requirements on AFIT. The first section discusses the assumptions of the model. The remaining sections discuss the slam and fortran modules used in the simulation model. The emphasis of this chapter will be placed on the logic used in the simulation model to emulate the APDP process. This discussion will explain the function of the model by detailing each fortran subroutine and major slam function.

Model Assumptions

No Population Cross Flow. Since the goal of APDP is to groom an acquisition workforce of educated, trained, and experienced personnel, job assignments to non-acquisition positions are not likely. The new pipeline structure of the acquisition workforce encourages cross flow within the different acquisition career fields, but does not foster career broadening outside the realm of acquisition. Hence, this model assumes all accessions will arrive as new lieutenants and no attrition will be suffered from total career cross flow.

Input Data. The simulation model utilizes a "snap shot" view of the Air Force since the data retrieved only represents that moment in time. This model assumes that the data extracted from the Air Force manpower and personnel databases is correct. Extensive error checking on each data field was included in the fortran routines to protect against invalid range or type data. However, erroneous data entries of the correct type and relevant range are assumed to be correct. For example, the rank data entry "3LT" for "2LT" would be identified as bad data and written to file stating the position, line, and data element of the error. Nonetheless, the rank data entry "CAPT" for "MAJ" would be accepted as a valid data entry.

Training Priority. During the first few years of the APDP implementation, the backlog training requirements are expected to cause an above average demand for AFIT courses. This model assumes that those people in the backlog population will be given priority selection for courses over on-time trainees. Furthermore, the backlog and on-time personnel files will be further prioritized by rank. This assumption is based on the time restrictions placed on backlog personnel to obtain the required certification. Although personnel wishing to remain permanently in their current position are not affected, the regular movement of military personnel ensures a large percentage of the backlog will seek an accelerated avenue to gain the required certification. Since APDP has imposed this requirement, course selection officials within individual units must provide a realistic opportunity for backlog personnel to obtain the necessary courses. The priority scheme utilized by the simulation model provides that opportunity.

Certification Requirement. Under the requirements of APDP, each acquisition career field has specific standards for personnel certification. This model prototypes the APDP process and assumes all people will be certified identically under the guidelines of the Acquisition Logistics career field. To make this model more accurate, specific requirements for each career stall need to be implemented. Hence all persons will be certified according to their associated career field. This effort is listed as one of the continued research recommendations.

Model Formulation

The SLAM simulation model used in this research is purely discrete event. With the exception of the initialization process, all process in the model can be categorized as "school events" or "end-of-year events." This model uses extensive FORTRAN code (See Appendix C) to accomplish each event. To maintain a logical order in the presentation of the model formulation, Pritsker's ten stages of The Simulation Process will be used (18:10-11). They are as follows:

1. Problem Formulation
2. Model Building
3. Data Acquisition
4. Model Translation
5. Verification
6. Validation
7. Strategic and Tactical Planning
8. Experimentation
9. Analysis of Results
10. Implementation and Documentation

Although some of these steps were discussed in other chapters, a brief synopsis will be provided again in this section to maintain a logical flow.

Problem Formulation. The new training and educational requirements for certification in the APDP will increase the demand for many AFIT courses. To prepare for the new wave of requirements, AFIT has initiated this research to analyze the APDP course demand and its impact. After visits to several headquarters Air Force organizations, it was determined that this problem could be analyzed using a discrete simulation model to emulate the process of APDP. The problem consist of preparing an initial population representative of the acquisition workforce, determining and implementing the policy of training and personnel advancement each year, and aging the population over a twenty-eight year expected life cycle of a military person.

Model Building. Major Glen Bailey of HQ AFMPC/DPMYA, aided immensely in determining the relationship between the data maintained on the acquisition workforce and the APDP processes requiring simulation. The relationships are graphically expressed in Figures 4.1 and 4.2. Figure 4.1 defines the preprocessing of data necessary to initialize the SLAM model. As mentioned earlier, the acquisition workforce consists of two major populations, a population of positions and a population of people. The first stage is to identify the positions. Using the position number attribute of the position, the personnel can be matched to their associated positions. The combination of these populations construct the "snap shot" image of the acquisition workforce. This population is then broken down into two components, the SLAM entities and an auxiliary attribute matrix. The SLAM entity file provide the initial input to the simulation model. Each individual in the workforce is placed in the appropriate position of Figure 4.2 for the start of the simulation. Additionally, initial statistics are captured such as population totals, certification level totals, position totals, and other critical inputs. The auxiliary attribute matrix is used to enhance file manipulation and locations of entities in the system. It represents an external organization such as AFMPC which may require a copy of an individual's records for review. During special events such as promotions, attrition, etc., this matrix is used and then the entity is located in the system for removal or update.

The second diagram, Figure 4.2, represents the flow of personnel throughout their career development. All accessions

Data Flow Diagram

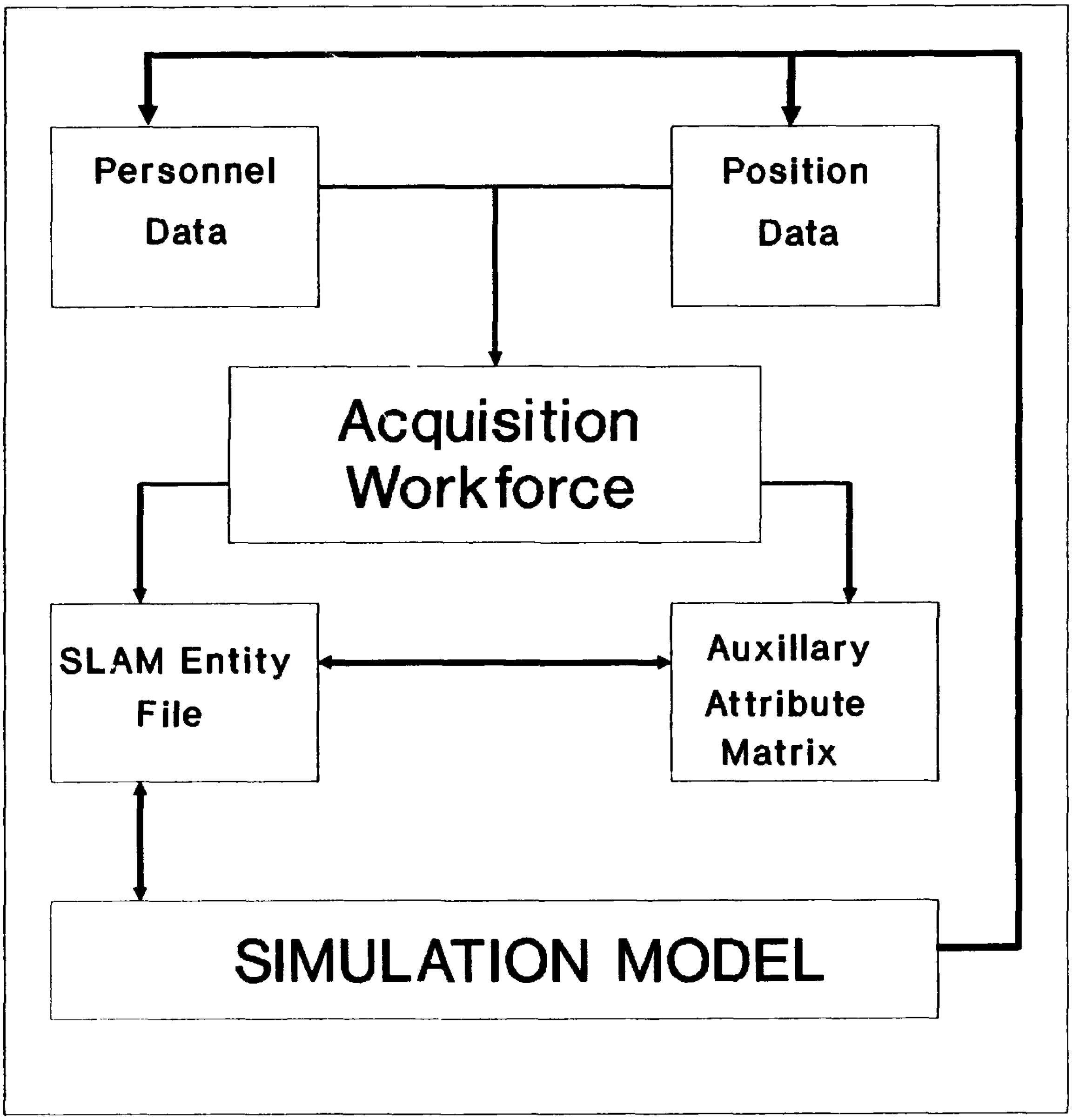


Figure 4.1

APDP Simulated Processes

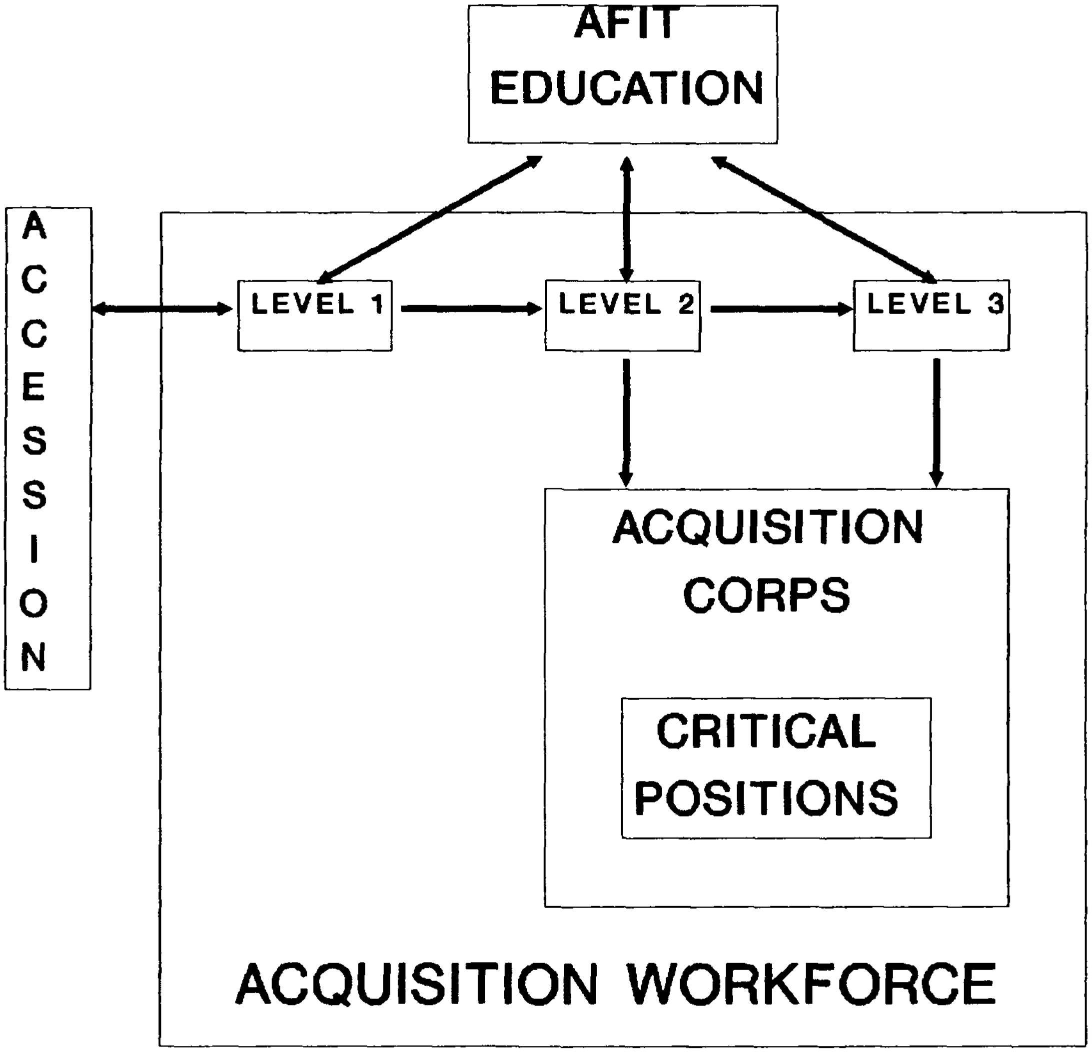


Figure 4.2

are directed to level one positions. Each level has a two-way path to AFIT for training. Throughout the year, personnel attend AFIT for certification courses and are returned to their work positions. At the end of the year, personnel certified for new levels may be moved to a position at the next higher level. As mandated by APDP, only levels two and three personnel are eligible to enter the acquisition corps. Other criteria such as rank, and experience are also required for corps entry. This model only accounts for educational and rank requirements. Also mandated by APDP, only corps members are eligible to serve in critical acquisition positions. These positions are considered to have the highest level of responsibility and require the most promising individuals of the acquisition corps.

Other than training TDYs to AFIT, no entity movement occurs until the end of the year. At that time, all APDP screening processes are conducted, attrition, promotions, and aging occurs. The workforce is then emptied back into the workforce file as displayed earlier in Figure 4.1. End of year statistics are collected and educational certification requirements evaluated. This stage closes the loop for the aging process during career development.

Data Acquisition. As mentioned in the previous chapter, all data was collected from HQ AFMPC. Fortran programs were written to convert character data to integer categories and to filter out bad data. Erroneous data having entries in the relevant range of the data field, are considered to be good data. These occurrences are considered to be minimal and thus having no real impact on the simulation model.

Model Translation. This stage will detail the functionality of each FORTRAN subroutine and major SLAM function. Modules will be presented and explained in the order of their occurrence in the simulation code, and not according to their input to the simulation model.

Program DATA. This program exists as an independent program to read and prepare the initial data retrieved for the AFMPC database. As alluded to in the data acquisition stage, this program extracts inputs from the position file, and searches the personnel file for the person filling that position. If a position is not filled, that is also determined and marked in a attribute called "STATUS" for that position. All position and personnel attributes are read in as character data, checked for validity, converted to integer categories and written to a consolidated acquisition workforce output file. Program DATA also creates data which is currently missing from the AFMPC database but is critical for the simulation model. These elements are the required certification level for each position record and the acquisition corps membership designator for each personnel record. This data was created within the rank and educational guidelines of DoD manual 5000.52. Once the next version of the AFMPC database software is released in June 1992, these constructed elements will not be necessary. The next software release will implement the new standardized requirements for tracking acquisition workforce position and personnel information according to DoD Instruction 5000.55. After all the position and personnel data has been sanitized and consolidated into the acquisition workforce file, the program captures the following initial population statistics:

1. Total number of acquisition positions
2. Total number of critical acquisition positions
3. Total number of filled positions
4. Total number of filled critical positions
5. Total number of Acquisition CORPS positions.
6. Starting population of positions at each certification level
7. Total number of personnel certified at each level.

These statistics will be compared to the additional yearly statistics gathered during the aging process to make inferences during the results section. Normally, this program would have been included in SLAM as a subroutine, but the run time exceeded 12 hours and would impact the time requirements for running subsequent runs.

SLAM Program Cards. This program is the actual code that starts the program running. Here, simulation duration, file prioritization, and important time persistent variables are initialized.

Program MAIN. Program MAIN serves two major purposes in this model. The first is to reinitialize the

NSET/QSET array to allow for a high number of entities and attributes per entity used during the simulation. The second purpose is to set default output values and call the SLAM execution program.

Subroutine Event(I). This subroutine provides a mapping between the SLAM scheduler for the event calendar and the actual FORTRAN routines being scheduled.

Subroutine INTLC. The main purpose of subroutine INTLC is to schedule the data preparation routine and the initial occurrence of the training courses at AFIT. Additionally, two counters for Selective Early Retirement Boards (SERB) and Specialty courses are initialized. The first counter, ICOUNT, limits the scheduling of SERB boards to only the first five years of the simulation. According to HQ USAF/DPXA, SERB boards have not been scheduled after 1997 and it is expected that they will be discontinued at that time. Hence, I have limited their use after the fifth year of the simulation. The second counter, SPECL, limits the occurrence of specialty courses to twelve times per year. Unlike the other core courses used in the simulation, the specialty courses do not represent a single class. A specialty course is schedule each month to represent a variety of courses offered during the month that could be used for certification purposes.

Subroutine PREP. This subroutine uses the output of program DATA to generate the simulation entities and auxiliary attribute matrix. The auxiliary attribute array is initialized first. Position and personnel attributes are read in from an input file. The first six data elements of the matrix represent position requirements. The remaining 64 elements represent personnel data. After the current data is input, other attributes such as the year and beginning population markers are set. A loop is used to read these values for each person into attribute array and then the SLAM entities are filed into the simulation workforce file #15. The last vital function PREP is to schedule subroutine DEMAND.

Subroutine DEMAND. This subroutine has four basic functions. The first is to compute the passover attribute(69) for each personnel each year during the aging process. The attribute PASOVER, will be used by the attrition routine to remove personnel from the workforce. The second function is to call subroutine ACQLOG which will be discussed next. Output from ACQLOG is used by DEMAND to compute the number of people in the backlog population and the demand for key certification courses each year during the aging process. Key courses include SYS100, SYS200, SYS225, SYS400 and a grand total demand for specialty courses. The third function of DEMAND serves as a check for function two. Course demands are computed above in a discrete fashion at the end of each year.

Additionally, DEMAND utilizes six time persistent variables for the backlog population and the five key courses. The previous method of calculating course demand does not account for the decreases in demand at the end of each course throughout the year. Hence the graph of this output should show a stepping decrease in demand each year. The use of time persistent variable to analyze demand should produce a more accurate pictorial representation of the course demand. Although the discrete and time persistent methods of analyzing course demand are calculated differently, the curves should show similar demand predictions. The last function of DEMAND is the most critical. Each year in the simulation, DEMAND cycles through the course history of each person using subroutine ACQLOG and files that individual in the appropriate waiting queue for a course that is needed for the next level of certification. This filing process is based on the binary attribute(62). If the person is determined to be among the backlog population, attribute(62) = 1, the individual is filed in the priority queue for the particular course. If the person is not a member of the backlog population, attribute(62) = 0, the individual is filed in the on-time training queue for that course. Otherwise, the person does not need any courses (level III certified) and is filed back in the main acquisition population. This process setup the file structure for course attendance selection used by each of the individual subroutines for key courses listed above.

Subroutine ACQLOG. As mentioned earlier, this subroutine is called by DEMAND. The purpose of ACQLOG is to review personnel records each year and assign the appropriate evaluated certification level. Additionally, ACQLOG stores indicators in the attributes of each person (ATRIB(50-60)) to point to specific course deficiencies for level three certification throughout career development. For each person in the workforce, DEMAND stores a copy of that person's record in a temporary array called TEMP. TEMP is passed to ACQLOG for an evaluation of each person's educational history. The educational history of each person is compared to the certification standards of the acquisition logistics career field. Credit for completed courses is maintained in a local requirements array called REQMT. When this array of credits is compared to the list of requirements, the deficiencies are determined. Using this information in each persons records, DEMAND can appropriately file people in the queue of the lowest level course they require for certification.

Subroutine CRSDMD. This subroutine is also called by DEMAND. CRSDMD compiles an output file of yearly statistics for courses required by the acquisition workforce. This information will be used in post analysis for graphs of the results.

Subroutine ATTRIT. The purpose of this subroutine is to provide attrition from the workforce each year of the aging process. Attrition is accomplished by two means. The first is based on the natural attrition rates for each year of service. A uniform distribution is used to distribute the attrition across the population for each year of service. The second method is by promotion passover. If an individual is passed over for promotion two times, that person is attrited from the system. The attrition routine does not account for force reduction due to SERB boards. That routine will be presented later.

Subroutine PRMOTE. This subroutine is used to promote people based on promotion rates for each rank. Below-the-zone, zone, and above-the-zone are also included. Personnel are selected to meet the promotion board by year of service. The expected year of service for promotion to each rank was obtained from HQ AFMPC. A uniform distribution was employed to distribute promotions across the population of people meeting the different promotion boards.

Subroutine ACCESS. This subroutine is responsible for introducing new lieutenants into vacant positions in the workforce. Accessions can only occur to empty positions requiring the rank of lieutenant. The accession rate was acquired from HQ AFMPC. The accession function occurs after promotions, SERBS, attrition, and position updates. This order of events ensures the population of positions is accurate before introducing personnel into the workforce.

Subroutine UPDATE. Subroutine UPDATE performs several major functions in the aging process. This subroutine calls all population manipulation routines such as promotions, attrition, etc. The first function of UPDATE is to empty all course queue into a single workforce file(#15). The second function of UPDATE is to call the population manipulation subroutines. Each of these subroutines require personnel to be in a single file for ease of location. The actual process of manipulating people occur in the auxiliary attribute array and then entities are located and updated. The third function of UPDATE is to update the membership of people in the acquisition corps. This will be necessary prior to filling critical positions since only corps personnel can fill critical positions. This is accomplished by creating a file prioritized by rank (file #13). This file is used as temporary storage for all personnel having an acquisition corps designator set to one (attribute 16 = 1 for membership) and are not currently filling a corps position. Selection to a critical position would emulate a promotion to a position of higher responsibility. The auxiliary attribute array is then searched for vacant critical positions. For each vacant position found, a search is made through file 13 for a person having the position qualifications. Since file 13 is prioritized by rank, the most ranking individual will be selected for the job. If a person does not meet the criteria for the position, he is then stored in file 14 until the search is complete for that position or until a person is found. Once the position is filled or determined to be vacant, all personnel are returned to file 13 in preparation for the next vacant position. Once the search is complete for critical positions, file 13 is emptied and successively filled with people ranking from general to lieutenant who fill positions which they out rank. Subsequent searches of the auxiliary attribute matrix are done by rank starting at general for vacant positions. Each position is filled with the highest ranking person meeting the position requirements. After all position updates are made, new accessions can be made to the vacant lieutenant positions. Hence the next important function of subroutine UPDATE is to call the accession routine. Finally, UPDATE calls subroutine DEMAND to evaluate the yearly training of the workforce, and separate it into the necessary course files again for the beginning of the next year.

Training Course Subroutines. This section combines the explanation of all training course subroutines. Individual routines exist for courses SYS100, SYS200, SYS225, SYS400, and a representation of specialty course training. Within each category, several routines may exist to emulate sequential offering of that course throughout the year. These subroutines use the files generated by subroutine DEMAND. An initial check is made to determine if enough personnel exist to offer the course. Initially, this checks use a variable equal to the total course quota, but can be changed to smaller values for sensitivity analysis. Each course has a priority (backlog) and on-time file of entities to select from. Using a base line of 70% priority opportunity, personnel are selected from the priority file first until 70% of the course quotas are filled, and then the remaining quotas are filled from the on-time training file. If there are not enough priority personnel to fill the 70% quota, all backlog personnel are selected and the remaining quotas filled by on-time personnel. Once a person is selected for a course, the TDY attribute (atrib(70)) is set to one to prevent that person from being selected to any other course during his current training. The entity is then filed into a school file associated with each category of courses above. The last function of each training course is to schedule its own graduation event.

Graduation Subroutines. Like the training courses above, discussion of all graduation subroutines are combined in this section because of their similar function. Only one graduation routine exists for each category of the courses above. The subroutine removes each entity from the course file and checks the school entry time against the length of the school. If an entity has completed the course duration it is removed. A check is made to find the next logical course to take and the entity is filed in that course queue.

Verification and Validation. To verify the model performs as intended, sample data sets were contrived to test the functionality of each subroutine. Additionally, each FORTRAN subroutine was implemented as an independent FORTRAN program and tested prior to integration into the SLAM environment. Since this is the initial year of implementation of the APDP program, model validation must be suspended until data under the new program is available for collection.

Strategic and Tactical Planning. Initially, the primary purpose of this model is to determine the demand for APDP courses on AFIT and the budgetary impact of adjusting training requirements to meet the demand. Once cost data has been appropriately integrated, this model will be instrumental in forecasting course demand and associated budgets. Additionally, this model can be used to analyze the difference between DoD policy for APDP certification issued in DoD manual 5000.52 and Air Force level requirements listed in AFR 36-27. This model can also be employed during yearly budgeting cycles to help justify operational expenses of current course media or the need for additional resources to meet the demand of APDP courses. Strategically, this model has been identified by several Air Force agencies as a tool to support ongoing projects. During the stages of exploratory study, agencies such as HQ USAF/DPXY and HQ AFMC/DPA were among the many that expressed an interest in the research efforts and simulation model contained in this thesis. Because of their contract with SRA corporation, HQ USAF/DPXY can use the model as a tool to validate the efforts of the contractor. Since the model provides yearly analysis of the certification status of the acquisition population, it will be an ideal tool for HQ AFMC/DP to monitor the APDP efforts to increase the proficiency of the acquisition workforce. The new command will have the responsibility to manage over 80% of the acquisition workforce. Because of their Air Force level manpower analysis requirements, HQ AFMPC/DPMYA has also requested a copy of the model. Efforts have recently begun to automate the APDP certification process. The accomplishments in this research and the resulting model will serve as a springboard for those efforts. Implementation of this model under the conditions requested by the organizations above is beyond the scope of this study, but will be listed as a recommendation for further study later in this document.

The final three stages listed by Pritsker, experimentation, analysis of results, and implementation will be discuss in the next chapter.

V. Results and Conclusions

This chapter includes an analysis of the acquisition workforce and some conclusions based on that analysis. As stated in Chapter I of this document, the purpose of this research is to analyze the acquisition workforce and determine the impact of the new APDP educational and training requirements on AFIT. The analysis is broken down into two parts; an analysis of the initial workforce based on the raw data received from HQ AFMPC and an analysis of the future workforce using the simulation model. Due to run-time limitations on AFIT's current computer capability, the future workforce will be analyzed using the simulation model and a representative sample of the total workforce. The following results and conclusions of the analysis are provided with reference to the objectives outlined in Chapters I.

Results

Initial Workforce Analysis. To truly capture the effects of the new APDP educational requirements and the importance of AFIT's role in APDP, an analysis of the initial acquisition workforce was completed using the raw data provided by HQ AFMPC (Military personnel only). The data throughout this analysis represents a "snap shot" of the acquisition workforce as of December 1991. The purpose of this analysis was to dissect the workforce into the backlog and on-time training requirements prior to onset of simulated AFIT training. This analysis was accomplished in two parts. First, the workforce was analyzed by certification level and secondly, by acquisition position type. The results of the analysis by certification level is summarized in Table 5.1.

Table 5.1: Workforce Analysis by Certification Level

	NUMBR POSN	NUMBR FILL	POSN MANNING	ON-TIME REQMTS	BACKLOG REQMTS
LEVEL I	4890	3733	76%	427	3306
LEVEL 2	2156	1716	80%	150	1566
LEVEL 3	2070	1710	82%	177	1533
TOTALS	9116	7159	79%	754	6405

Column two of Table 5.1 provides the number of positions required for each of the different certification levels. Since this information is not yet maintained in the manpower and personnel databases, this column of data is estimated using the guidelines established in DOD Instructions 5000.55 and DOD Manual 5000.52. The required level of certification estimated for each position was based on the authorized grade of the position. All lieutenant and captain positions were estimated as level one positions. Majors and a uniform distribution 50% of all lieutenant colonel positions were given estimates of level two. All remaining positions were estimated at level three.

Column two of the table provides the number of positions filled out of the total estimated at that level. This manning percentage is reflected in column three. During simulation analysis, the estimate of the total manning was rounded to 80% and used during the accession routines throughout the simulation. Columns four and five are subsets of column two. If a position is filled with a person who is certified at the estimated required level of certification, the person is a member of the on-time training population. Otherwise, the person does not meet the position requirements and is a member of the backlog population. Since each person in the backlog populations requires a minimum of one course, column six of the table provide a rough estimate of the absolute minimum course demand facing AFIT at the beginning of 1992. The actual course demand will be computed in the initial stages of the course simulation analysis. A graphical representation of the initial workforce as summarized by Table 5.1 is shown in Figures 5.1 and 5.2.

Now the workforce will be analyzed from an acquisition position perspective. Recall that three type of acquisition positions exist; critical positions, acquisition positions, and developmental acquisition positions. Critical positions are for senior level management personnel and can only be filled by members of the acquisition corps. Acquisition positions are all non-critical positions in the workforce that are designated by the Defense Acquisition Workforce Improvement Act (DAWIA) as a member of the acquisition

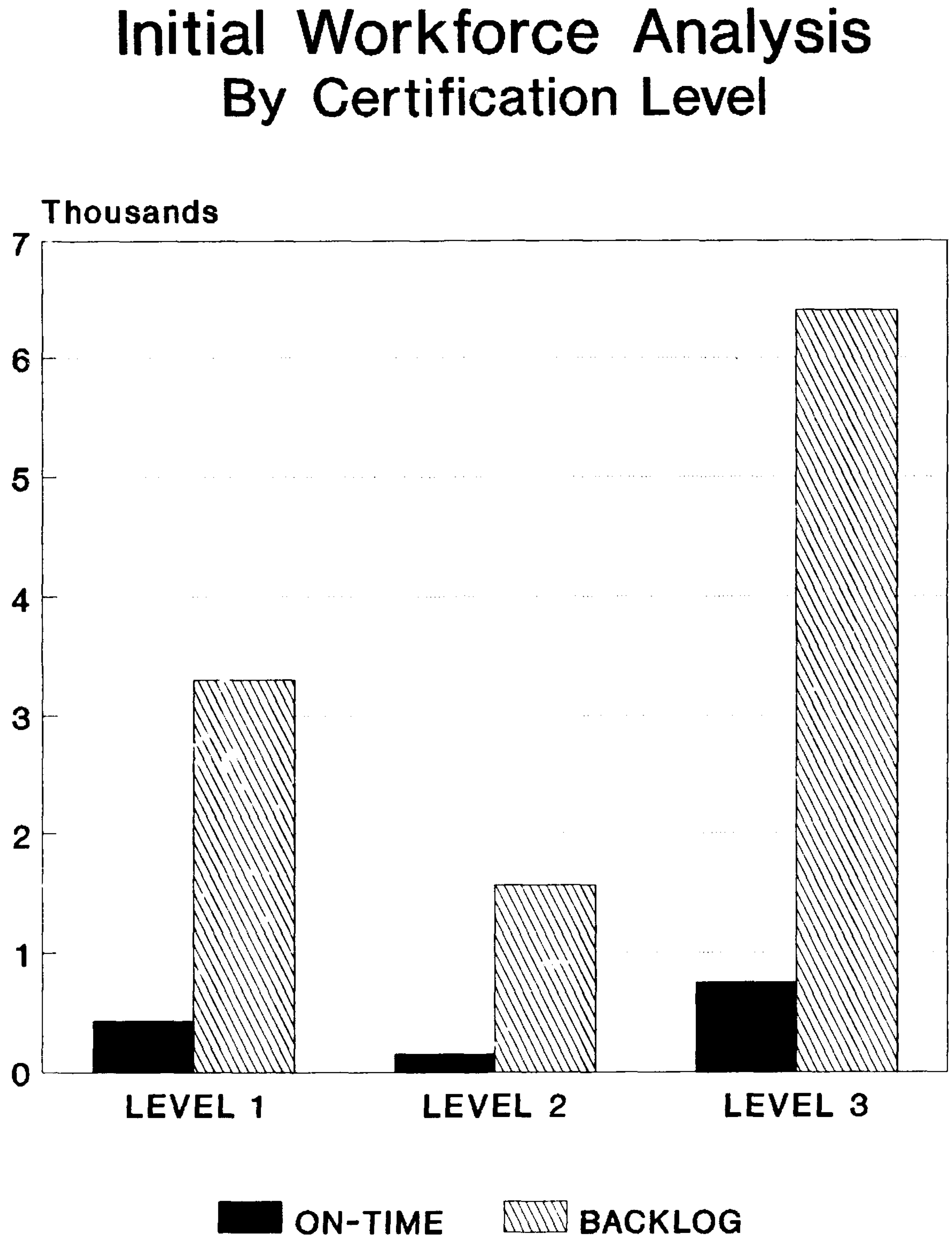

Initial Workforce Analysis
By Certification Level
Thousands
7
6
5
4
3
2
1
0
LEVEL 1
LEVEL 2
LEVEL 3
ON-TIME
BACKLOG

Figure 5.1

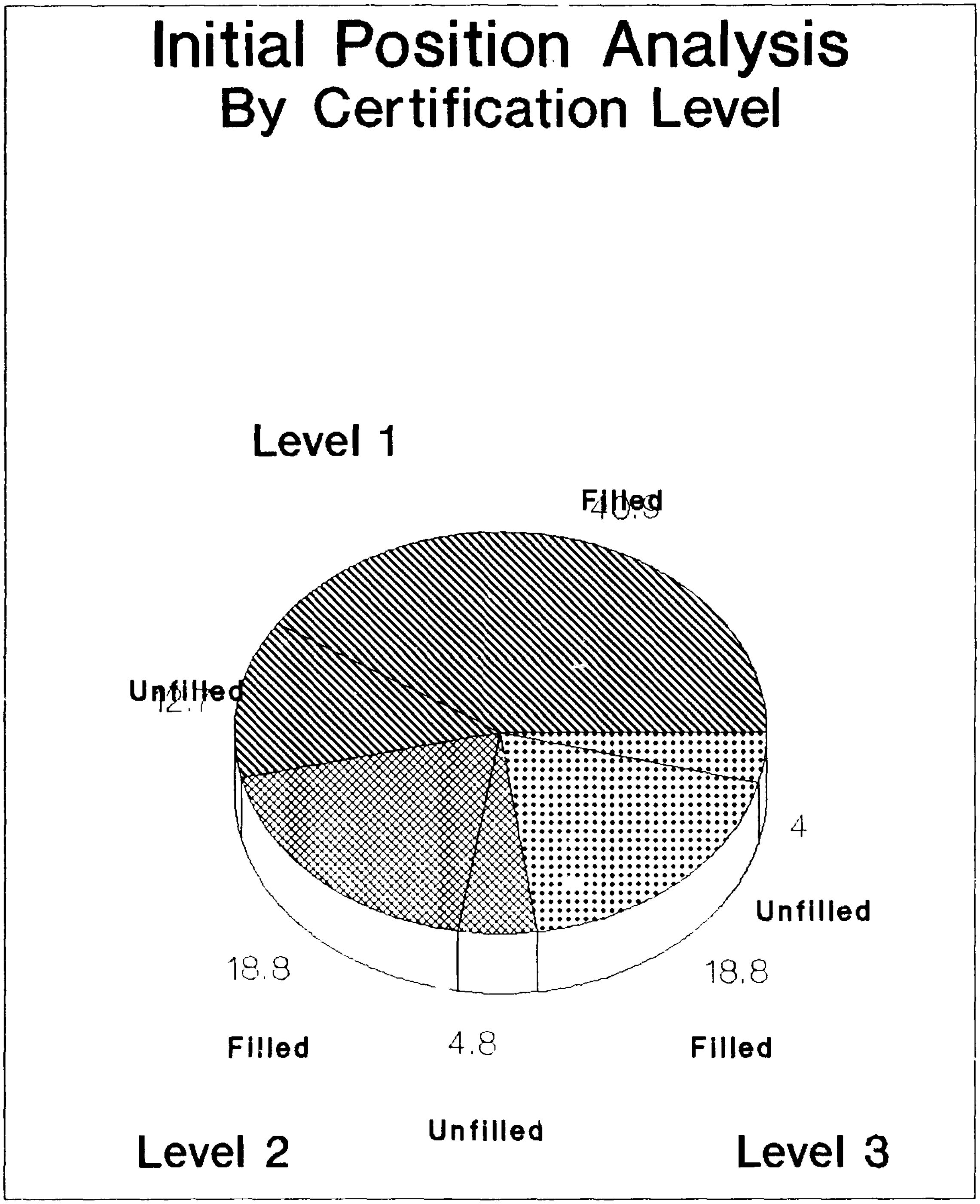

Initial Position Analysis
By Certification Level
Level 1
Filled
40.9
Unfilled
12.7
4
Unfilled
18.8
18.8
Filled
4.8
Filled
Unfilled
Level 2
Level 3

Figure 5.2

workforce. Furthermore, personnel filling these positions will carry one of the acquisition designated career field codes. Developmental acquisition positions are those which regularly interact with acquisition activities, but do not carry an acquisition workforce position designator. Table 5.2 summarizes the initial workforce analysis for the different types of acquisition positions.

Table 5.2: Workforce Analysis by Position Type

	NUMBR POSN	NUMBR FILL	POSN MANNING	ON-TIME REQMT	BACKLOG REQMT
CRITICAL	1565	1265	81%	178	1087
ACQUISITION	7050	5450	77%	560	4890
DEVELOP	501	444	89%	16	428
TOTALS	9116	7159	79%	754	6405

Table 5.2 emphasizes the large distribution workforce personnel in backlog status. The crucial element of this table is the population of backlog people in the critical positions. Their impact is more substantial since their duties contain the highest level of responsibility in the procurement process. Again, it is quite apparent that the initial state of training and educational status of the acquisition workforce is quite below the standards set by APDP. A graphical representation of Table 5.2 is shown in Figure 5.3. Although the greatest impact on the procurement process may be captured in the backlog status of the personnel filling critical positions, the greatest impact on AFIT will

Initial Workforce Analysis
By Acquisition Position Type

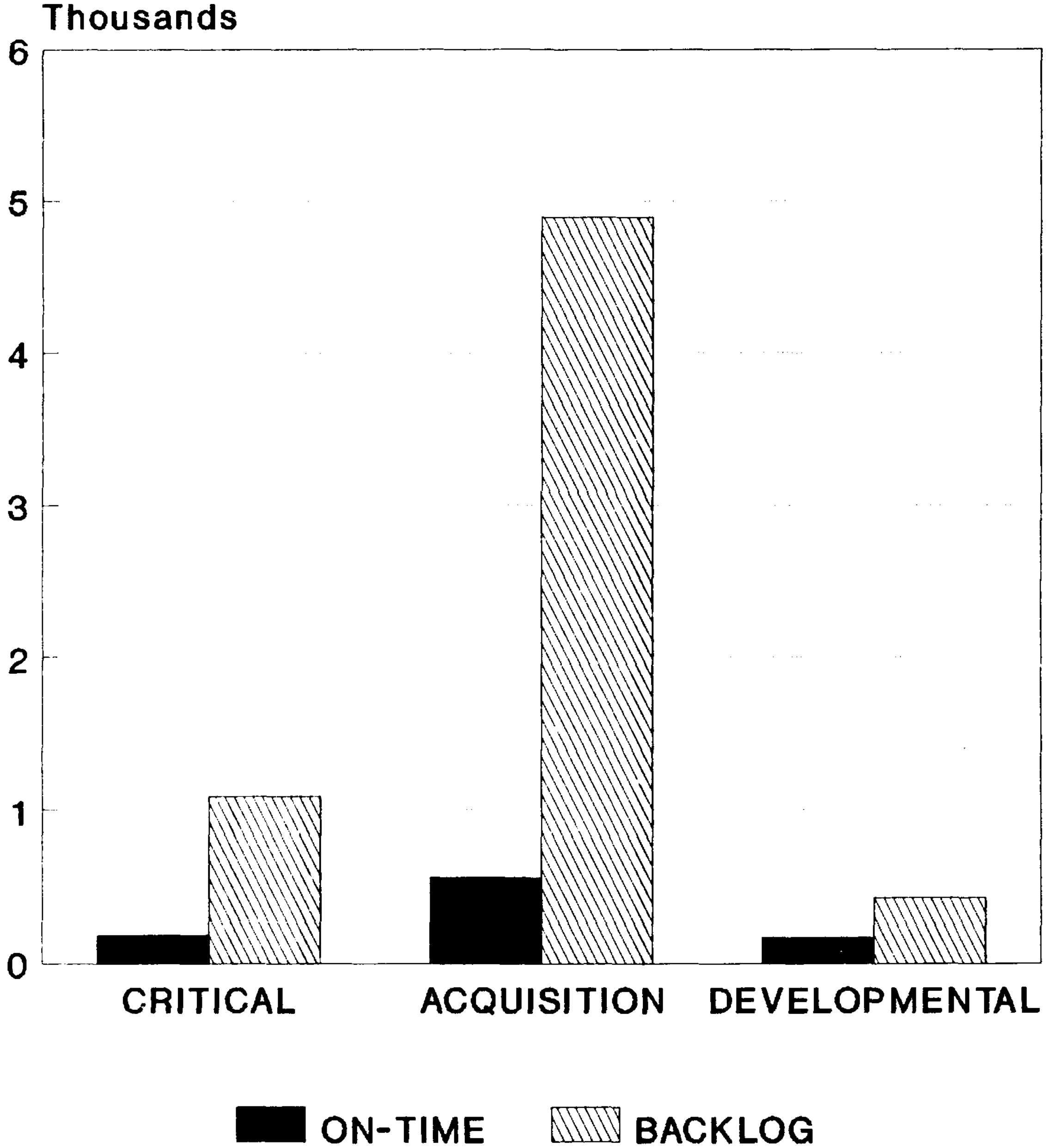


Figure 5.3.1

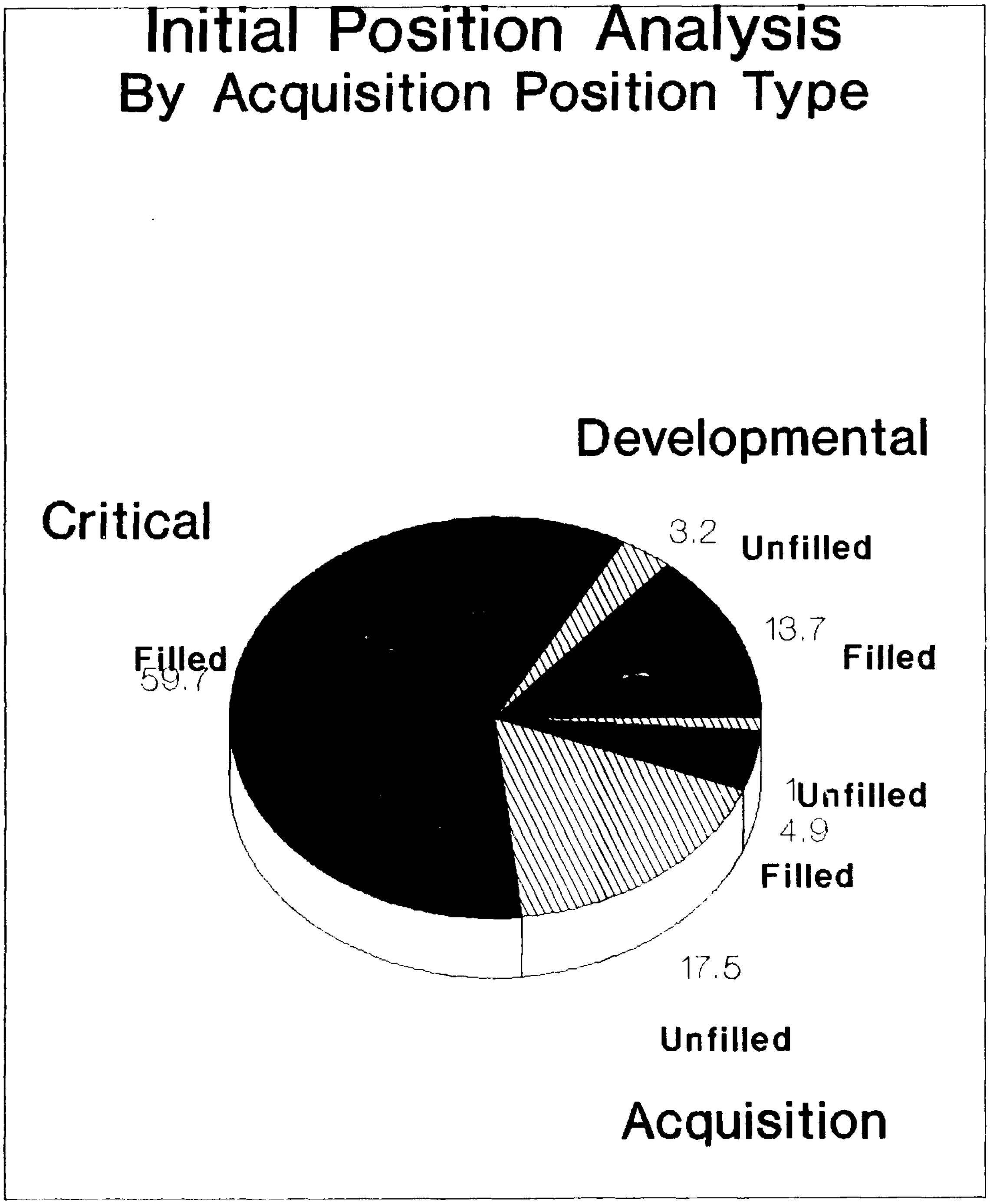

Initial Position Analysis
By Acquisition Position Type
Developmental
Critical
3.2
Unfilled
13.7
Filled
Filled
59.7
1
Unfilled
4.9
Filled
17.5
Unfilled
Acquisition

Figure 5.3.2

stem from personnel filling acquisition positions since the vast majority of the backlog population are filling these type positions.

Simulation Model Analysis. Since the APDP process requires constant evaluation of each individual in the acquisition workforce, the original approach to analyzing the impact of APDP on AFIT was to include each individual in the simulation model during analysis. This approach identified a severe limitation to the model as well as AFIT. The model has two representations of the acquisition population during the simulation. One representation is the actual entities in files and represent acquisition personnel filling positions in the workforce. The second representation is a matrix containing a duplicate of each person in the entity files and represent the record keeping function of organizations such as AFMPC which maintain information on the people and positions in the workforce. This structure was implemented to maintain independence between the entities and their associated attributes since the movement of people to different positions and training is independent of functions which occur at record keeping organizations such as promotions boards and selective early retirement boards (SERB). Therefore, keeping the two representations of the acquisition populations allows attributes of entities to be updated during simulation without necessarily removing the entities from their current files.

After effectively integrating both representations of the acquisition population in the simulation model, the initial status of he acquisition workforce was analyzed again prior to the start of simulated twenty-eight year cycle of personnel training and career development. This provided an independent check for the information found during the data analysis done by the FORTRAN program. This is where the limitation of the model was discovered. After 5 days of dedicated CPU time on a SUN SPARC Station II containing 28 MIPS of processing power, only 7 years of simulation was complete. The SUN stations are currently the fastest source computing time available at AFIT. Hence it can be concluded that the model has limited usage unless greater processing power is obtained. configuration.

To bypass this obstacle, a representative sample of the acquisition workforce was extracted from the data provided by AFMPC and applied to the model. The new population contained 1000 positions and the associated people in the filled positions. The extraction was done using a FORTRAN program based on a random number generator and a uniform distribution between zero and one. This program was written by Captain Lisa Belue in her thesis effort, *An Investigation of Multilayered Perceptrons for Classification.* The details of the code extracted from her thesis can be found in Appendix K. To maintain credibility of the conclusions that will be drawn from analysis of this sample population, analysis will be done to show that the sample population contains characteristics of the initial population analyzed in table 5.1. The results of this analysis is contained in table 5.3 below.

Table 5.3: Sample Workforce Analysis by Certification Level

	NUMBR POSN	NUMBR FILL	POSN MANNING	ON-TIME REQMTS	BACKLOG REQMTS
LEVEL I	554	426	77%	47	390
LEVEL 2	227	177	78%	13	164
LEVEL 3	219	169	78%	17	152
TOTALS	1000	772	77%	77	695

Comparing Table 5.1 to Table 5.3, it can be concluded that the sample population is indeed representative of the true population based on an analysis with respect to certification level. To complete the analysis of the sample population, the sample workforce will now be analyzed with respect to acquisition position type. A summary of this analysis can be found in Table 5.4. Once again it can be concluded that the sample population is representative of the actual acquisition workforce.

Table 5.4: Sample Workforce Analysis by Position Type

	NUMBR POSN	NUMBR FILL	POSN MANNING	ON-TIME REQMT	BACKLOG REQMT
CRITICAL	155	116	75%	17	99
ACQUISITION	793	613	77%	60	553
DEVELOP	52	43	83%	0	43
TOTALS	1000	772	77%	77	695

Simulation Results. Using the small sample size, ten simulation runs were generated to analyze course demand and behavior of the backlog population. The first simulation run utilized AFIT's 1991 training capability. The next five runs utilized increments of 50% increases to the baseline capability by increasing course quotas. Hence, the sixth run contains a training capability of 250% increase of the 1991 training baseline. The last four runs were done to analyze possible choke points in the system. To appropriately adjust the training capability for the sample workforce population, the percentage reduction in population was applied to the allowable course quotas for each course. All course quotas were provided by AFIT School of Logistics Administrative office, AFIT/LSA. The baseline quotas are shown in Appendix G. Two categories of statistics were capture during each simulation run. Time persistent variables were used to capture the behavior of the course demand over the entire 28 year time period and discrete variables were use to summarized course demand at the end of each year. The results of the simulation runs for time persistent variables are shown in Appendix G. Since the unit of increment during the simulation was days, the Means column in the table of time persistent statistics is interpreted as the expected demand for any given day during the 28 year cycle. The Minimum column represents the initial state of the system prior to start of the simulation run. The Maximum column represents the amount course demand present at the beginning of the first simulation year after the initial workforce data has been analyzed by subroutine PREP of the model. Because of the "averaging effect" experienced when collecting statistics continuously over time, the details of the behavior of course demand and the backlog population is obscured. However, these results can be obtained from the discrete variables.

The backlog status and course demand were analyzed at the beginning of each year. Graphs demonstrating this output for each run can be found in Appendix J. The graphs demonstrate the behavior of each variable as it decreases over years 1992 to 2019. It is clear that each of the training courses as well as the backlog population demonstrate a exponential decrease over time until the steady state zone is reached. Steady state is indicated when the bars on the chart cease decreasing by any significant amount. From the graphs contained in Appendix J which display the results of each year, the time frame for steady state training can be estimated. Using the current baseline training capability, the onset of steady state training is expected approximately year 2014. When this capability is increased by 50%, steady conditions are moved to approximately year 2013. Further increasing capability to 100% of the baseline causes steady state to start at year 2010. The numerical computations displayed in the graphs can be found in Appendix I. Consider the graphs contained in figures 5.4. This graph represents the behavior of the backlog population over the range of experimental training capability increases. After analyzing the backlog population status displayed in Figure 5.4, it can be determined that the backlog population is not responsive to the percentage increases in training capability. Thus some suspicion of the training methodology arises from the analysis of the backlog population alone. Now consider the graphs contained in Figure 5.5. This graph analyzes the demand for course SYS100 over the range of capability increases. It is obvious from this graphs that the demand for SYS100 is decreasing tremendously with each increment of training capability. This behavior is also observed for SYS200 course demand in Figure 5.6. However, for SYS225 the results are quite different. From Figure 5.7, it can be concluded that the demand is responsive to training increases for the first 6 years. However, after year 1997, the behavior is not responsive to capability increases. At this point, it is beginning to become clear why the backlog demand does not seem responsive to training increases. It appears that there exist a "choke point" in the system. Recall from Chapter 4, the model assumes personnel will progress systematically through the course requirements for certification. This approach to certification is displayed in Figure 5.8. According to the

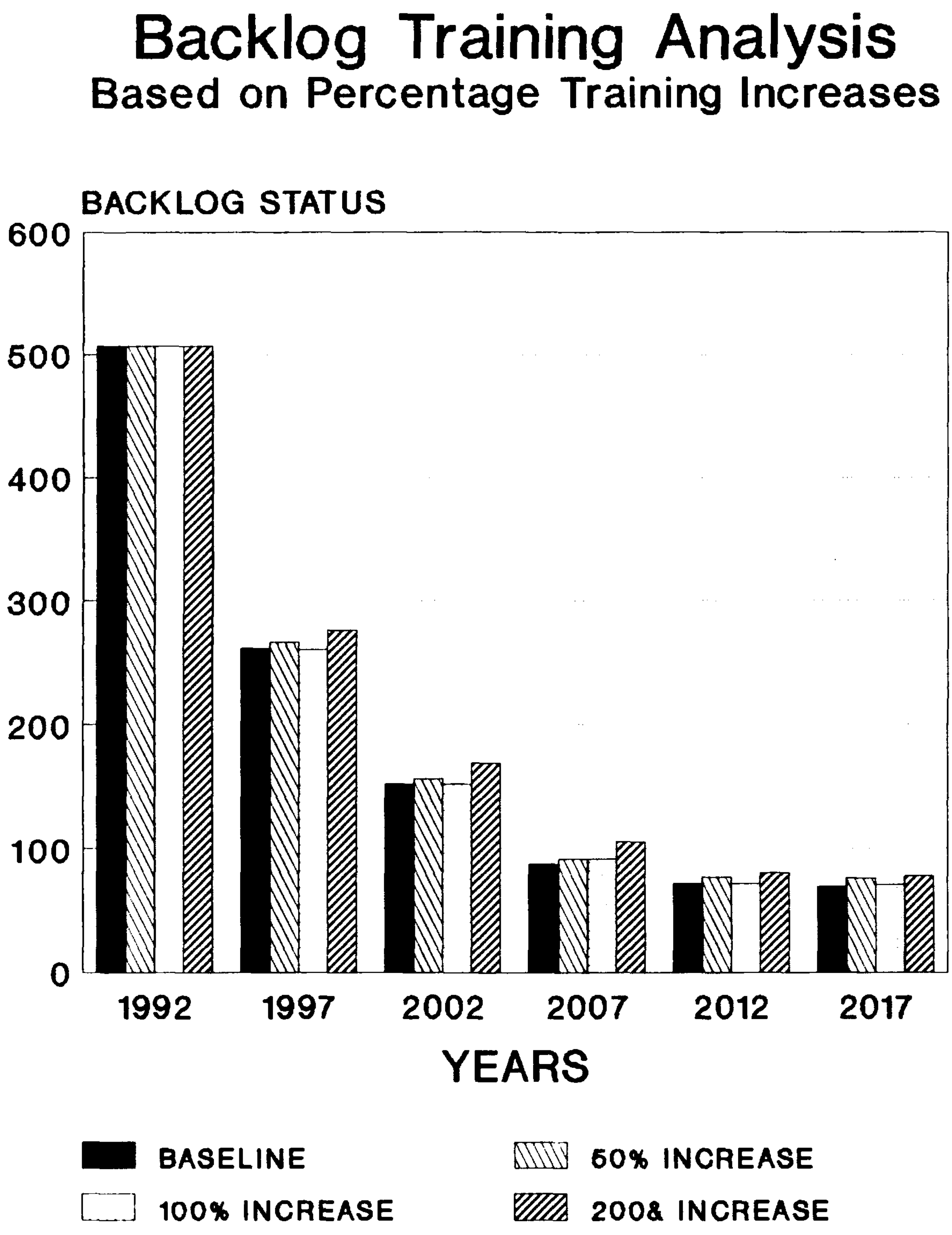

Backlog Training Analysis
Based on Percentage Training Increases
BACKLOG STATUS
600
500
400
300
200
100
0
1992
1997
2002
2007
2012
2017
YEARS
BASELINE
60% INCREASE
100% INCREASE
200& INCREASE

Figure 5.4

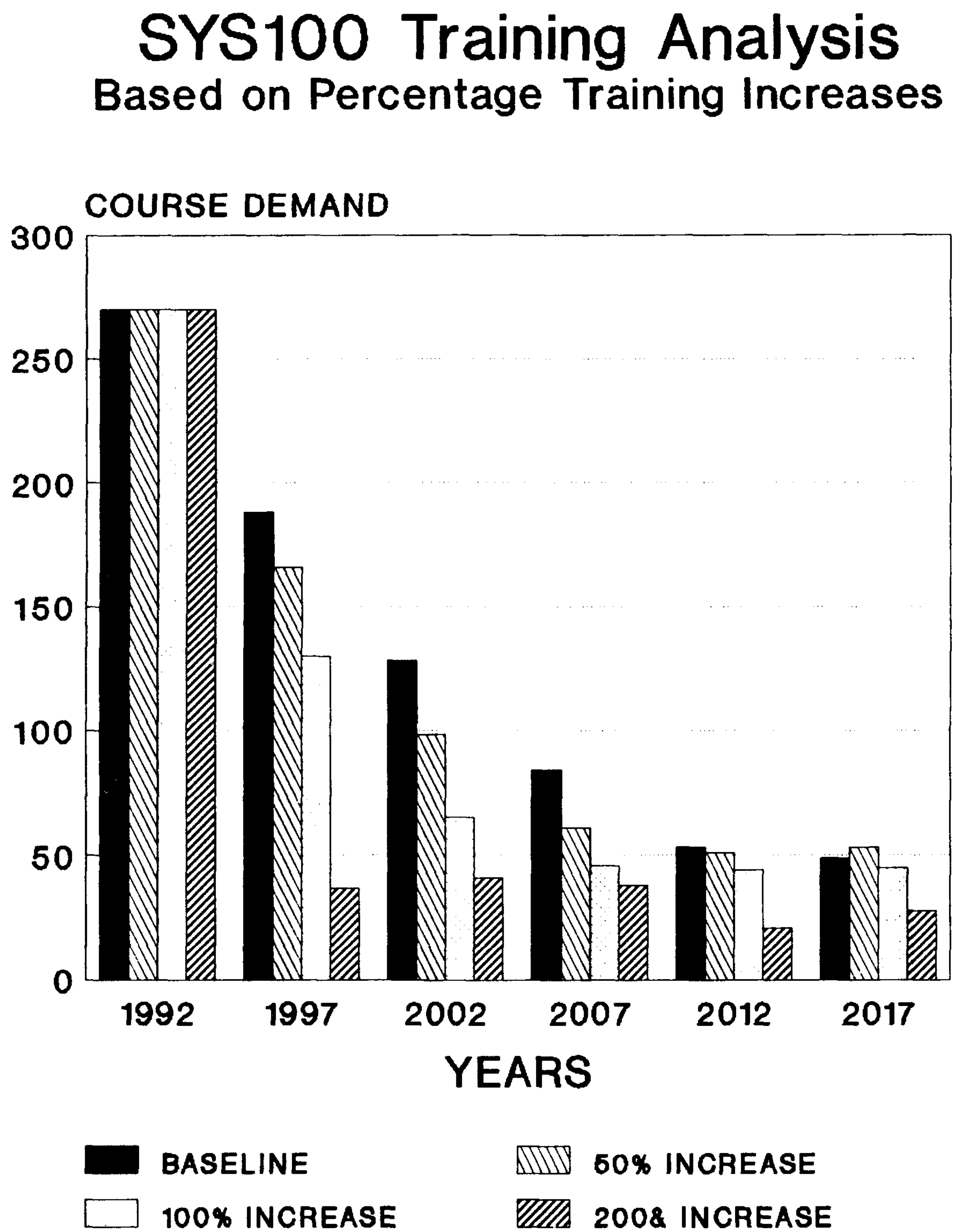

SYS100 Training Analysis
Based on Percentage Training Increases
COURSE DEMAND
300
250
200
150
100
50
0
1992
1997
2002
2007
2012
2017
YEARS
BASELINE
50% INCREASE
100% INCREASE
200& INCREASE

Figure 5.5

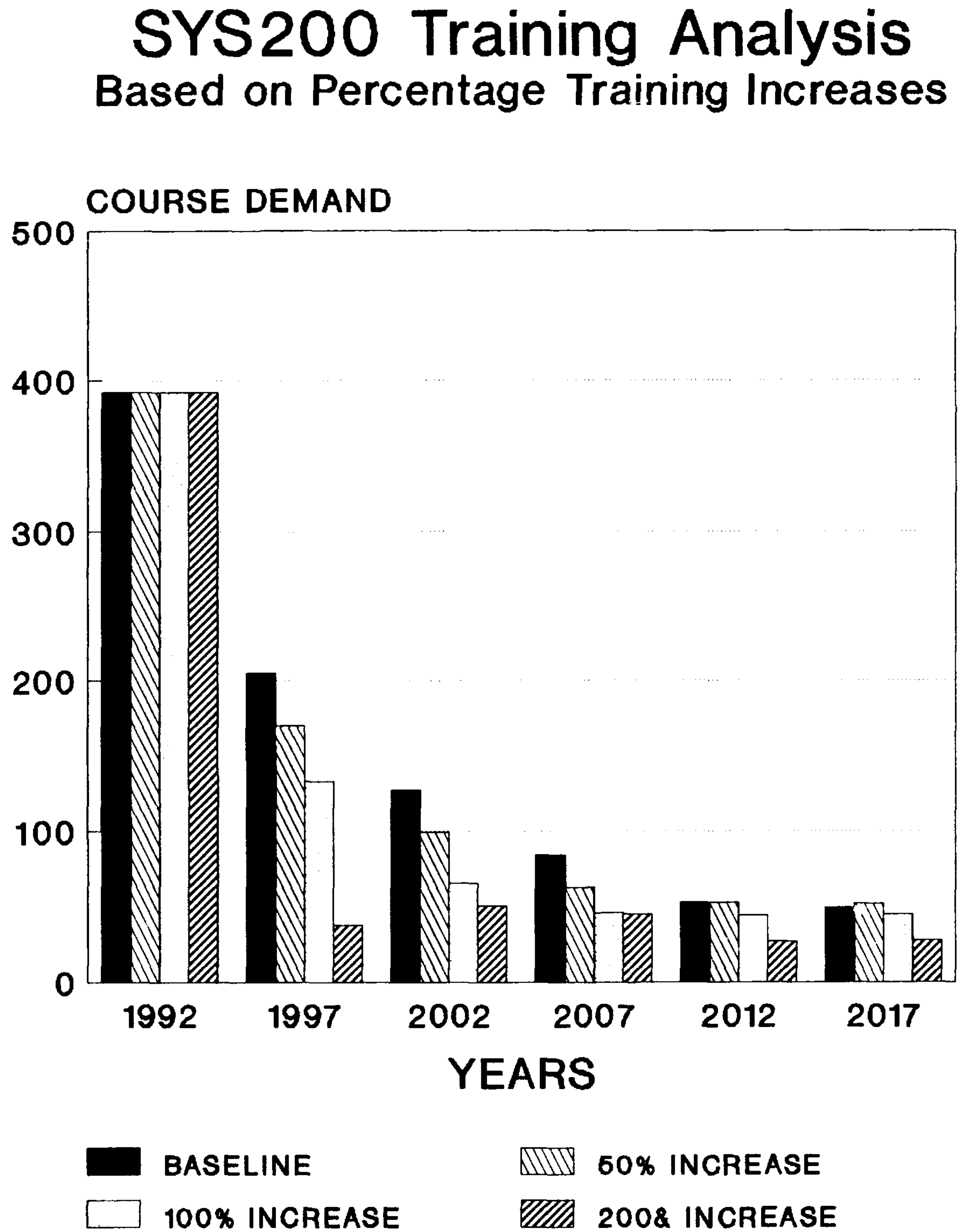

SYS200 Training Analysis
Based on Percentage Training Increases
COURSE DEMAND
500
400
300
200
100
0
1992
1997
2002
2007
2012
2017
YEARS
BASELINE
50% INCREASE
100% INCREASE
200& INCREASE

Figure 5.6

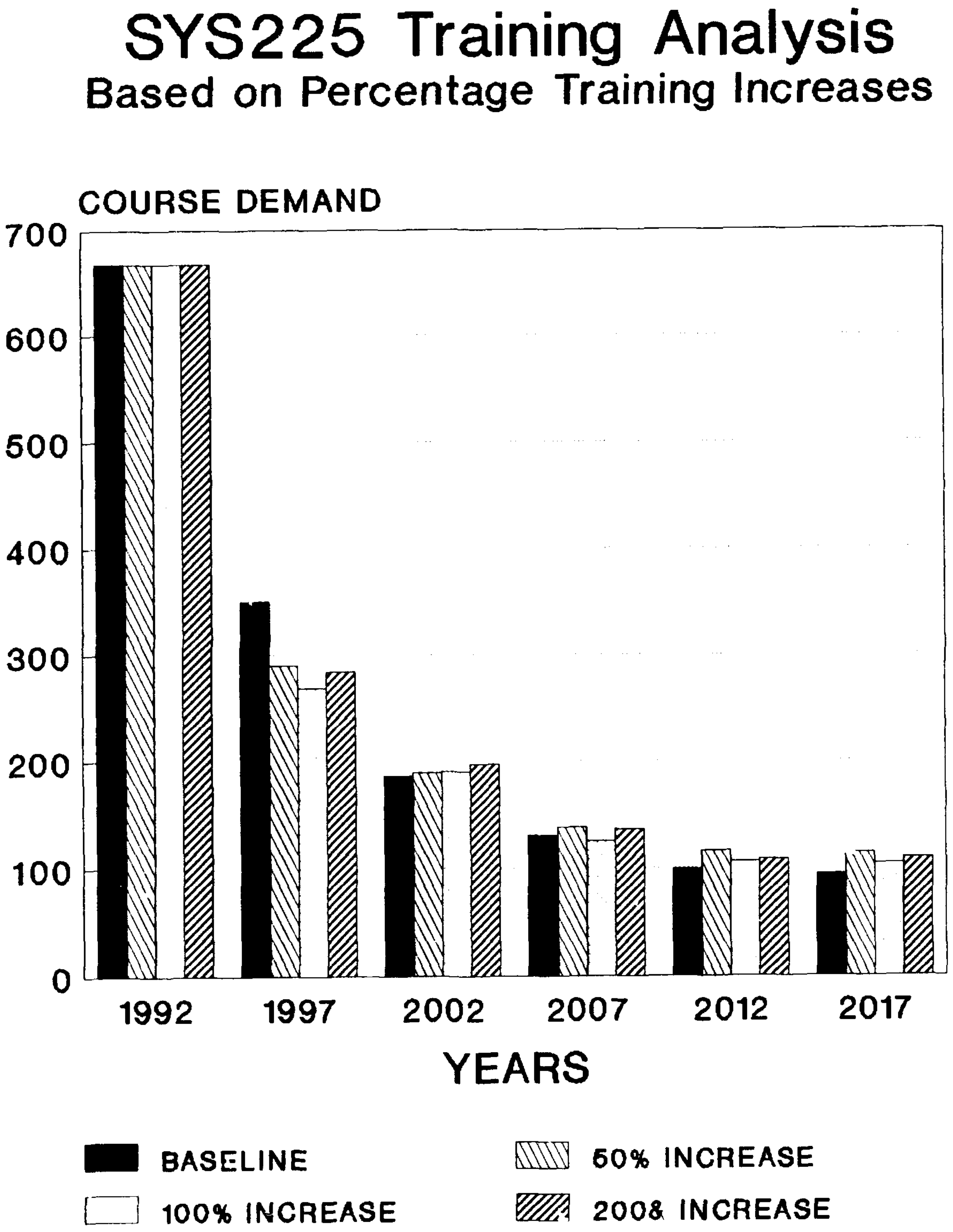

SYS225 Training Analysis
Based on Percentage Training Increases
COURSE DEMAND
700
600
500
400
300
200
100
0
1992
1997
2002
2007
2012
2017
YEARS
BASELINE
50% INCREASE
100% INCREASE
200& INCREASE

Figure 5.7

Course Sequencing

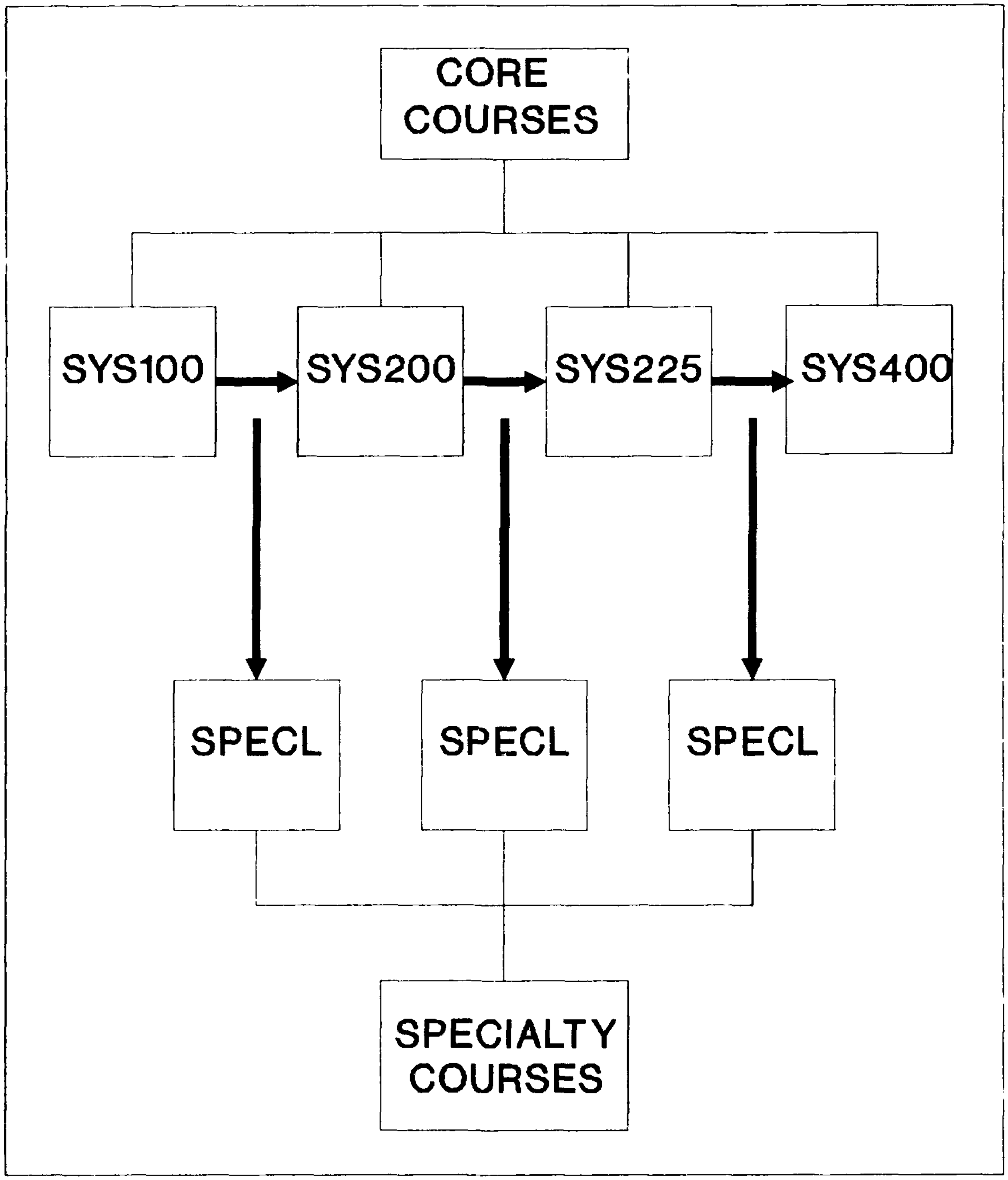


Figure 5.8

Air Force regulation on training courses, AFR 50-5, the courses in Figure 5.8 are prerequisites for each other such that the expected logic for attendance is as displayed. From the analysis thus far, it appears that the combined flow out of SYS100 and SYS200 is not sufficient to consistently fill course SYS225. Hence, the model would cause cancellations for the courses under this conditon. Since the current course quota is sufficient to sustain the flow of demand, any increase in capability without increasing the flow into the queue for the course would be useless. The initial experimental runs that generated these results assumed an equal increase in training capability for each course. Since SYS225 already had excess training capacity to handle increases in training capability for SYS100 and SYS200, the increases made during each run was not necessary. Hence, analysis indicate that course increases should not necessarily be equal but instead adjusted only when the flow into the course demand queue is greater than the course quota. Since backlog people requiring SYS225 for certification can only leave the backlog population at the rate of flow out of SYS225, the backlog population demonstates the same responsiveness to training increases as SYS225. This assumes that SYS225 has the lowest flow rate. A generalization to this condition is that the curved traced out by the backlog status responsiveness to training increases will demonstrate the behavior of the course having least throughput. Since people flow systematically through the certification process, the initial "bottle neck" affects each of the courses following it. This phenomenon also explains the course demand behavior displayed in Figures 5.8 and 5.9 for SYS400 and Specialty courses. To confirm this suspicion, additional runs were conducted using different course quotas to relieve the suspected "bottle neck" found in SYS100 . Since the existing quotas for SYS225 is capable of handling the output of courses SYS100 and SYS200, only SYS100 and SYS200 quotas will be increased. By only increasing the training capability at the "bottle necks" in the training system, training throughput is maximized with minimum resourses expended. Previously, the curve demonstrated by personnel leaving the backlog population was unresponsive to training increases. Using the new training capability with increased flow through the Systems 100 and 200 APDP courses (150% increases), Figure 5.11 was observed. This analysis implies that preparation for decreasing the backlog demand should begin with determining the critical flow points through the network of courses required by APDP for certification for each career field in acquisitions. Once the critical points have been established, iterative optimization is done by increasing training capability at that point until it is no longer a critical point. New critical points are then identified. This

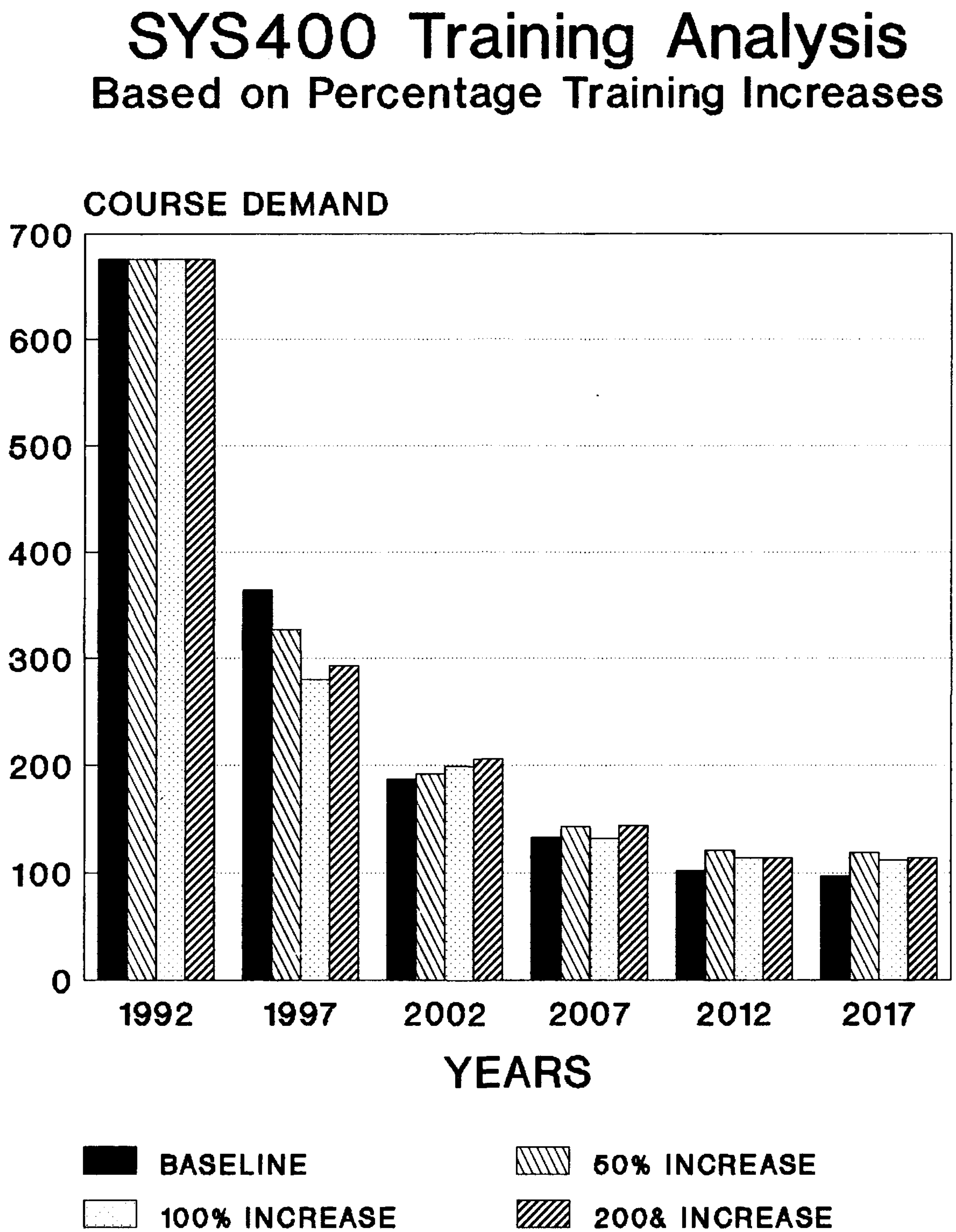

SYS400 Training Analysis
Based on Percentage Training Increases
COURSE DEMAND
700
600
500
400
300
200
100
0
1992
1997
2002
2007
2012
2017
YEARS
BASELINE
50% INCREASE
100% INCREASE
200& INCREASE

Figure 5.9

Specialty Course Analysis
Based on Percentage Training Increases

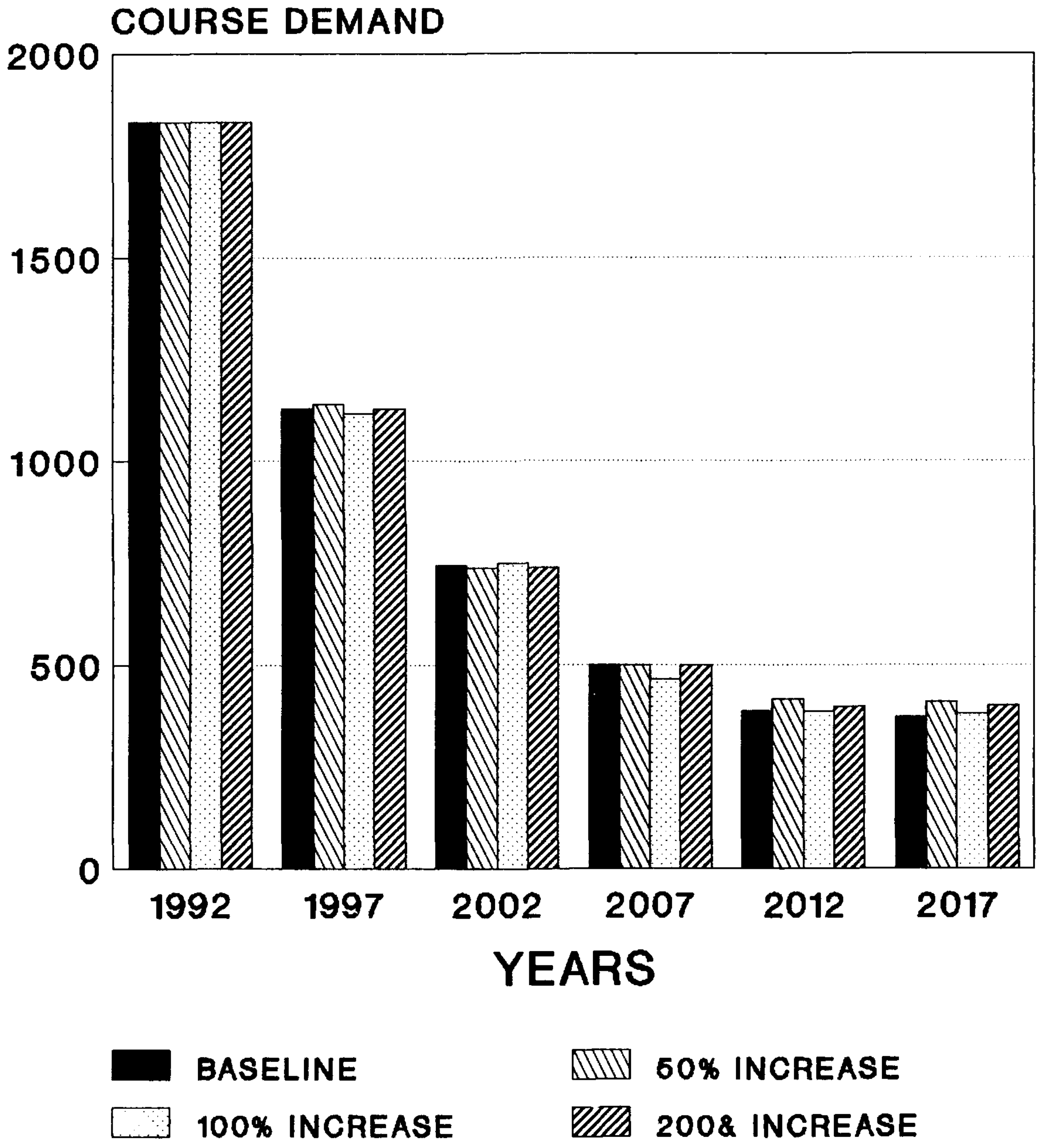


Figure 5.10

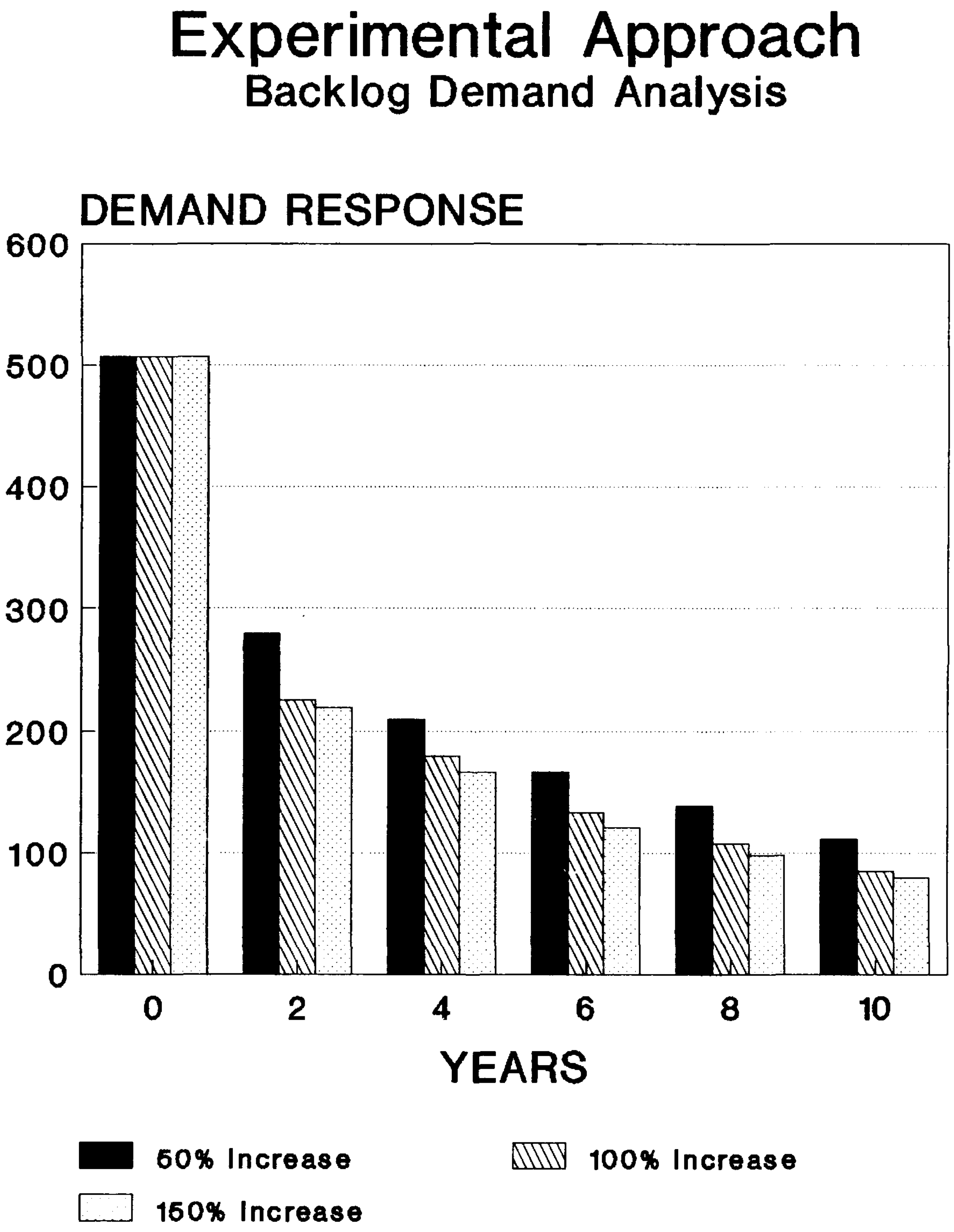

Experimental Approach
Backlog Demand Analysis
DEMAND RESPONSE
600
500
400
300
200
100
0
0
2
4
6
8
10
YEARS
50% Increase
100% Increase
150% Increase

Figure 5.12

process repeats until the desired level of training throughput is accomplished.

Conclusions

Thus far an analysis of the initial workforce using the raw data provided by HQ AFMPC has been completed. Using a prototype of that population, an analysis of the behavior of both the backlog population and the course demand was undertaken. Given that the prototype does behaves like the real world, the following conclusions can be made.

1. By the standards of APDP and the guidelines of the Acquisition Logistics career field, the acquisition workforce is severely undertrained. The vast majority of the population exists in the backlog training status. More specifically, our initial analysis shows that 89.5% of the acquisition workforce is in the backlog status.

2. It can also be concluded from the current training capability that steady state training requirements are not expected until approximately year 2014. This assumes that the prototype model behaves like the true population.

3. Analysis of the backlog population and course demands indicate that SYS100 and SYS200 combine to produce a "choke point" in the certification process for people in Acquisition Logistics career field. In order to minimize impact, training increases should start with increasing course quotas or course offerings to critical areas in the training network until

those areas nolonger constrain the certification process. The iterative process of increasing training capability only at critical points will minimize the cost of increasing training capability. From the analysis of the initial acquisition workforce by certification level, it was shown that the vast majority of the backlog population existed at levels one and three. Hence, analysis suggest initial training increases to reduce the backlog should cocentrate here.

4. Finally, the issue of impact on AFIT must be addressed. The magnitude of the impact on AFIT is directly determined by how AFIT management chooses to attack the backlog population. Aggressive goals which require large increases in training capability will naturally impact AFIT much more severely than passive goals which allow the backlog status to exist for over a decade. Several simulation runs were conducted at different capabilities to analyze the response of the backlog population status. Additional research to incorporate a cost module in the simulation model would be helpful in this iterative process. This topic will be discussed further in the recommendations for further study.

VI. Recommendations

Introduction

Since this research involves a wide variety of DOD efforts to analyze the acquisition workforce, there exist many directions of further study for future analyst, however, recommendations in this chapter will be limited to those that directly impact the problem addressed in this thesis.

Model Formulation. As discussed earlier in Chapter V, this model is severely limited due to data processing time requirements. Assuming that AFIT will not purchase computers capable of rendering this problem ineffective, other coding changes can be made. One recommendation is to dissect each entity into its critical and non-critical attributes. This will relieve the requirement for processing all seventy attributes each time an entity is utilized. Another recommendation is to use a different simulation language or programming language that has better file handling capability. SLAM only allows one priority statement per file. Thus, several files must be used to differentiate between people of the same class, but requiring different priority for processing.

Cost Modelling. To further analyze the impact of APDP on AFIT, the cost of increasing training capability must be considered. Furthermore, AFIT utilizes several methods of instruction for different courses. Each method has its benefits, disadvantages, and associated costs. A recommendation in this area is to first develop a cost function for each method of instruction. The function would contain all the major factors contributing to the expense of offering the course. Secondly, implement the cost functions as a module of the simulation model which provide cost estimates as a function of course demand and training capability. Finally, an automated capability of iteratively optimizing training capability for a given budget can be added. A model possessing these capabilities would be more effective and conclusive in identifying the total impact of APDP on AFIT.

Appendix A

Acquisition Courses

ACQUISITION COURSES				
CRSE #	COURSE TITLE	SIM #	PDS	Hr
SYS100	Intro. to acq. mgt	100	2RX	40
SYS100E	Intro. to acq. mgt test out	101	3KV	40
SYS200	Acq. Planning and analysis	102	2RY	120
SYS400	Intermediate Prog. Mgt.	103	2RZ	80
SYS225	Acq. Logistics	104	WYF	80
SYS128	Adv. Config. Mgt	105	3EM	40
SYS150	Engineering Data Mgt.	106	9AG	40
SYS212	Mission Crit. Cmp. Soft. Proj. Mgt	107	LOB	80
SYS227	Finan. Mgt Weapon Sys. Acq.	108	UGJ	80
SYS228	Applied Config. Mgt.	109	VG6	72
SYS229	Test and Eval. Mgt	110	TR9	64
SYS230	AF Tech. Order Acq. Mgt	111	4ZO	80
SYS362	Cost Sch. Ctrl. Sys. Criteria	112	22H	80
SYS363	Basic Analysis Perf. Measure Data	113	24X	40
SYS370	Defense Data Mgt.	114	EAY	72
SYS420	Laboratory Contract Mgt.	115	EBF	104
SAS001	Basic Sys. Acq. Mgt.	200	4HQ	40
SAS002	Computer Resource Acq. Course	201	8B3	80
SAS003	Sub-Contract Mgt. for Suply Acq.	202	81A	40
SAS004	Weapon Sys Supportability Course	203	9X2	40
SAS005	Adv. Computer Resource Acq.	204	92G	40
SAS006	Intermediate Sys. Acq. Mgt	205	9XQ 9X0	80

ACQUISITION COURSES				
CRSE #	COURSE TITLE	SIM #	PDS	Hr
DSMC-3	Program Mgt. Course	303	AN7	400
DSMC-5	Contract Fin. for Prog. Mangers	304	35B	40
DSMC-6	Contract Performance Measurement	305	QTT	40
DSMC-8	Multinational Prog. Mgt.	306	ZD4 1GZ	40 80
DSMC-9	Sys. Acq. Funds Mgt.	307	OHZ	40
DSMC-10	Mgt. of Software Acq.	308	1Q1	40
DSMC-11	Test and Evaluation Mgt.	309	3CN	40
DSMC-13	Defense Manufactuering Mgt.	310	28N	40
DSMC-20	Technichal Managers Adv. Workshop	311	5J7	40
DSMC-24	Mgt. of Acq. Logistics	312	4X1	40
DSMC-25	Prog. Mangers Briefing	313	9A9	40
DSMC-26	Fundamentals of Sys. Acq. Mgt	314	9A8	40
DSMC-28	Sys. Engineering Mgt. Course	315	HVF	40
DSMC-29	Contract Mgt. for Prog. Managers	316	AN6	40
DSMC-30	Executive Mgt. Course	317	APC	120
DSMC-31	Selected Acq. Report Course	318	JEJ	40
DSMC-32	TQM Course	319	L38	40
DSMC-33	Adv. International Mgt. Workshop	320	N85	40
DSMC-34	Sys. Acq. for Contract Persons	321	LRB	80
ALC001	Deputy Prog. Mangers for Logistics	400	7PO	80
LOG032	Relaiability Centrd. Maint. Anlys.	500	OPG	40
LOG131	Industrial Maint. Mgt.	501	EBB	120
LOG199	Intro. to Logistics	502	91N	80
LOG220	AFLC Material Mgt.	503	EAL	120
LOG221	Logistics Mgers. and Comptr Sim	504	WVH	40
LOG224	Logistics Mgt.	505	EBH	120

ACQUISITION COURSES				
CRSE #	COURSE TITLE	SIM #	PDS	Hr
LOG290	Combat Compatibility Assessment	508	4LE	80
LOG299	Combat Logisitics	509	1L4	80
LOG399	Strategic Logistics Mgt.	510	JBH	120
LOG499	Logistics Exec. Development Course	511	JBJ	80
PPM057	Exec. Contract Admin.	600	JBK	40
PPM151	Industrial Property Admin.	601	AAW	120
PPM153	Production Mgt.	602	EBP	240
PPM300	Adv. Property Admin.	603	QNN	80
PPM301	Contract Mgt. of Engineering Prog.	604	EAN	72
PPM302	Government Contract Law	605	D99	80
PPM304	Adv. Contract Admin.	606	AAS	80
PPM305	Production Mgt II	607	EAJ	80
PPM306	Contractual Aspects of Value Eng.	608	WW8	40
PPM309	Intro. to Sys. Production Mgt.	609	1R7	80
PPM355	Contract Overhead Mgt.	610	Z1M	80
PPM501	Planning for Sys. Production	611	1R8	80
PPM502	Productibility	612	1R9	80
QMT089	Adv. Process Control Methods	700	69A	40
QMT090	Statistical Process Control Method	701	69B	40
QMT170	Principles . of Contract Pricing	702	PBC	112
QMT175	Principles of Cost Analysis	703	8NU	80
QMT180	Cost Improvement Curve Analysis	704	ZYC	40
QMT185	Computer Software Applications	705	OHG	80
QMT335	Rely & Maint. Desgn in Sys. Acq	706	54G	80
QMT345	Quant. Tech. Cost and Price Analys	707	UGH	112
QMT353	Intro. to Life Cycle Costing	708	X8E	80
QMT365	Basic Programming	709	OHH	80

ACQUISITION COURSES				
CRSE #	COURSE TITLE	SIM #	PDS	Hr
QMT540	Advanced Contract Pricing	712	PBE	80
QMT550	Adv. Quant. Method for Cost Analys	713	AJU	120
QMT551	Adv. Cost & Economic Analysis	714	WW4	160
QMT578	Rely & Maint. Resch & Applications	715	O5B	120
DSMCXX	Acquisition Basics Course	322	PXF	160
TPS1	Flight Test Engineer Course	800	R4W	999
TPS2	Experimental Test Pilot Course	801	ABM	999
TPS3	Flight Test Navigator Course	802	T3Z	999
SYS123	Fundamentals of Acq. Mgt.	116	Z3E	40
DSMC-3	Program Managers Course	302	AJS	80
LOG262	Applied Maint. Mgt. Concepts	507	WW6	80
QMT375	FORTRAN Programming	711	OHJ	80
DSMC-2	Exex. Refresher in Acq. Mgt	301	APA	80
LOG260	Provisioning Mgt.	506	EMT	104
QMT372	Reliability	710	ENW	120
DSMC-1	Sys. Acq. Mgt. Course for Gen/SES	300	R3B	36
SAS009	Command Acq. Acessions	208	NZO	80
SAS007	General Managers Logistics Trng.	207	80P	24
SAS006E	Intermediate Sys. Acq. Mgt. Exam	206	SNU	80

Appendix B: Sample Input Data

Personnel Input Data

0100961	CPT	ZQ	9	T	10	TECH	M	4	H2825	HVF	4HQ	1M1	EBF			1891
0010604	MAJ	ZT			13	TECH	M	5	H2865B	EQU	EPD	VB3	URN	YYJ	V8D	1892
0208398	CPT	ZS	1	R	12	OTHR	M	3	H2671	2RY	4HQ	75E	AAA	32X	ZMQ	1893
0172943	CPT	ZK	1	K	13	BUS	M	7	6746	PBC	24X	UGJ	EJ3	E90		1894
0343813	CPT	ZK			6	OTHR	M	1	6746	54H	8NU	ZYC	MWV	IFF	8PP	1895
0002703	MAJ	ZC			13	BUS	M	5	T6534	8EE	UGH	AAS	D99	VGE	22H	1896
0174141	LTC	IC	2	T	19	BUS	M	4	H6516	N83	MV9	AJS	0HZ	AAS	DRH	1897
0145696	LTC	ZT	9	T	15	TECH	M	4	K2875B	3EL	TR9	2RX	T3Z	VZ7	VZ6	1898
0360998	LTC	IL	9	T	21	BUS	M	3	0046	N83	10E	10E	E06	EHC		1899
0219254	CPT	ZQ	9	T	6	TECH	B	1	H2845	MV4	TR9	3EL				1900
0138119	LTC	IP	1	T	18	BUS	M	4	2716	N83	2RZ	0HZ	VGE	9X0	XLE	1901
0142870	LTC	ZZ	1	T	19	BUS	M	6	2716	2RZ	VG6	3EL	2RY	4HQ	WWN	1902
0215931	CPT	ZQ	9	T	11	TECH	B	3	H2825	QTT	9XZ	81A	9X0	MMV	EAY	1903
0360291	CPT	ZL			8	BUS	B	2	4024	2RX	2RX	PZW	7AA	DRH	6QP	1904
0143729	MAJ	ZA	2	T	16	BUS	M	3	2716	2RZ	2RY	8C9	X8E	1UP	U76	1905
0021886	CPT	ZT	1	T	10	BUS	M	6	H2835	HVF	3CN	ZD4	9X0	4HQ	4HQ	1906
0010340	1LT	ZT	9	T	4	TECH	B	4	2855	1FF	WYK					1907
0160810	CPT	ZQ			8	TECH	B	3	2825	4HQ	8B3	1QG	6PZ	15F		1908
0091828	COL	ZS			16	OTHR	P	4	H9356	NFE	NFE	QL9	NFE	QL9	WSU	1909
0360605	COL	IL			23	OTHR	M	3	0046	2RX	7PQ	UWB	0EK	VBR	XBA	1910
0019507	LTC	IP	5	T	22	TECH	M	5	2806	NPI	N83	QTT	SY7	DRH	RW5	1911

0184438	CPT	ZT	1	P	9	TECH	B	2	G2815A	79D	9X0	1NA	3ZZ			1912
0016609	LTC	IY			22	TECH	P	2	5516	YBR	AMV	FEA				1913
0218572	MAJ	ZT			16	OTHR	M	3	H2865B	2RX	09N	09M	VBX	YH4	V8D	1914
0173240	MAJ	ZQ	1	Q	16	OTHR	B	2	U2806W	2RY	ZRP	S5S	2RX	B8Y	S5R	1915
0395375	MAJ	ZL			13	BUS	M	4	4016	2RX	WYF	44M	PZW	31D	UDG	1916
0136668	MAJ	ZQ	2	R	16	TECH	M	1	H2806	2RZ	2RY	3EL	MV4	VC5	ERR	1917
0082469	MAJ	ZP	9	T	14	TECH	M	3	2711	2RY	TR9	AAA	16D			1918
0131777	MAJ	ZP	1	K	15	BUS	M	6	0071	PBH	X8E	PBE	UGH	ZYC	PBC	1919
0200750	MAJ	ZT	9	T	14	BUS	M	1	U2711W	N83	4HQ	ABM	AHP	V8D	ERR	1920
0171503	CPT	ZP	1	T	11	BUS	M	2	2711	MMW	TR9	2RX	TZ2	TB3	ERR	1921
0165936	LTC	ZP	2	T	17	OTHR	M	3	2716	N83	1Q1	3HF	NC4			1922
0391036	LTC	ZT			19	TECH	P	4	G2806	2RZ	2RX	TR9	PZW	V8D	R4W	1923
0161012	CPT	ZG	1	G	9	OTHR	B	2	H6524	D99	2RX	EAJ	1FF	XLE		1924
0138145	CPT	ZQ			9	TECH	M	4	H2855	WYF	9X0	4HQ				1925
0071167	CPT	ZQ	1	Q	5	TECH	B	1	H2825	92G	81A	QTT	8B3	9X0	4HQ	1926
0146881	LTC	ZP			17	BUS	M	4	H3124G	3CN	2ST	2RX	7T1	7TL	54J	1927
0120722	CPT	ZT			7	TECH	M	1	H2825	2RX						1928
0172287	1LT	ZQ			3	TECH	B	3	H2665	IFF						1929
0248432	CPT	ZC	9	T	6	BUS	M	1	H6534	Z1M	D99	QTT	PBC	MMW	4HQ	1930
0209544	1LT	ZP			3	TECH	B	3	H2685							1931
0216902	2LT	ZP			1	TECH	B	1	2051B							1932
0245603	LTC	IL			19	OTHR	M	2	4096	JBH	1GG	X9D	YG0	PZW	ZRP	1933
0160788	CPT	ZQ	1	Q	10	TECH	M	5	H2855	2RY	2RX	WVD	D99	WVD	3EL	1934
0157114	1LT	ZT	9	T	4	TECH	B	4	H2845							1935
0081057	CPT	ZP	9	T	4	TECH	B	4	2724	9X0	9XZ	4HQ				1936

0144721	CPT	ZQ	9	T	4	TECH	M	4	2835	TR9	4HQ					1937
0222691	CPT	ZS			7	TECH	M	1	2825	1MS	56M	3GZ				1938
0262444	CPT	ZL			5	TECH	B	4	4024	WW6	DRH	YCN	UDG			1939
0313905	1LT	ZL			3	TECH	B	3	4024	UDG						1940
0000602	MAJ	OT			17	BUS	M	3	4916	2RX	J5J	8SG	0CZ	44B	5F5	1941
0441334	MAJ	OT			20	BUS	M	3	2225R	2RX	EHT	VDD	SST	V8D	AF2	1942
0146880	LTC	ZP	9	T	19	BUS	M	4	2716	4HQ	5CP	VDJ	AF6	EQQ	V8D	1943
0183977	CPT	ZT			11	TECH	B	4	2875A	ABM	4HQ	VDJ	ERR	TB3	VC0	1944
0006581	CPT	ZQ	9	T	9	TECH	M	4	H2825	79D	8B3	TR9	EAY			1945
0150609	MAJ	ZT	1	T	13	BUS	M	3	G2895	9X0						1946
0245416	MAJ	ZL			13	OTHR	M	2	4011	K9F	DRH	2R6	AAA	1FF	V2R	1947
0450779	CPT	ZA			8	BUS	M	3	4945C	VHD	U1G	TFL	AH6	TB3	ERR	1948
0065067	CPT	ZQ	9	T	7	TECH	B	3	H2855	X8E	TR9	05A	9X0	4HQ		1949
0171581	CPT	ZP	1	T	8	OTHR	M	3	2724	24X	2RY	2RX				1950
0427816	2LT	ZL			2	OTHR	B	1	4021	Q8V	TB3	ERR				1951
0134706	1LT	ZT			3	TECH	B	3	2855	1FF						1952
0100226	LTC	IP			22	TECH	M	3	V2716W	N83	ZZZ	WS0	U7A	AF2	V8D	1953
0028634	MAJ	IY			13	TECH	M	2	H2611							1954
0145230	CPT	ZS	1	T	11	TECH	M	5	H2855	2VD	1QG	2RY	UH7	2RX		1955
0058633	CPT	ZR			11	TECH	M	2	H4911	2RX	TB3	ERR				1956
0000474	CPT	OT			10	OTHR	B	1	4011	PZW	QQM	31C	DHR	YJ7	UDG	1957
0199955	1LT	ZT			3	TECH	B	3	2724	92G	8B3					1958
0228886	CPT	ZX			5	TECH	B	5	4925	3FV	2JJ					1959
0000526	LTC	OT			21	TECH	M	4	1435C	54H	4X1	VDK	ZK6	XLJ	AF2	1960
0005308	MAJ	ZC	2	C	18	BUS	M	1	H6516	JBK	Z1M	VGE	SRQ	AAS	PBC	1961

0142866	MAJ	ZS	1	T	13	TECH	M	8	H2645	9XO	9X0	4HQ				1962
0173155	CPT	ZG			10	BUS	M	2	H6524	4HQ	EAJ	XLH	Z2G	RW5	V2R	1963
0137909	CPT	ZT	9	T	8	TECH	M	3	2825	1MS	2RY					1964
0187011	1LT	ZP			2	TECH	B	2	5521E	9A8	2RX					1965
0021923	COL	IZ			29	BUS	M	3	0076	N85	NPI	APC	R92	WVG	V0D	1966
0360759	CPT	ZL			7	BUS	M	2	6624	6T4	7V0					1967
0408388	CPT	ZL			11	TECH	B	6	C4024	79E	Y9U	UDG				1968
0000919	CPT	OT			7	TECH	M	2	2685	2RX	54H	WYF	EBF			1969
0029148	CPT	ZC			11	BUS	M	4	H6534	Q3S	U7X					1970
0049412	CPT	ZQ			10	TECH	M	3	H2845	2RX						1971

Position Input Data

0322797	MAJ	IA	H4016
0419097	MAJ	IA	H4916
0004805	MAJ	IA	V2716W
0241662	MAJ	IA	H4916
0249181	MAJ	IA	2716
0249194	MAJ	IA	2716
0257946	MAJ	IA	2716
0276131	MAJ	IA	H4916
0013588	LTC	IA	H0046
0115159	LTC	IA	A1475N
0269410	LTC	IA	H2716
0297646	LTC	IA	H2716
0331317	LTC	IA	H0046
0335061	LTC	IA	H0046
0344671	LTC	IA	H4996
0378016	LTC	IA	U2716W
0385824	LTC	IA	H6616
0395845	LTC	IA	H6616
0396016	LTC	IA	H2716
0404253	LTC	IA	H6616
0405757	LTC	IA	A2875A
0413366	LTC	IA	H4996

0413870	LTC	IA	H0046
0426126	LTC	IA	H0046
0110531	LTC	IA	H2806
0174163	LTC	IA	2716
0182616	LTC	IA	8116
0206199	LTC	IA	2716
0001006	LTC	IA	T2716
0001007	LTC	IA	T2716
0001011	LTC	IA	T6516
0007612	LTC	IA	T2716
0142850	LTC	IA	H4916
0228959	LTC	IA	H4916
0242312	LTC	IA	2716
0282757	LTC	IA	H4996
0000158	LTC	IA	H2806
0000237	LTC	IA	H2716
0000574	LTC	IA	H2806
0001214	LTC	IA	H2806
0031458	GEN	IA	H0002
0041500	GEN	IA	H0002
0085044	GEN	IA	H0002
0118137	GEN	IA	H0002
0118139	GEN	IA	H0002
0134591	GEN	IA	H0002
0137666	GEN	IA	H0002

0141514	GEN	IA	H0002
0241904	GEN	IA	H0002
0426124	GEN	IA	H0002
0426125	GEN	IA	H0002
0000038	GEN	IA	H0002
0000938	GEN	IA	H0002
0032076	GEN	IA	H0002
0043948	GEN	IA	H0002
0067590	GEN	IA	H0002
0074973	GEN	IA	H0002
0100616	GEN	IA	H0002
0100617	GEN	IA	H0002
0100637	GEN	IA	H0002
0101837	GEN	IA	H0002
0101838	GEN	IA	H0002
0101929	GEN	IA	H0002
0102389	GEN	IA	H0002
0104911	GEN	IA	H0002
0116014	GEN	IA	H0002
0122733	GEN	IA	H0002
0126037	GEN	IA	H0002
0131194	GEN	IA	H0002
0142375	GEN	IA	H0002
0144456	GEN	IA	H0002
0156787	GEN	IA	H0002

0162236	GEN	IA	H0002
0166249	GEN	IA	H0002
0194412	GEN	IA	H0002
0261965	GEN	IA	G0002
0261966	GEN	IA	0002
0440889	GEN	IA	0002
0276651	CPT	IA	H3124G
0031003	CPT	IA	H2724
0021990	COL	IA	H4096
0061026	COL	IA	U0046W
0068900	COL	IA	H0046
0134620	COL	IA	U0046W
0135734	COL	IA	U0046W
0135759	COL	IA	U0046W
0136453	COL	IA	H0046
0136599	COL	IA	U0046W
0136646	COL	IA	U0046W
0137675	COL	IA	H0046
0138433	COL	IA	U0046W
0138486	COL	IA	H0046
0139178	COL	IA	H0046
0226259	COL	IA	H0046
0230630	COL	IA	H0046
0247840	COL	IA	H0046
0262093	COL	IA	U0046W

0276707	COL IA H0046
0304087	COL IA H0046
0309223	COL IA H0046
0310552	COL IA H0046
0335059	COL IA H0046
0336895	COL IA H0046
0344595	COL IA H0046
0344596	COL IA H0046
0344597	COL IA H0046
0344695	COL IA H0046
0347563	COL IA U2716W

Appendix C

PROGRAM DATA.FOR VARIABLE DEFINITION

<u>INPUT VARIABLES (Character Data)</u>

POSNUM
  Defn: position number in the manpower file.
  Field: 1-7

PERSON
  Defn: position number in the personnel file.
  Field: 1-7

POSRNK
  Defn: authorized grade for the position in the manpower file.
  Field: 10-12
  Values: LT -- GEN

PERRNK
  Defn: actual grade of an individual in the personnel file.
  Field: 10-12
  Values: 2LT -- GEN

POSTYP
  Defn: acquisition position type in manpower file.
  Field: 14
  Values: I, O, or Z

PERTYP
  Defn: acquisition position type in personnel file.
  Field: 14
  Values: I, O, or Z

MOCAT
  Defn: acquisition career field category for position
  Field: 15
  Values: A thru Z excluding F, I, J, M, N, O, P, Q, and V

DPCAT
Defn: acquisition career field category for person
Field: 15
Values: Same as MOCAT

PREFIX
Defn: prefix for AFSC
Field: 17
Values: not used in simulation

SUFFIX
Defn: suffix for AFSC
Field: 22
Values: not used in simulation

AFSC
Defn: AFSC for the position without prefix and suffix
Field: 18-21
Values: All acquisition AFSC

CLEVL
Defn: current certification level in personnel file.
Field: 17
Values: 0 - 9

FAREA
Defn: functional area of certification in personnel file.
Field: 19
Values: Same as MOCAT

TIME
Defn: years of service
Field: 21-22
Values: 0-40

BACHLR
Defn: indicates the type of bachelor's degree.
Field: 24-27
Values: BUS, TECH, and OTHR

DIPLMA
Defn: indicates the highest level of education
Field: 29
Values: B, M, and P

TOB
    Defn: time on station
    Field: 31-32
    Values: 0-20

DAFSC
    Defn: Duty AFSC
    Field: 35-38
    Values: all acquisition AFSCs

TRNG1 - TRNG24 (Attributes 17-40)
    Defn: special training courses taken
    Field: 41-EOL

<u>OUTPUT VARIABLES (Integer data)</u>

JOBNUM (Attribute 1)
    Defn: position number for each acquisition position
    Field: 1-8

POSGRD (Attribute 2)
    Defn: required grade for each acquisition position
    Field: 10-12
    Values: 10 = General
        9 = LT General
        8 = Major General
        7 = Brigadere General
        6 = COL
        5 = LTC
        4 = MAJ
        3 = CPT
        2 = 1LT
        1 = 2LT

RLEVL (Attribute 3)
    Defn: Required level of certification for the position
    Field: 14-16
    Values: 1-3

ACQPOS (Attribute 4)
    Defn: acquisition portion type
    Field: 18-20
    Values: 1 = I = critical
        2 = O = developmental
        3 = Z = related (other)

POSCAT (Attribute 5)

Defn: acquisition position career field category

Field: 22-24

Values:
1 = A
2 = B
3 = C
4 = D
5 = E
6 = F
7 = G
8 = H
9 = J
10 = K
11 = L
12 = M
13 = P
14 = Q
15 = R
16 = S
17 = T
18 = U
19 = W
20 = X
21 = Y
22 = Z
99 = BAD DATA

STATUS (Attribute 6)

Defn: status of the position

Field: 26-28

Values:
0 = vacant
1 = filled

PERNUM (Attribute 7)

Defn: position number of the person filling the job

Field: 30-37

PERGRD (Attribute 8)

Defn: grade of the person filling the position

Field: 39-41

Values: Same as POSGRD above

ACQPER (Attribute 9)

Defn: acquisition position type

Field: 43-45

Values: Same as ACQPOS above

PERCAT (Attribute 10)
Defn: category of career field acquisition position
Field: 47-49
Values: Same as POSCAT above

SLEVL (Attribute 11)
Defn: starting certification level
Field: 51-53
Values: 0 = no certification listed in records
1 = certification 1
2 = certification 2
3 = certification 3
4 = old level 2 on acquisition manager's list
5 = old level 2 on senior acquisition manager's list
6 = old level 3 on acquisition manager's list
7 = old level 3 on senior acquisition manager's list
9 = unknown
99 = erroneous data

FCTCAT (Attribute 12)
Defn: current functional acquisition career field category
Field: 55-57
Values: Same as POSCAT above

YOS (Attribute 13)
Defn: total active duty years of service
Field: 59-61
Values: 0-40

DEGREE (Attribute 14)
Defn: Type of bachelors degree received
Field: 59-61
Values: 0 = business
1 = technical
2 = other

EDLEVL (Attribute 15)
Defn: level of education received.
Field: 63-65
Values: 0 = bachelors
1 = masters
2 = doctorate

CORP (Attribute 16)
Defn: acquisition corps designator
Field: 67-69
Values: 0 = not a member
1 = member

TRNG1-TRNG24 (Attribute 17 - Attribute 40)
Defn: special training courses taken
Field: 71-190 (each 4 characters with 1 space between)
Values: see appendix A on acquisition training courses

NEWCRS1-NEWCRS10 (Attribute 41-50)
Defn: special training courses acquired during simulation
Field: 191-250
Values: see appendix A on acquisition training courses

NEED1-NEED10 (Attribute 51-60)
Defn: course deficiencies for level three certification
Field: 252-311
Values: AFIT courses offered from list in Appendix A

ELEVL (Attribute 61)
Defn: Evaluated level of certification
Field: 313
Values: 0 = no certification
1 = level 1
2 = level 2
3 = level 3

PRIORT (Attribute 62)
Defn: Designator between the backlog and on-time training
Field: 315
Values: 0 = on-time population
1 = backlog population

NEW (Attribute 63)
Defn: Distinguish between the starting population and accessions during aging process
Field: 317
Values: 0 = entered through accessions
1 = member of starting population

FREE (Attribute 64, 66, 67)

ENTRY (Attribute 65)
Defn: School entry date
Field: 319

Values: extracted from simulation clock as TNOW

YEAR (Attribute 68)
Defn: Current calendar Year
Field: 321
Values: 1992-2020

PASOVR (Attribute 69)
Defn: Promotion passover counter
Field: 323
Values: 0 = no passovers
1 = first passover
2 = second passover
3 = scheduled for attrition from system

TDY (Attribute 70)
Defn: Designator to prevent a person from being eligible for selection to a course when the person is currently in a course
Field: 325
Values: 0 = eligible for course selection
1 = not eligible for course selection

Appendix D

Attribute Definitions

ATTRIBUTE	DEFINITION
1	POSITION NUMBER
2	REQUIRED GRADE
3	REQUIRED CERTIFICATION LEVEL
4	ACQUISITION POSITION TYPE
5	POSITION CAREER STALL
6	POSITION STATUS
7	PERSON NUMBER
8	PERSON GRADE
9	PERSON ACQUISITION TYPE
10	PERSON CAREER STALL
11	STARTING CERTIFICATION LEVEL
12	FUNCTION CATEGORY
13	YEARS OF SERVICE
14	TYPE OF DEGREE OBTAINED
15	EDUCATIONAL LEVEL
16	ACQUISITION CORPS DESIGNATOR
17	COURSE 1 IN EDUCATIONAL HISTORY OF 24
18	COURSE 2 IN EDUCATIONAL HISTORY OF 24
19	CRSE3
20	CRSE4
21	CRSE5
22	CRSE6
23	CRSE7
24	CRSE8
25	CRSE9
26	CRSE10
27	CRSE11
28	CRSE12
29	CRSE13
30	CRSE14
31	CRSE15
32	CRSE16
33	CRSE17
34	CRSE18
35	CRSE19
36	CRSE20
37	CRSE21
38	CRSE22
39	CRSE23
40	CRSE24
41	NEW COURSE 1 OF 10
42	NEW COURSE 2 OF 10
43	NEWCRS
44	NEWCRS

45	NEWCRS
46	NEWCRS
47	NEWCRS
48	NEWCRS
49	NEWCRS
50	NEWCRS
51	NEEDED COURSE 1 OF 10
52	NEEDED COURSE 2 OF 10
53	NEED03
54	NEED04
55	NEED05
56	NEED06
57	NEED07
58	NEED08
59	NEED09
60	NEED10
61	EVALUATED CERTIFICATION LEVEL
62	PRIORITY/ON-TIME DESIGNATOR
63	NEW ACCESSION DESIGNATOR
64	FREE
65	SCHOOL ENTRY DATE
66	FREE
67	FREE
68	CURRENT YEAR
69	PROMOTION PASS OVER COUNTER
70	TDY DESIGNATOR

Appendix E

Historical Acquisition Courses

3EM	105
9AG	106
LOB	107
UGJ	108
VG6	109
TR9	110
4ZO	111
4Z0	111
22H	112
24X	113
EAY	114
EBF	115
8B3	201
81A	202
9X2	203
92G	204
9XO	205
9X0	205
SNU	206
8OP	207
80P	207
NZO	208
NZ0	208
35B	304
QTT	305
ZD4	306
1GZ	306
OHZ	307
0HZ	307
1Q1	308
3CN	309
28N	310
4X1	312
9A9	313
HVF	315
AN6	316
JEJ	318
L38	319
N85	320
LRB	321
7PO	400
7P0	400
OPG	500

0PG	500
EBB	501
91N	502
EAL	503
WVH	504
EBH	505
EMT	506
WW6	507
4LE	508
1L4	509
JBH	510
JBJ	511
JBK	600
AAW	601
EBP	602
QNN	603
EAN	604
D99	605
AAS	606
EAJ	607
WW8	608
1R7	609
ZIM	610
1R8	611
1R9	612
69A	700
69B	701
PBC	702
8NU	703
ZYC	704
OHG	705
0HG	705
54G	706
UGH	707
X8E	708
OHH	709
0HH	709
ENW	710
OHJ	711
0HJ	711
PBE	712
AJU	713
WW4	714
O5B	715
05B	715
RRR	777
SSS	888
TTT	999

Appendix F

Simulation Model Subroutines

```
      PROGRAM MAIN
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'
      DIMENSION NSET(900000)

      COMMON QSET(900000)
      EQUIVALENCE (NSET(1),QSET(1))
      NNSET=900000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      OPEN(UNIT=NCRDR,FILE='fort.5',STATUS='UNKNOWN')
      OPEN(UNIT=NPRNT,FILE='fort.6',STATUS='UNKNOWN')
      NPLOT=2
      SPECL = 1
      ICOUNT = 1
      CALL SLAM
      STOP
      END


      SUBROUTINE EVENT(I)
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
     1,15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
     1,28, 29, 30, 31, 32, 33, 34, 35, 36, 37),I


1     CALL PREP
      RETURN

2     CALL SY100A
      RETURN

3     CALL SY100B
      RETURN

4     CALL SY200A
      RETURN

5     CALL SY200B
```

```
      RETURN
 6    CALL SY200C
      RETURN
 7    CALL SY200D
      RETURN
 8    CALL SY200E
      RETURN
 9    CALL SY225A
      RETURN
10    CALL SY225B
      RETURN
11    CALL SY225C
      RETURN
12    CALL SY225D
      RETURN
13    CALL SY225E
      RETURN
14    CALL SY225F
      RETURN
15    CALL SY400A
      RETURN
16    CALL SY400B
      RETURN
17    CALL SY400C
      RETURN
18    CALL SY400D
      RETURN
19    CALL SY400E
      RETURN
20    CALL SY400F
      RETURN
21    CALL SY400G
      RETURN
```

```
22   CALL SY400H
     RETURN

23   CALL SY400I
     RETURN

24   CALL SPLCRS
     RETURN

25   CALL PRMOTE
     RETURN

26   CALL ATTRIT
     RETURN

27   CALL ACCESS
     RETURN

28   CALL GRD100
     RETURN

29   CALL UPDATE
     RETURN

30   CALL SERB
     RETURN

31   CALL DEMAND
     RETURN

32   CALL GRD200
     RETURN

33   CALL GRD225
     RETURN

34   CALL GRD400
     RETURN

35   CALL GRD999
     RETURN

36   CALL BATCH
     RETURN

37   CALL STATS
     RETURN
     END
```

```
**************************************************************
***
*  The purpose of subroutine INTLC.FOR is to call the initial
*
*  subroutines which setup population parameters for the
*
*  simulation to begin.  It also starts the initial training
*
*  and statistics collection cycles for subsequent iterations
*
*  by the simulation program.
*
**************************************************************
***

      SUBROUTINE INTLC
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      CALL SCHDL(1, 0.0, ATRIB)

      CALL SCHDL(2, 97.0, ATRIB)

      CALL SCHDL(3, 281.0, ATRIB)

      CALL SCHDL(4, 295.0, ATRIB)

      CALL SCHDL(5, 27.0, ATRIB)

      CALL SCHDL(6, 62.0, ATRIB)

      CALL SCHDL(7, 160.0, ATRIB)

      CALL SCHDL(8, 195.0, ATRIB)

      CALL SCHDL(9, 295.0, ATRIB)

      CALL SCHDL(10, 158.0, ATRIB)

      CALL SCHDL(11, 97.0, ATRIB)

      CALL SCHDL(12, 258.0, ATRIB)

      CALL SCHDL(13, 337.0, ATRIB)

      CALL SCHDL(14, 132.0, ATRIB)

      CALL SCHDL(15, 295.0, ATRIB)
```

```
      CALL SCHDL(16, 344.0, ATRIB)

      CALL SCHDL(17, 6.0, ATRIB)

      CALL SCHDL(18, 55.0, ATRIB)

      CALL SCHDL(19, 90.0, ATRIB)

      CALL SCHDL(20, 118.0, ATRIB)

      CALL SCHDL(21, 160.0, ATRIB)

      CALL SCHDL(22, 230.0, ATRIB)

      CALL SCHDL(23, 258.0, ATRIB)

      CALL SCHDL(24, 13.0, ATRIB)

      CALL SCHDL(36, 360.0, ATRIB)

      CALL SCHDL(25, 361.0, ATRIB)

      CALL SCHDL(30, 362.0, ATRIB)

      CALL SCHDL(26, 363.0, ATRIB)

      CALL SCHDL(29, 364.0, ATRIB)

      CALL SCHDL(27, 365.0, ATRIB)
      RETURN
      END

************************************************************
****
*  Subroutine PREP reads in the data and set up the initial
 *
*  simulation condtions.
 *
************************************************************
****

      SUBROUTINE PREP
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

************************************************************
****
*  The following procedures initialize and fill the auxillary
*
```

```
*  attribute array, AUX.  This array contains the critical
 *
*  attributes of each person in the acquisition workforce.
 *
*************************************************************
****

      OPEN(UNIT=10, FILE='NEW.DAT', STATUS='OLD')

      DO 10 I = 1,ROWS
         DO 20 J = 1, COLS
            AUX(I,J) = 0
20       CONTINUE
10    CONTINUE

      DO 30 I = 1, ROWS
         READ(10,100,END=41) (AUX(I,J), J=1,40)
30    CONTINUE
41    CLOSE(10)

      DO 70 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            AUX(I,70) = 0
            AUX(I,63) = 1
            AUX(I,68) = 1992
            DO 60 J = 1, COLS
               ATRIB(J) = REAL(AUX(I,J))
60          CONTINUE
            CALL FILEM(15,ATRIB)
         END IF
70    CONTINUE
      CLOSE(29)

100   FORMAT(I8, 5(1X,I3), 1X, I8, 9(1X,I3), 24(1X,I4))
      PRINT *, 'DONE WITH PREP'
      CALL SCHDL(31, 0.0, ATRIB)
      RETURN
      END


*************************************************************
****
*  Subroutine DEMAND uses the attribute arrays AUX and ATRIB
to*
*  to access the demand of critical courses taught by AFIT in
*
*  in support of APDP certification policy.
 *
*************************************************************
****
```

```
      SUBROUTINE DEMAND
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER BKLOG, NUM, TOTAL, POP, TEMP(70), YEAR, SPCRS,
     $DMD100, HI100, CRS100, LO100, DMD200, HI200, CRS200,
LO200,
     $DMD400, HI400, CRS400, LO400, DMD225, HI225, CRS225,
LO225,
     $PMC03, DSMC03, SPECLY, BAKLOG

      PRINT *,'STARTING DEMAND'
      BAKLOG = 0
      HI100 = 0
      LO100 = 0
      HI200 = 0
      LO200 = 0
      HI400 = 0
      LO400 = 0
      HI225 = 0
      LO225 = 0
      DSMC03 = 0
      SPECLY = 0

      DO 150 I = 1,70
         TEMP(I) = 0
150   CONTINUE

      DO 50 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            IF ((AUX(I,8) .EQ. 1) .AND. (AUX(I,13) .GT. 2))
THEN
               AUX(I,69) = AUX(I,13) - 2
            ELSE IF ((AUX(I,8) .EQ. 2) .AND.
     $      (AUX(I,13) .GT. 4)) THEN
                AUX(I,69) = AUX(I,13) - 4
            ELSE IF ((AUX(I,8) .EQ. 3) .AND.
     $      (AUX(I,13) .GT. 12)) THEN
               AUX(I,69) = AUX(I,13) - 12
            ELSE IF ((AUX(I,8) .EQ. 4) .AND.
     $      (AUX(I,13) .GT. 16)) THEN
               AUX(I,69) = AUX(I,13) - 16
            ELSE IF ((AUX(I,8) .EQ. 5) .AND.
     $      (AUX(I,13) .GT. 21)) THEN
               AUX(I,69) = AUX(I,13) - 21
            ELSE IF ((AUX(I,8) .EQ. 6) .AND.
     $      (AUX(I,13) .GT. 24)) THEN
               AUX(I,69) = AUX(I,13) - 24
            ELSE IF ((AUX(I,8) .EQ. 7) .AND.
     $      (AUX(I,13) .GT. 26)) THEN
```

```
                 AUX(I,69) = AUX(I,13) - 26
              END IF
           END IF
 50     CONTINUE

        OPEN(UNIT=19,FILE='FILE.CHK',STATUS='NEW')
        PRINT *, 'STARTING ACQLOG SECTION'
        DO 10 I = 1, ROWS
           IF (AUX(I,6) .EQ. 1) THEN
              DO 25 J = 1, COLS
                 TEMP(J) = AUX(I,J)
 25           CONTINUE
              CALL ACQLOG(TEMP, BKLOG, DMD100, CRS100, DMD200,
     $        CRS200, DMD400, CRS400, DMD225, CRS225, PMC03,
     $        SPCRS)
              POP = NNQ(15)
              DO 19 P = 1, POP
                 CALL RMOVE(1, 15, ATRIB)
                 IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                    DO 140 J = 1, COLS
                       AUX(I,J) = TEMP(J)
                       ATRIB(J) = REAL(TEMP(J))
 140                CONTINUE
                    CALL FILEM(15,ATRIB)
                    GO TO 21
                 ELSE
                    CALL FILEM(15, ATRIB)
                 END IF
 19           CONTINUE
 21           BAKLOG = BAKLOG + BKLOG
              HI100 = HI100 + DMD100
              LO100 = LO100 + CRS100
              HI200 = HI200 + DMD200
              LO200 = LO200 + CRS200
              HI400 = HI400 + DMD400
              LO400 = LO400 + CRS400
              HI225 = HI225 + DMD225
              LO225 = LO225 + CRS225
              DSMC03 = DSMC03 + PMC03
              SPECLY = SPECLY + SPCRS
              YEAR = AUX(I,68)
           END IF
 10     CONTINUE
        XX(2) = SUMQ(62,15)
        XX(3) = SUMQ(51,15)
        XX(4) = SUMQ(52,15)
        XX(5) = SUMQ(55,15)
        XX(6) = SUMQ(54,15)
        XX(7) = SUMQ(53,15) + SUMQ(56,15) + SUMQ(57,15)
        NUM = NNQ(15)
```

```
      WRITE(19, *) 'CHECKING THE SUMQ FUNCTION'
      WRITE(19, *) XX(2), XX(3), XX(4), XX(5), XX(6), XX(7)
      WRITE(19, *) NNQ(15),' IN SYSTEM BEFORE SPLIT'
      DO 111 I = 1, NUM
         CALL RMOVE(1, 15, ATRIB)
         IF (ATRIB(62) .EQ. 1) THEN
            IF (ATRIB(51) .EQ. 1) THEN
               CALL FILEM(11,ATRIB)
            ELSE IF (ATRIB(52) .EQ. 1) THEN
               CALL FILEM(21,ATRIB)
            ELSE IF (ATRIB(53) .EQ. 1) THEN
               CALL FILEM(31,ATRIB)
            ELSE IF (ATRIB(54) .EQ. 1) THEN
               CALL FILEM(41,ATRIB)
            ELSE IF (ATRIB(55) .EQ. 1) THEN
               CALL FILEM(25,ATRIB)
            ELSE IF (ATRIB(56) .EQ. 1) THEN
               CALL FILEM(31,ATRIB)
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               CALL FILEM(31,ATRIB)
            ELSE
               CALL FILEM(15,ATRIB)
            END IF
         ELSE IF (ATRIB(62) .EQ. 0) THEN
            IF (ATRIB(51) .EQ. 1) THEN
               CALL FILEM(1, ATRIB)
            ELSE IF (ATRIB(52) .EQ. 1) THEN
               CALL FILEM(2, ATRIB)
            ELSE IF (ATRIB(53) .EQ. 1) THEN
               CALL FILEM(3, ATRIB)
            ELSE IF (ATRIB(54) .EQ. 1) THEN
               CALL FILEM(4, ATRIB)
            ELSE IF (ATRIB(55) .EQ. 1) THEN
               CALL FILEM(22, ATRIB)
            ELSE IF (ATRIB(56) .EQ. 1) THEN
               CALL FILEM(3, ATRIB)
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               CALL FILEM(3, ATRIB)
            ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
111   CONTINUE
      WRITE(19, *) NNQ(15), NNQ(1), NNQ(2), NNQ(11), NNQ(21),
     $NNQ(4), NNQ(41), NNQ(3), NNQ(31), NNQ(25), NNQ(22)

      OPEN(UNIT=50, FILE='DEMAND.OUT', STATUS='NEW')
      CALL CRSDMD(BAKLOG, HI100, LO100, HI200, LO200, HI400,
     $LO400, HI225, LO225, DSMC03, YEAR, SPECLY)
```

```
      PRINT *, 'DONE WITH DEMAND'
      CALL SCHDL(37, 0.0, ATRIB)

      RETURN
      END


*************************************************************
****
*   Given that a person is in an acquisition logistics
position,*
*  subroutine ACQLOG compares the courses taken by the person
*
*  passed through the ATRIB array to the courses required for
*
*  that career field.  The resultant is a certification level
*
*  based solely upon the training accomplishments of the
 *
*  individual.  If this value is less than the starting
 *
*  certification level read in from the original data base, it
*
*  is discarded.  Again we have assumed the information in the
*
*  original database is correct.  The new certification is
 *
*  added to the ATRIB array as one  of the attributes of the
 *
*  person and used later during subsequent record evaluations.
*
*************************************************************
****


      SUBROUTINE ACQLOG(TEMP, BKLOG, DMD100, CRS100, DMD200,
     $CRS200, DMD400, CRS400, DMD225, CRS225, PMC03, SPCRS)
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER TEMP(70), ELEVL, DSMC26, SYS100, SYS200, TST100,
     $SYS400, DSMC01, DSMC02, DSMC3A, DSMC20, DSMC30, SAS001,
     $SAS006, SYS225, NUM, DSMABC, DSMC3B, TPS100, TPS200,
     $TPS300, SYS123, NEED(10), REQMT(6,34), LEVL(3),
     $BKLOG, DMD100, CRS100, DMD200, CRS200, DMD400,
     $CRS400, DMD225, CRS225, SPCRS, NDX, CHK(1:10),
     $TEST(6), PMC03, COUNT
```

```
      CHARACTER*3 PDS

      REAL CUTOFF

*************************************************************
****
*  The following list of courses represent the basic mandatory
*
*   course requirements and their equivalent courses.
 *
*************************************************************
****


      COUNT = 0
      CRS100 = 0
      CRS200 = 0
      CRS400 = 0
      CRS225 = 0
      COUNT = 0
      SPCRS = 0
      ELEVL = 0
      ALC001 = 400
      DSMC26 = 314
      SYS100 = 100
      SYS123 = 116
      TST100 = 101
      SYS200 = 102
      SYS400 = 103
      DSMC3A = 302
      DSMC3B = 303
      SAS001 = 200
      SAS006 = 205
      SYS225 = 104
      DSMC01 = 300
      DSMC02 = 301
      DSMC20 = 311
      DSMC30 = 317
      DSMABC = 322
      TPS100 = 800
      TPS200 = 801
      TPS300 = 802

      DO 125 R = 1, 3
         LEVL(R) = 0
125   CONTINUE

      DO 115 I = 1,6
         DO 105 R = 1, 34
            REQMT(I,R) = 0
```

```
105        CONTINUE
115     CONTINUE

        DO 85 R = 1, 10
           CHK(R) = 0
85      CONTINUE

        DO 25 R = 1, 6
           TEST(R) = 0
25      CONTINUE

************************************************************
****
*  The following DO-Loop reads each course occurance in a
 *
*  person's record and compares it to the APDP requirements.
 *
************************************************************
****


        DO 10 J = 1,34

************************************************************
****
*  The following IF-statement checks for the completion of
 *
*  SYS100 or its equivalents.  A boolean variable is set to
 *
*  true if the requirement is met and false if it is not.
 *
************************************************************
****
           NDX = J + 16
           IF ((TEMP(NDX) .EQ. DSMABC) .OR.
     $              (TEMP(NDX) .EQ. SYS100) .OR.
     $              (TEMP(NDX) .EQ. TST100) .OR.
     $              (TEMP(NDX) .EQ. SYS123) .OR.
     $              (TEMP(NDX) .EQ. SYS225) .OR.
     $              (TEMP(NDX) .EQ. DSMC26) .OR.
     $              (TEMP(NDX) .EQ. SAS001)) THEN
              REQMT(1,J) = 1
           ELSE
              REQMT(1,J) = 0
           END IF

           IF ((TEMP(NDX) .EQ. TPS100) .OR.
     $              (TEMP(NDX) .EQ. TPS200) .OR.
```

```
     $           (TEMP(NDX) .EQ. TPS300)) THEN
           REQMT(2,J) = 1
        ELSE
           REQMT(2,J) = 0
        END IF

*************************************************************
****
* The following statements check for the completion of SYS200
*
*  or its equivalents.
 *
*************************************************************
****

        IF ((TEMP(NDX) .EQ. SYS200) .OR.
     $           (TEMP(NDX) .EQ. SAS006)) THEN
           REQMT(3,J) = 1
        ELSE
           REQMT(3,J) = 0
        END IF

*************************************************************
****
* The following statements check for the completion of SYS400
*
*  or its equivalents.
 *
*************************************************************
****

        IF ((TEMP(NDX) .EQ. SYS400) .OR.
     $           (TEMP(NDX) .EQ. DSMC01) .OR.
     $           (TEMP(NDX) .EQ. DSMC02) .OR.
     $           (TEMP(NDX) .EQ. DSMC3A) .OR.
     $           (TEMP(NDX) .EQ. DSMC3B) .OR.
     $           (TEMP(NDX) .EQ. DSMC20) .OR.
     $           (TEMP(NDX) .EQ. DSMC30)) THEN
           REQMT(4,J) = 1
        ELSE
           REQMT(4,J) = 0
        END IF

        IF ((TEMP(NDX) .EQ. SYS225) .OR.
     $           (TEMP(NDX) .EQ. ALC001)) THEN
           REQMT(5,J) = 1
        ELSE
           REQMT(5,J) = 0
```

```
      END IF

************************************************************
*  THIS REQUIREMENT WILL BE GRANTED REGUARDLESS, SINCE    *
*  AFIT DOES NOT OFFER THIS COURSE, AND HENCE IT CAN NOT  *
*  BE APART OF THE IMPACT OF COURSE DEMAND.               *
************************************************************

      IF ((TEMP(NDX) .EQ. DSMC3A) .OR.
    $           (TEMP(NDX) .EQ. DSMC3B)) THEN
         REQMT(6,J) = 1
      ELSE
         REQMT(6,J) = 1
      END IF
10   CONTINUE

     DO 35 I = 1, 6
        DO 45 J = 1, 34
           TEST(I) = TEST(I) + REQMT(I,J)
45      CONTINUE
35   CONTINUE


     DO 65 S = 1, 6
        IF (TEST(S) .GT. 0) THEN
           CHK(S) = 1
        ELSE
           CHK(S) = 0
        END IF
65   CONTINUE

************************************************************
****
*  Using a set list of possible acquisition courses, the next
*
*  Do-loop check for the number of specialty courses a person
*
*  has completed.
 *
************************************************************
****

     DO 20 K = 1,34
        NDX = K + 16
        OPEN(UNIT=12, FILE='ACQLOG.DAT', STATUS='OLD')
        READ(12,120,END=15) PDS, NUM
153      IF (TEMP(NDX) .NE. NUM) THEN
           READ(12,120,END=15) PDS, NUM
           GO TO 153
```

```
         END IF
15        IF ((TEMP(NDX) .EQ. NUM) .AND. (TEMP(NDX) .GT. 0))
THEN
            COUNT = COUNT + 1
         END IF
         CLOSE(12)
20    CONTINUE

      IF (COUNT .GE. 1) THEN
         CHK(7) = 1
      ELSE
         CHK(7) = 0
      END IF

      IF (COUNT .GE. 3) THEN
         CHK(8) = 1
      ELSE
         CHK(8) = 0
      END IF


************************************************************
****
*  The follow condition statements assign an evaluated level
of*
*   of certification to each person based on the previous
checks*
*  for the requirements at each level.
 *
************************************************************
****

      IF (((CHK(1) .EQ. 1) .AND. (CHK(3) .EQ. 1) .AND.
     $      (CHK(7) .EQ. 1)) .OR. ((CHK(2) .EQ. 1) .AND.
     $      (CHK(3) .EQ. 1)) .OR. ((CHK(3) .EQ. 1) .AND.
     $      (CHK(7) .EQ. 1)) .OR. ((CHK(4) .EQ. 1) .AND.
     $      (CHK(7) .EQ. 1)) .OR. ((CHK(6) .EQ. 1) .AND.
     $      (CHK(7) .EQ. 1))) THEN
         ELEVL = ELEVL + 1
         LEVL(1) = 1
      ELSE
         LEVL(1) = 0
      END IF

      IF (((LEVL(1) .EQ. 1) .AND. (CHK(4) .EQ. 1) .AND.
     $      (CHK(8) .EQ. 1)) .OR. ((LEVL(1) .EQ. 1) .AND.
     $      (CHK(6) .EQ. 1) .AND. (CHK(8) .EQ. 1))) THEN
         ELEVL = ELEVL + 1
         LEVL(2) = 1
      ELSE
```

```
         LEVL(2) = 0
      END IF

      IF ((LEVL(2) .EQ. 1) .AND. (CHK(6) .EQ. 1) .AND.
     $    (TEMP(15) .GE. 1)) THEN
         ELEVL = ELEVL + 1
         LEVL(3) = 1
      ELSE
         LEVL(3) = 0
      END IF

************************************************************
****
*  The following if structure checks to see if the evaluated
 *
*  level of certification is greater than the one reported in
*
*  the individual's records.  The greater certification level
*
*  will be used during the simulation.
 *
************************************************************
****

      IF (ELEVL .GE. TEMP(11)) THEN
         TEMP(61) = ELEVL
      ELSE
         TEMP(61) = TEMP(11)
      END IF



************************************************************
****
*  The following set of conditional statements compute the
 *
*  initial states for backlog population of people, backlog
 *
*  demand for courses and total demand for courses.
 *
************************************************************
****

      IF ((TEMP(3) .EQ. 1) .AND. (TEMP(13) .GE. 2)) THEN
         TEMP(63) = 1
      END IF
```

```
      IF ((TEMP(3) .GT. TEMP(61)) .AND. (TEMP(63) .EQ. 1))
THEN
         BKLOG = 1
         TEMP(62) = 1
      ELSE
         BKLOG = 0
         TEMP(62) = 0
      END IF

      IF ((LEVL(1) .EQ. 0) .AND. (CHK(1) .EQ. 0) .AND.
     $   (CHK(2) .EQ. 0)) THEN
         TEMP(51) = 1
         DMD100 = 1
         CUTOFF = UNFRM(0.0, 1.0, 2)
         IF (CUTOFF .GE. 0.5) THEN
            CRS100 = 1
         END IF
      ELSE
         DMD100 = 0
         TEMP(51) = 0
      END IF


      IF ((LEVL(1) .EQ. 0) .AND. (CHK(3) .EQ. 0) .AND.
     $(CHK(2) .EQ. 0)) THEN
         TEMP(52) = 1
         DMD200 = 1
         CUTOFF = UNFRM(0.0, 1.0, 1)
         IF (CUTOFF .GE. 0.5) THEN
            CRS200 = 1
         END IF
      ELSE
         DMD200 = 0
         TEMP(52) = 0
      END IF

      IF ((LEVL(1) .EQ. 0) .AND. (CHK(7) .EQ. 0)) THEN
         TEMP(53) = 1
         SPCRS = 1
      ELSE
         SPCRS = 0
         TEMP(53) = 0
      END IF

      IF ((LEVL(2) .EQ. 0) .AND. (CHK(4) .EQ. 0)) THEN
         TEMP(54) = 1
         DMD400 = 1
         CUTOFF = UNFRM(0.0, 1.0, 3)
         IF (CUTOFF .GE. 0.20) THEN
            CRS400 = 1
```

```
         END IF
      ELSE
         DMD400 = 0
         TEMP(54) = 0
      END IF

      IF ((LEVL(2) .EQ. 0) .AND. (CHK(5) .EQ. 0)) THEN
         TEMP(55) = 1
         DMD225 = 1
         CUTOFF = UNFRM(0.0, 1.0, 2)
         IF (CUTOFF .GE. 0.5) THEN
            CRS225 = 1
         END IF
      ELSE
         DMD225 = 0
         TEMP(55) = 0
      END IF

      IF ((LEVL(2) .EQ. 0) .AND. (CHK(8) .EQ. 0)) THEN
         TEMP(56) = 1
         SPCRS = SPCRS + 1
      ELSE
         TEMP(56) = 0
      END IF

      IF ((LEVL(3) .EQ. 0) .AND. (CHK(8) .EQ. 0)) THEN
         TEMP(57) = 1
         SPCRS = SPCRS + 1
      ELSE
         TEMP(57) = 0
      END IF

      IF ((LEVL(3) .EQ. 0) .AND. (CHK(6) .EQ. 0)) THEN
         TEMP(58) = 0
         PMC03 = 1
      ELSE
         PMC03 = 0
         TEMP(58) = 0
      END IF

120   FORMAT(A3, 2X, I4)
      RETURN
      END

************************************************************
****
*  The purpose of subroutine CRSDMD is to keep statistics on
 *
```

```
*  the backlog population and course demand.
 *
****************************************************************
****

      SUBROUTINE CRSDMD(BAKLOG, HI100, LO100, HI200, LO200,
     $HI400, LO400, HI225, LO225, DSMC03, YEAR, SPECLY)

      INTEGER BAKLOG, HI100, LO100, HI200, LO200, HI400,
     $LO400, HI225, LO225, DSMC03, YEAR, SPECLY


      WRITE(50, 220) YEAR, BAKLOG, HI100, LO100, HI200,
     $LO200, HI400, LO400, HI225, LO225, DSMC03, SPECLY

220   FORMAT(I6, 1X, I6, 1X, I5, 1X, I5, 1X, I5, 1X, I5, 1X,
     $I5, 1X, I5, 1X, I5, 1X, I5, 1X, I5, 1X, I5, 1X, I5)

      PRINT *, 'DONE WITH COURSE DEMAND'

      RETURN
      END

****************************************************************
****
*  The purpose of subroutine STATS  is to keep statistics on
 *
*  the population of people certified at each level.
 *
****************************************************************
****

      SUBROUTINE STATS
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER YEAR

      REAL LEVL1, LEVL2, LEVL3, RLEVL1, RLEVL2, RLEVL3, CORPS,
     $CRIT, DEVELP, RELATE, NONE, FILLCR, FILRLT, FILDVP

      REAL MOCNT1, MOCNT2, MOCNT3, MOCRIT, TTCNT2, TTCNT1,
     $EXCNT3, DPCNT1, DPCNT2, DPCNT3, DPCRIT, EXCNT2,
     $DPRELT, DPDVLP, EXCNT1, TTNONE, TTCNT3

      OPEN(UNIT=35, FILE='LEVEL.OUT',STATUS='NEW')
      PRINT *,'STARTING STATS'
      LEVL1 = 0
      LEVL2 = 0
      LEVL3 = 0
```

```
      RLEVL1 = 0
      RLEVL2 = 0
      RLEVL3 = 0
      FILLCR = 0
      RELATE = 0
      DEVELP = 0
      CORPS = 0
      NONE = 0
      FILDVP = 0
      FILRLT = 0
      CRIT = 0
      DEVELP = 0



      DO 10 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            YEAR = AUX(I,68)
            IF (AUX(I,61) .EQ. 1) THEN
               LEVL1 = LEVL1 + 1
            ELSE IF (AUX(I,61) .EQ. 2) THEN
               LEVL2 = LEVL2 + 1
            ELSE IF (AUX(I,61) .EQ. 3) THEN
               LEVL3 = LEVL3 + 1
            ELSE IF (AUX(I,61) .EQ. 0) THEN
               NONE = NONE + 1
            END IF
            IF (AUX(I,16) .EQ. 1) THEN
               CORPS = CORPS + 1
            END IF
            IF (AUX(I,4) .EQ. 1) THEN
               FILLCR = FILLCR + 1
            ELSE IF (AUX(I,4) .EQ. 2) THEN
               FILDVP = FILDVP + 1
            ELSE IF (AUX(I,4) .EQ. 3) THEN
               FILRLT = FILRLT + 1
            END IF
         END IF
10    CONTINUE

      DO 20 I = 1, ROWS
         IF (AUX(I,3) .EQ. 1) THEN
            RLEVL1 = RLEVL1 + 1
         ELSE IF (AUX(I,3) .EQ. 2) THEN
            RLEVL2 = RLEVL2 + 1
         ELSE IF (AUX(I,3) .EQ. 3) THEN
            RLEVL3 = RLEVL3 + 1
         END IF

         IF (AUX(I,4) .EQ. 1) THEN
```

```
            CRIT = CRIT + 1
         ELSE IF (AUX(I,4) .EQ. 2) THEN
            DEVELP = DEVELP + 1
         ELSE IF (AUX(I,4) .EQ. 3) THEN
            RELATE = RELATE + 1
         END IF
20    CONTINUE

******WORKFORCE POSITION STATISTICS (2 PIE CHARTS)


      MOCNT1 = (RLEVL1/ROWS)*100
      MOCNT2 = (RLEVL2/ROWS)*100
      MOCNT3 = (RLEVL3/ROWS)*100

      MOCRIT = (CRIT/ROWS)*100


******WORKFORCE PERSONNEL STATISTICS (3 BAR CHARTS AND 1 PIE
CHART)


      IF (RLEVL1 .GT. 0) THEN
         DPCNT1 = (LEVL1/RLEVL1)*100
      END IF

      IF (RLEVL2 .GT. 0) THEN
         DPCNT2 = (LEVL2/RLEVL2)*100
      END IF

      IF (RLEVL3 .GT. 0) THEN
         DPCNT3 = (LEVL3/RLEVL3)*100
      END IF


      IF (RLEVL1 .GT. 0) THEN
         EXCNT1 = ((LEVL1/RLEVL1)/ROWS)*100
      END IF

      IF (RLEVL2 .GT. 0) THEN
         EXCNT2 = ((LEVL2/RLEVL2)/ROWS)*100
      END IF

      IF (RLEVL3 .GT. 0) THEN
         EXCNT3 = ((LEVL3/RLEVL3)/ROWS)*100
      END IF


******MEASURE OF TRAINED COMPAIRED TO WORKFORCE (1 PIE CHARTS)
```

```
      TTCNT1 = (LEVL1/ROWS)*100
      TTCNT2 = (LEVL2/ROWS)*100
      TTCNT3 = (LEVL3/ROWS)*100
      TTNONE = (NONE/ROWS)*100

******STATISTICS ON FILLED ACQUISITION POSITIONS (1 PIE CHART)


      IF (CRIT .GT. 0) THEN
         DPCRIT = (FILLCR/CRIT)*100
      END IF

      IF (DEVELP .GT. 0) THEN
         DPDVLP = (FILDVP/DEVELP)*100
      END IF

      IF (RELATE .GT. 0) THEN
         DPRELT = (FILRLT/RELATE)*100
      END IF

******FILE OUTPUT

      WRITE(35, 135) YEAR, ROWS

      WRITE(35, 145) RLEVL1, LEVL1, RLEVL2, LEVL2, RLEVL3,
LEVL3,
     $CRIT, FILLCR, DEVELP, FILDVP, RELATE, FILRLT

      WRITE(35, 155) MOCNT1, MOCNT2, MOCNT3, MOCRIT, DPCNT1,
DPCNT2,
     $DPCNT3, EXCNT1, EXCNT2, EXCNT3

      WRITE(35, 165) TTCNT1, TTCNT2, TTCNT3, TTNONE, DPCRIT,
DPDVLP,
     $DPRELT

135   FORMAT(I5, 1X, I6)
145   FORMAT(12(F5.0, 1X))
155   FORMAT(10(F5.2, 1X))
165   FORMAT(7(F5.2, 1X))

      PRINT *, 'DONE WITH STATS'
      RETURN
      END


************************************************************
****
*  The following subroutine is used to attrite people from the
*
```

```
*  system based on passover promotion opportunities and yearly
*
*   attrition rates for each year of service.
 *
**********************************************************************
****

      SUBROUTINE ATTRIT
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER NDX1, YR, POP

      REAL YOS(28), KILL


      CALL SCHDL(26, 365.0, ATRIB)

      WRITE(19, *) NNQ(15),' ENTERED ATTRIT'

      YOS(1) = .9757
      YOS(2) = .9805
      YOS(3) = .9536
      YOS(4) = .8565
      YOS(5) = .8942
      YOS(6) = .9188
      YOS(7) = .9370
      YOS(8) = .9418
      YOS(9) = .9463
      YOS(10) = .9590
      YOS(11) = .9343
      YOS(12) = .9051
      YOS(13) = .9919
      YOS(14) = .9926
      YOS(15) = .9944
      YOS(16) = .9843
      YOS(17) = .9507
      YOS(18) = .9522
      YOS(19) = .8827
      YOS(20) = .5469
      YOS(21) = .5969
      YOS(22) = .4072
      YOS(23) = .2879
      YOS(24) = .2124
      YOS(25) = .1713
      YOS(26) = .1493
      YOS(27) = .2090
      YOS(28) = .0000

      DO 10 I = 1, ROWS
```

```
          IF ((AUX(I,13)) .GT. 28) THEN
             AUX(I,13) = 28
          END IF
          YR = (AUX(I,13))
          KILL = UNFRM(0.0, 1.0, 2)
          IF ((AUX(I,6) .EQ. 1) .AND. (AUX(I,8) .LT. 7) .AND.
     $        ((AUX(I,69) .GE. 2) .OR. (KILL .GT. YOS(YR))))
THEN
             POP = NNQ(15)
             DO 19 P = 1, POP
                CALL RMOVE(1, 15, ATRIB)
                IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                   DO 20 J = 1, 65
                      NDX1 = J + 5
                      AUX(I,NDX1) = 0
20                 CONTINUE
                   GO TO 10
                ELSE
                   CALL FILEM(15, ATRIB)
                END IF
19           CONTINUE
          END IF
10     CONTINUE

       WRITE(19, *) NNQ(15),' LEFT ATTRIT'

       PRINT *, 'DONE WITH ATTRIT'

       RETURN
       END


*************************************************************
****
*  The following routine is used to promote personnel in the
 *
*  system based on promotion rates for each rank.  Three and
 *
*  Four star general officers are promoted based on the
 *
*  attrition of other equivalent officers in the system.
 *
*************************************************************
****


       SUBROUTINE PRMOTE
       INCLUDE '/usr/local/Slam/PARAM.INC'
       INCLUDE 'USER.ONE'
```

```
      INTEGER POP, RANK

      REAL PRATE(10), PASS, MAJBTZ, MAJABZ, LTCBTZ, LTCABZ,
     $COLBTZ, COLABZ
      WRITE(19,*) NNQ(15),' ENTERED PROMOTE'
      CALL SCHDL(25, 365.0, ATRIB)
      RANK = 0
      PRATE(1) = .97
      PRATE(2) = .94
      MAJBTZ = .012
      PRATE(3) = .673
      MAJABZ = .095
      LTCBTZ = .025
      PRATE(4) = .62
      LTCABZ = .059
      COLBTZ = .023
      PRATE(5) = .427
      COLABZ = .032
      PRATE(6) = .0135
      PRATE(7) = .35

      DO 10 I = 1, ROWS
         IF ((AUX(I,6) .EQ. 1) .AND. (AUX(I,8) .LE. 7)) THEN
            POP = NNQ(15)
            DO 19 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 21
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
19          CONTINUE
21          PASS = UNFRM(0.0, 1.0, 2)
            IF (((AUX(I,8)) .EQ. 1) .AND.
     $         ((AUX(I,13)) .EQ. 2)) THEN
               IF (PASS .LE. PRATE(1)) THEN
                  AUX(I,8) = AUX(I,8) + 1
                  ATRIB(8) = ATRIB(8) + 1
                  CALL FILEM(15,ATRIB)
               ELSE
                  AUX(I,69) = AUX(I,69) + 1
                  ATRIB(69) = ATRIB(69) + 1
                  CALL FILEM(15, ATRIB)
               END IF
            ELSE IF (((AUX(I,8)) .EQ. 2) .AND.
     $         ((AUX(I,13)) .EQ. 4)) THEN
               IF (PASS .LE. PRATE(2)) THEN
                  AUX(I,8) = AUX(I,8) + 1
                  ATRIB(8) = ATRIB(8) + 1
```

```
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
       ELSE IF (((AUX(I,8)) .EQ. 3) .AND.
     $    ((AUX(I,13)) .EQ. 11)) THEN
         IF (PASS .LE. MAJBTZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 3) .AND.
     $    ((AUX(I,13)) .EQ. 12)) THEN
         IF (PASS .LE. PRATE(3)) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 3) .AND.
     $    ((AUX(I,13)) .GT. 12)) THEN
         IF (PASS .LE. MAJABZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 4) .AND.
     $    ((AUX(I,13)) .EQ. 15)) THEN
         IF (PASS .LE. LTCBTZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 4) .AND.
     $    ((AUX(I,13)) .EQ. 16)) THEN
         IF (PASS .LE. PRATE(4)) THEN
            AUX(I,8) = AUX(I,8) + 1
```

```
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 4) .AND.
$        ((AUX(I,13)) .GT. 15)) THEN
         IF (PASS .LE. LTCABZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 5) .AND.
$        ((AUX(I,13)) .EQ. 20)) THEN
         IF (PASS .LE. COLBTZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 5) .AND.
$        ((AUX(I,13)) .EQ. 21)) THEN
         IF (PASS .LE. PRATE(5)) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 5) .AND.
$        ((AUX(I,13)) .GT. 21)) THEN
         IF (PASS .LE. COLABZ) THEN
            AUX(I,8) = AUX(I,8) + 1
            ATRIB(8) = ATRIB(8) + 1
            CALL FILEM(15, ATRIB)
         ELSE
            AUX(I,69) = AUX(I,69) + 1
            ATRIB(69) = ATRIB(69) + 1
            CALL FILEM(15, ATRIB)
         END IF
      ELSE IF (((AUX(I,8)) .EQ. 6) .AND.
```

```
     $          ((AUX(I,13)) .EQ. 24)) THEN
                IF (PASS .LE. PRATE(6)) THEN
                   AUX(I,8) = AUX(I,8) + 1
                   ATRIB(8) = ATRIB(8) + 1
                   CALL FILEM(15, ATRIB)
                ELSE
                   AUX(I,69) = AUX(I,69) + 1
                   ATRIB(69) = ATRIB(69) + 1
                   CALL FILEM(15, ATRIB)
                END IF
             ELSE IF (((AUX(I,8)) .EQ. 7) .AND.
     $          ((AUX(I,13)) .EQ. 26)) THEN
                IF (PASS .LE. PRATE(7)) THEN
                   AUX(I,8) = AUX(I,8) + 1
                   ATRIB(8) = ATRIB(8) + 1
                   CALL FILEM(15, ATRIB)
                ELSE
                   AUX(I,69) = AUX(I,69) + 1
                   ATRIB(69) = ATRIB(69) + 1
                   CALL FILEM(15, ATRIB)
                END IF
             ELSE
                CALL FILEM(15, ATRIB)
             END IF
          END IF
10    CONTINUE

*************************************************************
****
*    The following condition provides acquisition corp
membership*
*  based on a promotion to at least major and an evaluated
 *
*  certification  level of at least 2.
 *
*************************************************************
****

      DO 50 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            IF ((AUX(I,61) .GE. 2) .AND. (AUX(I,8) .GE. 4))
THEN
               AUX(I,16) = 1
            END IF
         END IF
50    CONTINUE

      DO 60 I = 1, NNQ(15)
         CALL RMOVE(1, 15, ATRIB)
         IF ((ATRIB(61) .GE. 2) .AND. (ATRIB(8) .GE. 4)) THEN
```

```
            ATRIB(16) = 1
            CALL FILEM(15, ATRIB)
         ELSE
            CALL FILEM(15, ATRIB)
         END IF
60    CONTINUE

      WRITE(19, *) NNQ(15),' LEFT PROMOTE'
      RETURN
      END


************************************************************
****
*  The purpose of subroutine ACCESS is to bring new second
 *
*  lieutenants in the system for simulation purposes.  Each
new*
*  lieutenant will have identical attributes.  They will be
 *
*  accessed according the military acquisition career field
 *
*  accession rate.  Vacant positions will be searched for
 *
*  lieutenant requirements and filled according to the career
*
*  field rate.
 *
************************************************************
****


      SUBROUTINE ACCESS
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      REAL NEW, SCHOOL, PERCNT

      INTEGER NDX1, POP, NEED, CURRNT, GOAL, FILL

      CALL SCHDL(27, 365.0, ATRIB)

************************************************************
***
*  The following code calculated the number of accessions
*
*  required to maintain 85% manning of acquisition positions.
*
```

```
*  If the goal is under, and lieutenant positions are vacant,
*
*  accessions are made, else no acessions are made.
*
****************************************************************
***
      WRITE(19, *) NNQ(15),' ENTERED ACCESS'
      FILL = 0
      PERCNT = 85.0
      GOAL = NINT(PERCNT*ROWS)
      CURRNT = NNQ(15)
      NEED = GOAL - CURRNT


      DO 10 I = 1, ROWS
         IF ((NEED .GT. 0) .AND. (FILL .LE. NEED)) THEN
            IF ((AUX(I,6) .EQ. 0) .AND. ((AUX(I,2) .EQ. 1)
.OR.
     $      (AUX(I,2) .EQ. 2))) THEN
               DO 20 J = 1 ,65
                  NDX1 = J + 5
                  AUX(I,NDX1) = 0
20             CONTINUE
               AUX(I,6) = 1
               AUX(I,7) = AUX(I,1)
               AUX(I,8) = 1
               AUX(I,9) = 1
               AUX(I,10) = NINT(UNFRM(1.0, 22.0, 2))
               AUX(I,13) = 1
               SCHOOL = UNFRM(0.0, 1.0, 2)
               IF (SCHOOL .GE. 0.5) THEN
                  AUX(I,15) = 1
               ELSE
                  AUX(I,15) = 2
               END IF
               AUX(I,16) = 0
               AUX(I,11) = 1
               AUX(I,12) = AUX(I,5)
               AUX(I,63) = 0
               AUX(I,68) = NINT(TNOW/365) + 1992
               NEW = UNFRM(0.0, 1.0, 2)
               IF (NEW .GT. .5) THEN
                  AUX(I,14) = 0
               ELSE
                  AUX(I,14) = 1
               END IF
*             POP = NNQ(15)
*             DO 19 P = 1, POP
*                CALL RMOVE(1, 15, ATRIB)
*                IF (ATRIB(7) .EQ. AUX(I,7)) THEN
```

```
*                    GO TO 21
*                 ELSE
*                    CALL FILEM(15, ATRIB)
*                 END IF
*19            CONTINUE
21                DO 15 L = 1, 70
                     ATRIB(L) = AUX(I,L)
15                CONTINUE
                  CALL FILEM(15, ATRIB)
               END IF
               FILL = FILL + 1
            END IF
10    CONTINUE
      WRITE(19, *) NNQ(15),' LEFT ACCESS'
      RETURN
      END

************************************************************
****
*   The  following  routine  represents  the  selective  early
retire-*
*  ment boards (SERB).  A board will be conducted each year
for*
*  the next five years and then this subroutine will become
 *
*  inactive.
 *
************************************************************
****

      SUBROUTINE SERB
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER NDX1, POP

      REAL OUT

      WRITE(19, *) NNQ(15),' ENTERED SERB'
      IF (ICOUNT .LT. 6) THEN
         CALL SCHDL(30, 365.0, ATRIB)
         ICOUNT = ICOUNT  + 1
      END IF

      DO 10 I = 1, ROWS
         OUT = UNFRM(0.0, 1.0, 2)
         IF (((AUX(I, 8) .EQ. 3) .OR. (AUX(I,8) .EQ. 4)) .AND.
     $   (AUX(I,13) .GE. 20) .AND. (OUT .LE. 0.30)) THEN
            POP = NNQ(15)
```

```
            DO 19 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 21
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
19          CONTINUE
21          DO 20 J = 1, 65
               NDX1 = J + 5
               AUX(I,NDX1) = 0
20          CONTINUE
         END IF
10    CONTINUE
      WRITE(19, *) NNQ(15),' LEFT SERB'
      RETURN
      END

************************************************************
****
*  Subroutine BATCH is used to collect all entities from the
 *
*  various school files and place them in file 15.  This
action*
*  is in preparation for the different population routines
such*
*  promotions and attritions which require all entities to be
*
*  in a single file.
 *
************************************************************
****

      SUBROUTINE BATCH
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER POP, NDX1, YEAR
      WRITE(19, *) NNQ(15), 'ENTERED BATCH'
      CALL SCHDL(36, 365.0, ATRIB)

       YEAR = NINT(TNOW/365) + 1992


      IF (NNQ(11) .GT. 0) THEN
         DO 15 I = 1, NNQ(11)
            CALL RMOVE(1, 11, ATRIB)
            CALL FILEM(15, ATRIB)
15       CONTINUE
      END IF
```

```
      IF (NNQ(21) .GT. 0) THEN
         DO 20 I = 1, NNQ(21)
            CALL RMOVE(1, 21, ATRIB)
            CALL FILEM(15, ATRIB)
20       CONTINUE
      END IF

      IF (NNQ(25) .GT. 0) THEN
         DO 30 I = 1, NNQ(25)
            CALL RMOVE(1, 25, ATRIB)
            CALL FILEM(15, ATRIB)
30       CONTINUE
      END IF

      IF (NNQ(41) .GT. 0) THEN
         DO 40 I = 1, NNQ(41)
            CALL RMOVE(1, 41, ATRIB)
            CALL FILEM(15, ATRIB)
40       CONTINUE
      END IF

      IF (NNQ(31) .GT. 0) THEN
         DO 50 I = 1, NNQ(31)
            CALL RMOVE(1, 31, ATRIB)
            CALL FILEM(15, ATRIB)
50       CONTINUE
      END IF

      IF (NNQ(1) .GT. 0) THEN
         DO 60 I = 1, NNQ(1)
            CALL RMOVE(1, 1, ATRIB)
            CALL FILEM(15, ATRIB)
60       CONTINUE
      END IF

      IF (NNQ(2) .GT. 0) THEN
         DO 70 I = 1, NNQ(2)
            CALL RMOVE(1, 2, ATRIB)
            CALL FILEM(15, ATRIB)
70       CONTINUE
      END IF

      IF (NNQ(22) .GT. 0) THEN
         DO 80 I = 1, NNQ(22)
            CALL RMOVE(1, 22, ATRIB)
            CALL FILEM(15, ATRIB)
80       CONTINUE
      END IF

      IF (NNQ(3) .GT. 0) THEN
```

```
            DO 90 I = 1, NNQ(3)
               CALL RMOVE(1, 3, ATRIB)
               CALL FILEM(15, ATRIB)
90          CONTINUE
         END IF

         IF (NNQ(4) .GT. 0) THEN
            DO 100 I = 1, NNQ(4)
               CALL RMOVE(1, 4, ATRIB)
               CALL FILEM(15, ATRIB)
100         CONTINUE
         END IF
         WRITE(19,*) NNQ(15),' LEFT BATCH'
         OPEN(UNIT=27,FILE='TRNG.OUT', STATUS='NEW')
         WRITE(27, *) 'TRAINING CAPABILITY FOR SYS100'
         WRITE(27, *) YEAR, GD100
         WRITE(27, *) 'TRAINING CAPABILITY FOR SYS200'
         WRITE(27, *) YEAR, GD200
         WRITE(27, *) 'TRAINING CAPABILITY FOR SYS400'
         WRITE(27, *) YEAR, GD400
         WRITE(27, *) 'TRAINING CAPABILITY FOR SYS225'
         WRITE(27, *) YEAR, GD225
         WRITE(27,   *)   'TRAINING   CAPABILITY   FOR   SPEACIALTY
COURSES'
         WRITE(27, *) YEAR, GDSPL

         WRITE(19,  *)  'CHECKING  THE  SCHOOL  AND  GRADUATIOIN
ROUTINES'
         WRITE(19, *) CNT100,' ENTERED SYS100'
         WRITE(19, *) GD100,' LEFT SYS100'
         WRITE(19, *) CNT200,' ENTERED SYS200'
         WRITE(19, *) GD200,' LEFT SYS200'
         RETURN
         END


************************************************************
****
*  Subroutine UPDATE is used to update critical attributes of
*
*  personnel that is not updated during the regular simulation
*
*  and other support routines.
 *
************************************************************
****

         SUBROUTINE UPDATE
```

```
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER POP, NDX1, NDX2, CHK

      CALL SCHDL(29, 365.0, ATRIB)

************************************************************
******
*  The following condition updates the years of service,
calendar*
*  year and provides an educational level of masters by the
   *
*  eighth year of service.
   *
************************************************************
******
      WRITE(19, *) NNQ(15),' ENTERED UPDATE'
      DO 10 I = 1, ROWS
         IF ((AUX(I,6)) .EQ. 1) THEN
            AUX(I,13) = AUX(I,13) + 1
            AUX(I,68) = AUX(I,68) + 1
            IF ((AUX(I,13) .GE. 8) .AND. (AUX(I,15) .LT. 1))
THEN
               AUX(I,15) = 1
            END IF
         END IF
10    CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 1'

      POP = NNQ(15)
      DO 194 P = 1, POP
         CALL RMOVE(1, 15, ATRIB)
         ATRIB(13) = ATRIB(13) + 1
         ATRIB(68) = ATRIB(68) + 1
         IF ((ATRIB(13) .GE. 8) .AND. (ATRIB(15) .LT. 1)) THEN
            ATRIB(15) = 1
         END IF
         CALL FILEM(15, ATRIB)
194   CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 2'

      DO 25 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 19 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
```

```
                 GO TO 21
              ELSE
                 CALL FILEM(15, ATRIB)
              END IF
19         CONTINUE
21         IF ((ATRIB(16) .EQ. 1) .AND. (ATRIB(4) .NE. 1))
THEN
              CALL FILEM(13, ATRIB)
           ELSE
              CALL FILEM(15, ATRIB)
           END IF
        END IF
25    CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 3'

      DO 35 I = 1, ROWS
         IF ((AUX(I,6) .EQ. 0) .AND. ((AUX(I,4)) .EQ. 1)) THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 26 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1045 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1047 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1047                       CONTINUE
                           GO TO 1046
                        END IF
1045                 CONTINUE
1046                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 45 J = 1, 64
                        NDX2 = J + 6
                        AUX(I,NDX2) = ATRIB(NDX2)
45                   CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 46 L = 1, NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
46                      CONTINUE
                     END IF
                  ELSE
```

```
                    CALL FILEM(14, ATRIB)
                 END IF
26            CONTINUE
           END IF
           IF (NNQ(14) .GT. 0) THEN
              DO 47 L = 1, NNQ(14)
                 CALL RMOVE(1, 14, ATRIB)
                 CALL FILEM(13, ATRIB)
47            CONTINUE
           END IF
        END IF
35    CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 4'

      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 55 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
55       CONTINUE
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK4.5'

      DO 65 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 119 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 121
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
119         CONTINUE
121          IF ((ATRIB(2) .EQ. 9) .AND. (ATRIB(8) .GE. 9))
THEN
               CALL FILEM(13, ATRIB)
            ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
65    CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 5'

      DO 75 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .EQ. 10))
THEN
            IF (NNQ(13) .GT. 0) THEN
```

```
                  DO 76 K = 1, NNQ(13)
                     CALL RMOVE(1, 13, ATRIB)
                     IF (ATRIB(61) .GE. AUX(I,3)) THEN
                        DO 1001 A = 1, ROWS
                           IF ((AUX(A,6) .EQ. 1) .AND.
     $                     (AUX(A,7) .EQ. ATRIB(7))) THEN
                              DO 1002 Z = 1, 65
                                 NDX1 = Z + 5
                                 AUX(A,NDX1) = 0
1002                          CONTINUE
                              GO TO 1003
                           END IF
1001                    CONTINUE
1003                    ATRIB(1) = AUX(I,1)
                        ATRIB(2) = AUX(I,2)
                        ATRIB(3) = AUX(I,3)
                        ATRIB(4) = AUX(I,4)
                        ATRIB(5) = AUX(I,5)
                        ATRIB(6) = 1
                        AUX(I,6) = 1
                        DO 85 J = 1, 64
                           NDX1 = J + 6
                           AUX(I,J) = ATRIB(J)
85                      CONTINUE
                        CALL FILEM(15, ATRIB)
                        IF (NNQ(14) .GT. 0) THEN
                           DO 86 L = 1,NNQ(14)
                              CALL RMOVE(1, 14, ATRIB)
                              CALL FILEM(13, ATRIB)
86                         CONTINUE
                        END IF
                     ELSE
                        CALL FILEM(14, ATRIB)
                     END IF
76                CONTINUE
                  IF (NNQ(14) .GT. 0) THEN
                     DO 87 L = 1,NNQ(14)
                        CALL RMOVE(1, 14, ATRIB)
                        CALL FILEM(13, ATRIB)
87                   CONTINUE
                  END IF
               END IF
            END IF
75       CONTINUE

*         WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 6'

         IF (NNQ(13) .GT. 0) THEN
            NDX1 = NNQ(13)
            DO 88 I = 1, NDX1
```

```
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
88       CONTINUE
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 7'
      DO 95 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 219 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 221
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
219         CONTINUE
221          IF ((ATRIB(2) .EQ. 8) .AND. (ATRIB(8) .GE. 8))
THEN
               CALL FILEM(13, ATRIB)
            ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
95    CONTINUE


*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 8'

      DO 105 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 9))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 106 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1004 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1005 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1005                       CONTINUE
                           GO TO 1006
                        END IF
1004                 CONTINUE
1006                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
```

```
                  ATRIB(5) = AUX(I,5)
                  ATRIB(6) = 1
                  AUX(I,6) = 1
                  DO 115 J = 1, 64
                     NDX1 = J + 6
                     AUX(I,J) = ATRIB(J)
115               CONTINUE
                  CALL FILEM(15, ATRIB)
                  IF (NNQ(14) .GT. 0) THEN
                     DO 116 L = 1,NNQ(14)
                        CALL RMOVE(1, 14, ATRIB)
                        CALL FILEM(13, ATRIB)
116                  CONTINUE
                  END IF
               ELSE
                  CALL FILEM(14, ATRIB)
               END IF
106         CONTINUE
            IF (NNQ(14) .GT. 0) THEN
               DO 107 L = 1,NNQ(14)
                  CALL RMOVE(1, 14, ATRIB)
                  CALL FILEM(13, ATRIB)
107            CONTINUE
            END IF
         END IF
      END IF
105   CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 9'
      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 125 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
125      CONTINUE
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 10'

      DO 135 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 319 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                 GO TO 321
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
319         CONTINUE
```

```
321          IF ((ATRIB(2) .EQ. 7) .AND. (ATRIB(8) .GE. 7))
THEN
               CALL FILEM(13, ATRIB)
            ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
135   CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 11'

      DO 145 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 8))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 146 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1007 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1008 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1008                       CONTINUE
                           GO TO 1009
                        END IF
1007                 CONTINUE
1009                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 155 J = 1, 64
                        NDX1 = J + 6
                        AUX(I,J) = ATRIB(J)
155                  CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 156 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
156                     CONTINUE
                     END IF
                  ELSE
                     CALL FILEM(14, ATRIB)
                  END IF
```

```
146              CONTINUE
                 IF (NNQ(14) .GT. 0) THEN
                    DO 157 L = 1,NNQ(14)
                       CALL RMOVE(1, 14, ATRIB)
                       CALL FILEM(13, ATRIB)
157                 CONTINUE
                 END IF
              END IF
           END IF
145     CONTINUE

* *      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 12'

        IF (NNQ(13) .GT. 0) THEN
           NDX1 = NNQ(13)
           DO 165 I = 1, NDX1
              CALL RMOVE(1,13,ATRIB)
              CALL FILEM(15, ATRIB)
165        CONTINUE
        END IF
*        WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 13'


        DO 175 I = 1, ROWS
           IF (AUX(I,6) .EQ. 1) THEN
              POP = NNQ(15)
              DO 419 P = 1, POP
                 CALL RMOVE(1, 15, ATRIB)
                 IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                    GO TO 421
                 ELSE
                    CALL FILEM(15, ATRIB)
                 END IF
419           CONTINUE
421            IF ((ATRIB(2) .EQ. 6) .AND. (ATRIB(8) .GE. 6))
THEN
                 CALL FILEM(13, ATRIB)
              ELSE
                 CALL FILEM(15, ATRIB)
              END IF
           END IF
175     CONTINUE

*        WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 14'

        DO 185 I = 1, ROWS
           IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 7))
THEN
              IF (NNQ(13) .GT. 0) THEN
                 DO 186 K = 1, NNQ(13)
```

```
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1010 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                     (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1011 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
 1011                      CONTINUE
                           GO TO 1012
                        END IF
 1010                CONTINUE
 1012                ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 195 J = 1, 64
                        NDX1 = J + 6
                        AUX(I,J) = ATRIB(J)
 195                 CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 196 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
 196                    CONTINUE
                     END IF
                  ELSE
                     CALL FILEM(14, ATRIB)
                  END IF
 186           CONTINUE
               IF (NNQ(14) .GT. 0) THEN
                  DO 197 L = 1,NNQ(14)
                     CALL RMOVE(1, 14, ATRIB)
                     CALL FILEM(13, ATRIB)
 197              CONTINUE
               END IF
            END IF
         END IF
 185  CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 15'
      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 205 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
 205     CONTINUE
```

```
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 16'

      DO 215 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 519 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 521
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
519         CONTINUE
521          IF ((ATRIB(2) .EQ. 5) .AND. (ATRIB(8) .GE. 5))
THEN
               CALL FILEM(13, ATRIB)
            ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
215   CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 17'
      DO 225 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 6))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 226 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1013 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1014 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1014                       CONTINUE
                           GO TO 1015
                        END IF
1013                 CONTINUE
1015                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 235 J = 1, 64
                        NDX1 = J + 6
```

```
                        AUX(I,J) = ATRIB(J)
235                  CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 236 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
236                     CONTINUE
                     END IF
                  ELSE
                     CALL FILEM(14, ATRIB)
                  END IF
226            CONTINUE
               IF (NNQ(14) .GT. 0) THEN
                  DO 237 L = 1,NNQ(14)
                     CALL RMOVE(1, 14, ATRIB)
                     CALL FILEM(13, ATRIB)
237               CONTINUE
               END IF
            END IF
         END IF
225   CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 18'

      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 245 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
245      CONTINUE
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 19'

      DO 255 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 619 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 621
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
619         CONTINUE
621          IF ((ATRIB(2) .EQ. 4) .AND. (ATRIB(8) .GE. 4))
THEN
               CALL FILEM(13, ATRIB)
            ELSE
```

```
              CALL FILEM(15, ATRIB)
           END IF
        END IF
255   CONTINUE

*       WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 20'

      DO 265 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 5))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 266 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1016 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1017 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1017                       CONTINUE
                           GO TO 1018
                        END IF
1016                 CONTINUE
1018                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 275 J = 1, 64
                        NDX1 = J + 6
                        AUX(I,J) = ATRIB(J)
275                  CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 276 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
276                     CONTINUE
                     END IF
                  ELSE
                     CALL FILEM(14, ATRIB)
                  END IF
266            CONTINUE
               IF (NNQ(14) .GT. 0) THEN
                  DO 277 L = 1,NNQ(14)
                     CALL RMOVE(1, 14, ATRIB)
                     CALL FILEM(13, ATRIB)
```

```
277                 CONTINUE
                 END IF
              END IF
           END IF
265     CONTINUE

*        WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 20'

        IF (NNQ(13) .GT. 0) THEN
           NDX1 = NNQ(13)
           DO 285 I = 1, NDX1
              CALL RMOVE(1,13,ATRIB)
              CALL FILEM(15, ATRIB)
285        CONTINUE
        END IF

*        WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 21'

        DO 295 I = 1, ROWS
           IF (AUX(I,6) .EQ. 1) THEN
              POP = NNQ(15)
              DO 719 P = 1, POP
                 CALL RMOVE(1, 15, ATRIB)
                 IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                    GO TO 721
                 ELSE
                    CALL FILEM(15, ATRIB)
                 END IF
719           CONTINUE
721            IF ((ATRIB(2) .EQ. 3) .AND. (ATRIB(8) .GE. 3))
THEN
                 CALL FILEM(13, ATRIB)
              ELSE
                 CALL FILEM(15, ATRIB)
              END IF
           END IF
295     CONTINUE

*        WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 22'
        DO 305 I = 1, ROWS
           IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 4))
THEN
              IF (NNQ(13) .GT. 0) THEN
                 DO 306 K = 1, NNQ(13)
                    CALL RMOVE(1, 13, ATRIB)
                    IF (ATRIB(61) .GE. AUX(I,3)) THEN
                       DO 1019 A = 1, ROWS
                          IF ((AUX(A,6) .EQ. 1) .AND.
     $                    (AUX(A,7) .EQ. ATRIB(7))) THEN
                             DO 1020 Z = 1, 65
```

```
                          NDX1 = Z + 5
                          AUX(A,NDX1) = 0
1020                   CONTINUE
                       GO TO 1021
                    END IF
1019             CONTINUE
1021             ATRIB(1) = AUX(I,1)
                 ATRIB(2) = AUX(I,2)
                 ATRIB(3) = AUX(I,3)
                 ATRIB(4) = AUX(I,4)
                 ATRIB(5) = AUX(I,5)
                 ATRIB(6) = 1
                 AUX(I,6) = 1
                 DO 315 J = 1, 64
                    NDX1 = J + 6
                    AUX(I,J) = ATRIB(J)
315              CONTINUE
                 CALL FILEM(15, ATRIB)
                 IF (NNQ(14) .GT. 0) THEN
                    DO 316 L = 1,NNQ(14)
                       CALL RMOVE(1, 14, ATRIB)
                       CALL FILEM(13, ATRIB)
316                 CONTINUE
                 END IF
              ELSE
                 CALL FILEM(14, ATRIB)
              END IF
306        CONTINUE
           IF (NNQ(14) .GT. 0) THEN
              DO 317 L = 1,NNQ(14)
                 CALL RMOVE(1, 14, ATRIB)
                 CALL FILEM(13, ATRIB)
317           CONTINUE
           END IF
        END IF
     END IF
305   CONTINUE

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 23'

      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 325 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
325      CONTINUE
      END IF

*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 24'
```

```
      DO 335 I = 1, ROWS
         IF (AUX(I,6) .EQ. 1) THEN
            POP = NNQ(15)
            DO 819 P = 1, POP
               CALL RMOVE(1, 15, ATRIB)
               IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                  GO TO 821
               ELSE
                  CALL FILEM(15, ATRIB)
               END IF
819         CONTINUE
821          IF ((ATRIB(2) .EQ. 2) .AND. (ATRIB(8) .GE. 2))
THEN
               CALL FILEM(13, ATRIB)
             ELSE
               CALL FILEM(15, ATRIB)
            END IF
         END IF
335   CONTINUE
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 25'
      DO 345 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 3))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 346 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1022 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1023 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A.NDX1) = 0
1023                       CONTINUE
                           GO TO 1024
                        END IF
1022                 CONTINUE
1024                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 355 J = 1, 64
                        NDX1 = J + 6
                        AUX(I,J) = ATRIB(J)
355                  CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
```

```
                              DO 356 L = 1,NNQ(14)
                                 CALL RMOVE(1, 14, ATRIB)
                                 CALL FILEM(13, ATRIB)
356                           CONTINUE
                           END IF
                        ELSE
                           CALL FILEM(14, ATRIB)
                        END IF
346                  CONTINUE
                     IF (NNQ(14) .GT. 0) THEN
                        DO 357 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
357                     CONTINUE
                     END IF
                  END IF
               END IF
345         CONTINUE

*            WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 26'

            IF (NNQ(13) .GT. 0) THEN
               NDX1 = NNQ(13)
               DO 365 I = 1, NDX1
                  CALL RMOVE(1,13,ATRIB)
                  CALL FILEM(15, ATRIB)
365            CONTINUE
            END IF

*            WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 27'

            DO 375 I = 1, ROWS
               IF (AUX(I,6) .EQ. 1) THEN
                  POP = NNQ(15)
                  DO 919 P = 1, POP
                     CALL RMOVE(1, 15, ATRIB)
                     IF (ATRIB(7) .EQ. AUX(I,7)) THEN
                        GO TO 921
                     ELSE
                        CALL FILEM(15, ATRIB)
                     END IF
919               CONTINUE
921                IF ((ATRIB(2) .EQ. 1) .AND. (ATRIB(8) .GE. 1))
THEN
                     CALL FILEM(13, ATRIB)
                  ELSE
                     CALL FILEM(15, ATRIB)
                  END IF
               END IF
375         CONTINUE
```

```
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 28'

      DO 395 I = 1, ROWS
         IF (((AUX(I,6)) .EQ. 0) .AND. ((AUX(I,2)) .GE. 2))
THEN
            IF (NNQ(13) .GT. 0) THEN
               DO 396 K = 1, NNQ(13)
                  CALL RMOVE(1, 13, ATRIB)
                  IF (ATRIB(61) .GE. AUX(I,3)) THEN
                     DO 1025 A = 1, ROWS
                        IF ((AUX(A,6) .EQ. 1) .AND.
     $                  (AUX(A,7) .EQ. ATRIB(7))) THEN
                           DO 1026 Z = 1, 65
                              NDX1 = Z + 5
                              AUX(A,NDX1) = 0
1026                       CONTINUE
                           GO TO 1027
                        END IF
1025                 CONTINUE
1027                 ATRIB(1) = AUX(I,1)
                     ATRIB(2) = AUX(I,2)
                     ATRIB(3) = AUX(I,3)
                     ATRIB(4) = AUX(I,4)
                     ATRIB(5) = AUX(I,5)
                     ATRIB(6) = 1
                     AUX(I,6) = 1
                     DO 405 J = 1, 64
                        NDX1 = J + 6
                        AUX(I,J) = ATRIB(J)
405                  CONTINUE
                     CALL FILEM(15, ATRIB)
                     IF (NNQ(14) .GT. 0) THEN
                        DO 406 L = 1,NNQ(14)
                           CALL RMOVE(1, 14, ATRIB)
                           CALL FILEM(13, ATRIB)
406                     CONTINUE
                     END IF
                  ELSE
                     CALL FILEM(14, ATRIB)
                  END IF
396            CONTINUE
               IF (NNQ(14) .GT. 0) THEN
                  DO 407 L = 1,NNQ(14)
                     CALL RMOVE(1, 14, ATRIB)
                     CALL FILEM(13, ATRIB)
407               CONTINUE
               END IF
            END IF
         END IF
395   CONTINUE
```

```
*      WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 29'

      IF (NNQ(13) .GT. 0) THEN
         NDX1 = NNQ(13)
         DO 415 I = 1, NDX1
            CALL RMOVE(1,13,ATRIB)
            CALL FILEM(15, ATRIB)
415      CONTINUE
      END IF

**     WRITE(19, *) NNQ(15), ' PEOPLE AT CHECK 30'

      WRITE(19, *) NNQ(15),' LEFT UPDATE'
      WRITE(19, *) CANCEL,' CLASSES WERE CANCELLED THIS YEAR'
      CANCEL = 0
      CALL SCHDL(31, 2.0, ATRIB)
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS100 class 1 of two per year.
 *
************************************************************
****

      SUBROUTINE SY100A
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ1, NUMQ11, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(2, 365.0, ATRIB)

      SIZE = 45
      NUMQ1 = NNQ(1)
      NUMQ11 = NNQ(11)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ1 + NUMQ11
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ11 .GE. TYPE1) .AND. (NUMQ1 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
```

```
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 11, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 1, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ11 .GT. 0) .AND. (NUMQ11 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ11
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 11, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 1, ATRIB)
               ATRIB(70) = 1
```

```
            ATRIB(65) = TNOW
            DO 80 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(10, ATRIB)
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ1 .GT. 0) .AND. (NUMQ1 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ1
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 11, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(10, ATRIB)
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 1, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(10, ATRIB)
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ11 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 1, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
```

```
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130           CONTINUE
         ELSE IF (NUMQ1 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 11, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END

************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS100 class 2 of two per year.
 *
************************************************************
****

      SUBROUTINE SY100B
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ1, NUMQ11, QUOTA1, QUOTA2

      REAL PERCNT
```

```
      CALL SCHDL(3, 365.0, ATRIB)

      SIZE = 45
      NUMQ1 = NNQ(1)
      NUMQ11 = NNQ(11)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ1 + NUMQ11
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ11 .GE. TYPE1) .AND. (NUMQ1 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 11, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 1, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(10, ATRIB)
                     CNT100 = CNT100 + 1
                     CALL SCHDL(28, 7.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ11 .GT. 0) .AND. (NUMQ11 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ11
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 11, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
```

```
                DO 60 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(10, ATRIB)
                      CNT100 = CNT100 + 1
                      CALL SCHDL(28, 7.0, ATRIB)
                      GO TO 50
                   END IF
60              CONTINUE
50           CONTINUE
             DO 70 I = 1,QUOTA2
                CALL RMOVE(1, 1, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 80 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(10, ATRIB)
                      CNT100 = CNT100 + 1
                      CALL SCHDL(28, 7.0, ATRIB)
                      GO TO 70
                   END IF
80              CONTINUE
70           CONTINUE
          ELSE IF ((NUMQ1 .GT. 0) .AND. (NUMQ1 .LT. TYPE2))
THEN
             QUOTA2 = NUMQ1
             QUOTA1 = SIZE - QUOTA2
             DO 90 I = 1, QUOTA1
                CALL RMOVE(1, 11, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 100 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(10, ATRIB)
                      CNT100 = CNT100 + 1
                      CALL SCHDL(28, 7.0, ATRIB)
                      GO TO 90
                   END IF
100             CONTINUE
90           CONTINUE
             DO 110 I = 1,QUOTA2
                CALL RMOVE(1, 1, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 120 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(10, ATRIB)
```

```
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ11 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 1, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(10, ATRIB)
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
      ELSE IF (NUMQ1 .EQ. 0) THEN
         QUOTA1 = SIZE
         DO 150 I = 1, QUOTA1
            CALL RMOVE(1, 11, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(10, ATRIB)
                  CNT100 = CNT100 + 1
                  CALL SCHDL(28, 7.0, ATRIB)
                  GO TO 150
               END IF
160         CONTINUE
150      CONTINUE
      END IF
   ELSE
      CANCEL = CANCEL + 1
   END IF
   RETURN
   END
```

```
*************************************************************
****
*  Subroutine GRD100 represents the graduation of individuals
*
*  from the numerous acquisition courses taken for APDP certi-
*
*   fication.  This routine updates updates their records to
 *
*  reflect the new course and call subroutine ACQLOG for a new
*
*   evaluated certification.
 *
*************************************************************
****

      SUBROUTINE GRD100
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER NUM
      REAL SCHOOL, TODAY, TIME

      TODAY = TNOW
      SCHOOL = 7
      NUM = NNQ(10)
      DO 10 I = 1, NUM
         CALL RMOVE(1, 10, ATRIB)
         TIME = TODAY - ATRIB(65)
         IF (TIME .GE. SCHOOL) THEN
            DO 20 J = 1, ROWS
               IF (ATRIB(7) .EQ. AUX(J,7)) THEN
                  AUX(J,41) = 100
                  ATRIB(41) = 100
                  AUX(J,70) = 0
                  ATRIB(70) = 0
                  AUX(J,51) = 0
                  ATRIB(51) = 0
                  GO TO 30
               END IF
20          CONTINUE
30          GD100 = GD100 + 1
            IF (ATRIB(52) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(21, ATRIB)
               ELSE
                  CALL FILEM(2, ATRIB)
               END IF
            ELSE IF (ATRIB(53) .EQ. 1) THEN
```

```
            IF (ATRIB(62) .EQ. 1) THEN
               CALL FILEM(31, ATRIB)
            ELSE
               CALL FILEM(3, ATRIB)
            END IF
         ELSE IF (ATRIB(55) .EQ. 1) THEN
            IF (ATRIB(62) .EQ. 1) THEN
               CALL FILEM(25, ATRIB)
            ELSE
               CALL FILEM(22, ATRIB)
            END IF
         ELSE IF (ATRIB(54) .EQ. 1) THEN
            IF (ATRIB(62) .EQ. 1) THEN
               CALL FILEM(41, ATRIB)
            ELSE
               CALL FILEM(4, ATRIB)
            END IF
         ELSE IF (ATRIB(56) .EQ. 1) THEN
            IF (ATRIB(62) .EQ. 1) THEN
               CALL FILEM(31, ATRIB)
            ELSE
               CALL FILEM(3, ATRIB)
            END IF
         ELSE IF (ATRIB(57) .EQ. 1) THEN
            IF (ATRIB(62) .EQ. 1) THEN
               CALL FILEM(31, ATRIB)
            ELSE
               CALL FILEM(3, ATRIB)
            END IF
         END IF
      ELSE
         CALL FILEM(10, ATRIB)
      END IF
10    CONTINUE
      D100 = NUM - NNQ(10)
*      WRITE(19, *) 'DOUBLE CHECK FOR GRADUATE ROUTINE 100'
*      WRITE(19,*) D100,' REMOVED'
      D100 = 0
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS200 class one of five per year.
 *
************************************************************
****
```

```
      SUBROUTINE SY200A
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ2, NUMQ21, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(4, 365.0, ATRIB)

      SIZE = 72
      NUMQ2 = NNQ(2)
      NUMQ21 = NNQ(21)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ2 + NUMQ21
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ21 .GE. TYPE1) .AND. (NUMQ2 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
```

```
          ELSE IF ((NUMQ21 .GT. 0) .AND. (NUMQ21 .LT. TYPE1))
THEN
             QUOTA1 = NUMQ21
             QUOTA2 = SIZE - QUOTA1
             DO 50 I = 1, QUOTA1
                CALL RMOVE(1, 21, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 60 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(20, ATRIB)
                      CNT200 = CNT200 + 1
                      CALL SCHDL(32, 21.0, ATRIB)
                      GO TO 50
                   END IF
60              CONTINUE
50           CONTINUE
             DO 70 I = 1,QUOTA2
                CALL RMOVE(1, 2, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 80 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(20, ATRIB)
                      CNT200 = CNT200 + 1
                      CALL SCHDL(32, 21.0, ATRIB)
                      GO TO 70
                   END IF
80              CONTINUE
70           CONTINUE
          ELSE IF ((NUMQ2 .GT. 0) .AND. (NUMQ2 .LT. TYPE2))
THEN
             QUOTA2 = NUMQ2
             QUOTA1 = SIZE - QUOTA2
             DO 90 I = 1, QUOTA1
                CALL RMOVE(1, 21, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 100 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(20, ATRIB)
                      CNT200 = CNT200 + 1
                      CALL SCHDL(32, 21.0, ATRIB)
                      GO TO 90
                   END IF
100             CONTINUE
90           CONTINUE
```

```
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ21 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130          CONTINUE
         ELSE IF (NUMQ2 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
```

```
      RETURN
      END

************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS200 class two of five per year.
 *
************************************************************
****

      SUBROUTINE SY200B
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ2, NUMQ21, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(5, 365.0, ATRIB)

      SIZE = 72
      NUMQ2 = NNQ(2)
      NUMQ21 = NNQ(21)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ2 + NUMQ21
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ21 .GE. TYPE1) .AND. (NUMQ2 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
```

```
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ21 .GT. 0) .AND. (NUMQ21 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ21
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 70
                  END IF
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ2 .GT. 0) .AND. (NUMQ2 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ2
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
```

```
            CALL RMOVE(1, 21, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 2, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ21 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 2, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
      ELSE IF (NUMQ2 .EQ. 0) THEN
         QUOTA1 = SIZE
         DO 150 I = 1, QUOTA1
            CALL RMOVE(1, 21, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
```

```
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 150
                  END IF
160             CONTINUE
150           CONTINUE
            END IF
          ELSE
            CANCEL = CANCEL + 1

          END IF
          RETURN
          END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS200 class three of five per year.
 *
************************************************************
****

      SUBROUTINE SY200C
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ2, NUMQ21, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(6, 365.0, ATRIB)

      SIZE = 72
      NUMQ2 = NNQ(2)
      NUMQ21 = NNQ(21)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ2 + NUMQ21
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ21 .GE. TYPE1) .AND. (NUMQ2 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
```

```
          DO 10 I = 1, QUOTA1
             CALL RMOVE(1, 21, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 20 J = 1, ROWS
                IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(20, ATRIB)
                   CNT200 = CNT200 + 1
                   CALL SCHDL(32, 21.0, ATRIB)
                   GO TO 10
                END IF
20           CONTINUE
10        CONTINUE
          DO 30 I = 1,QUOTA2
             CALL RMOVE(1, 2, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 40 J = 1, ROWS
                IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(20, ATRIB)
                   CNT200 = CNT200 + 1
                   CALL SCHDL(32, 21.0, ATRIB)
                   GO TO 30
                END IF
40           CONTINUE
30        CONTINUE
       ELSE IF ((NUMQ21 .GT. 0) .AND. (NUMQ21 .LT. TYPE1))
THEN
          QUOTA1 = NUMQ21
          QUOTA2 = SIZE - QUOTA1
          DO 50 I = 1, QUOTA1
             CALL RMOVE(1, 21, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 60 J = 1, ROWS
                IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(20, ATRIB)
                   CNT200 = CNT200 + 1
                   CALL SCHDL(32, 21.0, ATRIB)
                   GO TO 50
                END IF
60           CONTINUE
50        CONTINUE
          DO 70 I = 1,QUOTA2
             CALL RMOVE(1, 2, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
```

```
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 70
                  END IF
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ2 .GT. 0) .AND. (NUMQ2 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ2
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ21 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
```

```
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130         CONTINUE
        ELSE IF (NUMQ2 .EQ. 0) THEN
           QUOTA1 = SIZE
           DO 150 I = 1, QUOTA1
              CALL RMOVE(1, 21, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 160 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 150
                 END IF
160           CONTINUE
150        CONTINUE
        END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS200 class four of five per year.
 *
************************************************************
****

      SUBROUTINE SY200D
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ2, NUMQ21, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(7, 365.0, ATRIB)
```

```
      SIZE = 72
      NUMQ2 = NNQ(2)
      NUMQ21 = NNQ(21)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ2 + NUMQ21
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ21 .GE. TYPE1) .AND. (NUMQ2 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ21 .GT. 0) .AND. (NUMQ21 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ21
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
```

```
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(20, ATRIB)
                       CNT200 = CNT200 + 1
                       CALL SCHDL(32, 21.0, ATRIB)
                       GO TO 50
                    END IF
60               CONTINUE
50            CONTINUE
              DO 70 I = 1,QUOTA2
                 CALL RMOVE(1, 2, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 80 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(20, ATRIB)
                       CNT200 = CNT200 + 1
                       CALL SCHDL(32, 21.0, ATRIB)
                       GO TO 70
                    END IF
80               CONTINUE
70            CONTINUE
           ELSE IF ((NUMQ2 .GT. 0) .AND. (NUMQ2 .LT. TYPE2))
THEN
              QUOTA2 = NUMQ2
              QUOTA1 = SIZE - QUOTA2
              DO 90 I = 1, QUOTA1
                 CALL RMOVE(1, 21, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 100 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(20, ATRIB)
                       CNT200 = CNT200 + 1
                       CALL SCHDL(32, 21.0, ATRIB)
                       GO TO 90
                    END IF
100              CONTINUE
90            CONTINUE
              DO 110 I = 1,QUOTA2
                 CALL RMOVE(1, 2, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 120 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(20, ATRIB)
                       CNT200 = CNT200 + 1
```

```
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 110
                 END IF
120            CONTINUE
110         CONTINUE
        ELSE IF (NUMQ21 .EQ. 0) THEN
           QUOTA2 = SIZE
            DO 130 I = 1,QUOTA2
              CALL RMOVE(1, 2, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 140 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130         CONTINUE
        ELSE IF (NUMQ2 .EQ. 0) THEN
           QUOTA1 = SIZE
           DO 150 I = 1, QUOTA1
              CALL RMOVE(1, 21, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 160 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 150
                 END IF
160           CONTINUE
150        CONTINUE
        END IF
     ELSE
        CANCEL = CANCEL + 1
     END IF
     RETURN
     END

************************************************************
****
*  The following routine is used to schedule training for
class*
```

```
*  SYS200 class five of five per year.
 *
************************************************************
****

      SUBROUTINE SY200E
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ2, NUMQ21, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(8, 365.0, ATRIB)

      SIZE = 72
      NUMQ2 = NNQ(2)
      NUMQ21 = NNQ(21)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ2 + NUMQ21
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ21 .GE. TYPE1) .AND. (NUMQ2 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 21, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
                     CALL SCHDL(32, 21.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 2, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(20, ATRIB)
                     CNT200 = CNT200 + 1
```

```
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 30
                 END IF
40            CONTINUE
30         CONTINUE
        ELSE IF ((NUMQ21 .GT. 0) .AND. (NUMQ21 .LT. TYPE1))
THEN
           QUOTA1 = NUMQ21
           QUOTA2 = SIZE - QUOTA1
           DO 50 I = 1, QUOTA1
              CALL RMOVE(1, 21, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 60 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 50
                 END IF
60            CONTINUE
50         CONTINUE
           DO 70 I = 1,QUOTA2
              CALL RMOVE(1, 2, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 80 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
                    CALL SCHDL(32, 21.0, ATRIB)
                    GO TO 70
                 END IF
80            CONTINUE
70         CONTINUE
        ELSE IF ((NUMQ2 .GT. 0) .AND. (NUMQ2 .LT. TYPE2))
THEN
           QUOTA2 = NUMQ2
           QUOTA1 = SIZE - QUOTA2
           DO 90 I = 1, QUOTA1
              CALL RMOVE(1, 21, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 100 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(20, ATRIB)
                    CNT200 = CNT200 + 1
```

```
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 2, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ21 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 2, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
      ELSE IF (NUMQ2 .EQ. 0) THEN
         QUOTA1 = SIZE
         DO 150 I = 1, QUOTA1
            CALL RMOVE(1, 21, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(20, ATRIB)
                  CNT200 = CNT200 + 1
                  CALL SCHDL(32, 21.0, ATRIB)
                  GO TO 150
               END IF
160         CONTINUE
```

```
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  Subroutine GRD200 represents the graduation of individuals
*
*  from the numerous acquisition courses taken for APDP certi-
*
*   fication.  This routine updates updates their records to
 *
*  reflect the new course and call subroutine ACQLOG for a new
*
*   evaluated certification.
 *
************************************************************
****

      SUBROUTINE GRD200
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      REAL NUM, SCHOOL, TODAY, TIME

      TODAY = TNOW
      SCHOOL = 21
      NUM = NNQ(20)
      DO 10 I = 1, NUM
         CALL RMOVE(1, 20, ATRIB)
         TIME = TODAY - ATRIB(65)
         IF (TIME .GE. SCHOOL) THEN
            DO 20 J = 1, ROWS
               IF (ATRIB(7) .EQ. AUX(J,7)) THEN
                  AUX(J,42) = 102
                  ATRIB(42) = 102
                  AUX(J,70) = 0
                  ATRIB(70) = 0
                  AUX(J,52) = 0
                  ATRIB(52) = 0
                  GO TO 30
               END IF
20          CONTINUE
30          GD200 = GD200 + 1
            IF (ATRIB(53) .EQ. 1) THEN
```

```
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            ELSE IF (ATRIB(55) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(25, ATRIB)
               ELSE
                  CALL FILEM(22, ATRIB)
               END IF
            ELSE IF (ATRIB(54) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(41, ATRIB)
               ELSE
                  CALL FILEM(4, ATRIB)
               END IF
            ELSE IF (ATRIB(56) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            END IF
         ELSE
            CALL FILEM(20, ATRIB)
         END IF
10    CONTINUE
      D200 = NUM - NNQ(20)
*      WRITE(19, *) 'DOUBLE CHECK FOR GRADUATE ROUTINE 200'
*      WRITE(19,*) D200,' REMOVED'
      D200 = 0
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class one of six per year.
 *
************************************************************
****
```

```
      SUBROUTINE SY225A
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(9, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
THEN
```

```
          QUOTA1 = NUMQ25
          QUOTA2 = SIZE - QUOTA1
          DO 50 I = 1, QUOTA1
             CALL RMOVE(1, 25, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 60 J = 1, ROWS
                IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(12, ATRIB)
                   CALL SCHDL(33, 14.0, ATRIB)
                   GO TO 50
                END IF
60           CONTINUE
50        CONTINUE
          DO 70 I = 1,QUOTA2
             CALL RMOVE(1, 22, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 80 J = 1, ROWS
                IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(12, ATRIB)
                   CALL SCHDL(33, 14.0, ATRIB)
                   GO TO 70
                END IF
80           CONTINUE
70        CONTINUE
       ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
          QUOTA2 = NUMQ22
          QUOTA1 = SIZE - QUOTA2
          DO 90 I = 1, QUOTA1
             CALL RMOVE(1, 25, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 100 J = 1, ROWS
                IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                   AUX(J,70) = 1
                   CALL FILEM(12, ATRIB)
                   CALL SCHDL(33, 14.0, ATRIB)
                   GO TO 90
                END IF
100          CONTINUE
90        CONTINUE
          DO 110 I = 1,QUOTA2
             CALL RMOVE(1, 22, ATRIB)
             ATRIB(70) = 1
             ATRIB(65) = TNOW
             DO 120 J = 1, ROWS
```

```
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(12, ATRIB)
                       CALL SCHDL(33, 14.0, ATRIB)
                       GO TO 110
                    END IF
120               CONTINUE
110            CONTINUE
            ELSE IF (NUMQ25 .EQ. 0) THEN
               QUOTA2 = SIZE
                DO 130 I = 1,QUOTA2
                 CALL RMOVE(1, 22, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 140 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(12, ATRIB)
                       CALL SCHDL(33, 14.0, ATRIB)
                       GO TO 130
                    END IF
140               CONTINUE
130            CONTINUE
            ELSE IF (NUMQ22 .EQ. 0) THEN
               QUOTA1 = SIZE
               DO 150 I = 1, QUOTA1
                  CALL RMOVE(1, 25, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 160 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 150
                     END IF
160               CONTINUE
150            CONTINUE
            END IF
         ELSE
            CANCEL = CANCEL + 1
         END IF
         RETURN
         END
```

```
***********************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class two of six per year.
 *
***********************************************************
****

      SUBROUTINE SY225B
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(10, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
```

```
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 30
               END IF
40          CONTINUE
30       CONTINUE
      ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
THEN
         QUOTA1 = NUMQ25
         QUOTA2 = SIZE - QUOTA1
         DO 50 I = 1, QUOTA1
            CALL RMOVE(1, 25, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 60 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 50
               END IF
60          CONTINUE
50       CONTINUE
         DO 70 I = 1,QUOTA2
            CALL RMOVE(1, 22, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 80 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ22
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 25, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
```

```
                    GO TO 90
                 END IF
100           CONTINUE
90         CONTINUE
           DO 110 I = 1,QUOTA2
              CALL RMOVE(1, 22, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 120 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 110
                 END IF
120            CONTINUE
110         CONTINUE
        ELSE IF (NUMQ25 .EQ. 0) THEN
           QUOTA2 = SIZE
            DO 130 I = 1,QUOTA2
              CALL RMOVE(1, 22, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 140 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130         CONTINUE
        ELSE IF (NUMQ22 .EQ. 0) THEN
           QUOTA1 = SIZE
           DO 150 I = 1, QUOTA1
              CALL RMOVE(1, 25, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 160 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 150
                 END IF
160           CONTINUE
150        CONTINUE
        END IF
     ELSE
        CANCEL = CANCEL + 1
```

```
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class three of six per year.
 *
************************************************************
****

      SUBROUTINE SY225C
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(11, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
```

```
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ25
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 70
                  END IF
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ22
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
```

```
                  ATRIB(65) = TNOW
                  DO 100 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 90
                     END IF
100               CONTINUE
90             CONTINUE
               DO 110 I = 1,QUOTA2
                  CALL RMOVE(1, 22, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 120 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 110
                     END IF
120                CONTINUE
110             CONTINUE
            ELSE IF (NUMQ25 .EQ. 0) THEN
               QUOTA2 = SIZE
                DO 130 I = 1,QUOTA2
                  CALL RMOVE(1, 22, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 140 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 130
                     END IF
140                CONTINUE
130             CONTINUE
            ELSE IF (NUMQ22 .EQ. 0) THEN
               QUOTA1 = SIZE
               DO 150 I = 1, QUOTA1
                  CALL RMOVE(1, 25, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 160 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 150
```

```
                  END IF
160             CONTINUE
150          CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class four of six per year.
 *
************************************************************
****

      SUBROUTINE SY225D
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(12, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
```

```
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 10
                 END IF
20            CONTINUE
10         CONTINUE
           DO 30 I = 1,QUOTA2
              CALL RMOVE(1, 22, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 40 J = 1, ROWS
                 IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 30
                 END IF
40            CONTINUE
30         CONTINUE
        ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
THEN
           QUOTA1 = NUMQ25
           QUOTA2 = SIZE - QUOTA1
           DO 50 I = 1, QUOTA1
              CALL RMOVE(1, 25, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 60 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 50
                 END IF
60            CONTINUE
50         CONTINUE
           DO 70 I = 1,QUOTA2
              CALL RMOVE(1, 22, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 80 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(12, ATRIB)
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 70
                 END IF
80            CONTINUE
70         CONTINUE
```

```
         ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ22
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ25 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130          CONTINUE
         ELSE IF (NUMQ22 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
```

```
                ATRIB(65) = TNOW
                DO 160 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(12, ATRIB)
                      CALL SCHDL(33, 14.0, ATRIB)
                      GO TO 150
                   END IF
160             CONTINUE
150          CONTINUE
          END IF
      ELSE
          CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class five of six per year.
 *
************************************************************
****

      SUBROUTINE SY225E
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(13, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
```

```
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ25
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 22, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
```

```
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ22
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 25, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 22, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ25 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 22, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(12, ATRIB)
                  CALL SCHDL(33, 14.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
```

```
         ELSE IF (NUMQ22 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END

************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS225 class six of six per year.
 *
************************************************************
****

      SUBROUTINE SY225F
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ22, NUMQ25, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(14, 365.0, ATRIB)

      SIZE = 45
      NUMQ22 = NNQ(22)
      NUMQ25 = NNQ(25)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ22 + NUMQ25
      IF (CHECK .GE. SIZE) THEN
```

```
          IF ((NUMQ25 .GE. TYPE1) .AND. (NUMQ22 .GE. TYPE2))
     THEN
             QUOTA1 = TYPE1
             QUOTA2 = TYPE2
             DO 10 I = 1, QUOTA1
                CALL RMOVE(1, 25, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 20 J = 1, ROWS
                   IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(12, ATRIB)
                      CALL SCHDL(33, 14.0, ATRIB)
                      GO TO 10
                   END IF
20              CONTINUE
10           CONTINUE
             DO 30 I = 1,QUOTA2
                CALL RMOVE(1, 22, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 40 J = 1, ROWS
                   IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(12, ATRIB)
                      CALL SCHDL(33, 14.0, ATRIB)
                      GO TO 30
                   END IF
40              CONTINUE
30           CONTINUE
          ELSE IF ((NUMQ25 .GT. 0) .AND. (NUMQ25 .LT. TYPE1))
     THEN
             QUOTA1 = NUMQ25
             QUOTA2 = SIZE - QUOTA1
             DO 50 I = 1, QUOTA1
                CALL RMOVE(1, 25, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 60 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(12, ATRIB)
                      CALL SCHDL(33, 14.0, ATRIB)
                      GO TO 50
                   END IF
60              CONTINUE
50           CONTINUE
             DO 70 I = 1,QUOTA2
                CALL RMOVE(1, 22, ATRIB)
                ATRIB(70) = 1
```

```
                  ATRIB(65) = TNOW
                  DO 80 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 70
                     END IF
80                CONTINUE
70             CONTINUE
            ELSE IF ((NUMQ22 .GT. 0) .AND. (NUMQ22 .LT. TYPE2))
THEN
               QUOTA2 = NUMQ22
               QUOTA1 = SIZE - QUOTA2
               DO 90 I = 1, QUOTA1
                  CALL RMOVE(1, 25, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 100 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 90
                     END IF
100               CONTINUE
90             CONTINUE
               DO 110 I = 1,QUOTA2
                  CALL RMOVE(1, 22, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 120 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
                        CALL SCHDL(33, 14.0, ATRIB)
                        GO TO 110
                     END IF
120                CONTINUE
110             CONTINUE
            ELSE IF (NUMQ25 .EQ. 0) THEN
               QUOTA2 = SIZE
                DO 130 I = 1,QUOTA2
                  CALL RMOVE(1, 22, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 140 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(12, ATRIB)
```

```
                    CALL SCHDL(33, 14.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130          CONTINUE
         ELSE IF (NUMQ22 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 25, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(12, ATRIB)
                     CALL SCHDL(33, 14.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END

************************************************************
****
*  Subroutine GRD225 represents the graduation of individuals
*
*  from the numerous acquisition courses taken for APDP certi-
*
*  fication.  This routine updates updates their records to
 *
*  reflect the new course and call subroutine ACQLOG for a new
*
*  evaluated certification.
 *
************************************************************
****

      SUBROUTINE GRD225
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      REAL NUM, SCHOOL, TODAY, TIME

      TODAY = TNOW
```

```
      SCHOOL = 14
      NUM = NNQ(12)
      DO 10 I = 1, NUM
         CALL RMOVE(1, 12, ATRIB)
         TIME = TODAY - ATRIB(65)
         IF (TIME .GE. SCHOOL) THEN
            DO 20 J = 1, ROWS
               IF (ATRIB(7) .EQ. AUX(J,7)) THEN
                  AUX(J,45) = 104
                  ATRIB(45) = 104
                  AUX(J,70) = 0
                  ATRIB(70) = 0
                  AUX(J,55) = 0
                  ATRIB(55) = 0
                  GO TO 30
               END IF
20          CONTINUE
30          GD225 = GD225 + 1
            IF (ATRIB(54) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(41, ATRIB)
               ELSE
                  CALL FILEM(4, ATRIB)
               END IF
            ELSE IF (ATRIB(56) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            END IF
         ELSE
            CALL FILEM(12, ATRIB)
         END IF
10    CONTINUE
      RETURN
      END

****************************************************************
****
*  The following routine is used to schedule training for
class*
```

```
*  SYS400 class one of nine per year.
 *
************************************************************
****

      SUBROUTINE SY400A
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(15, 365.0, ATRIB)

      SIZE = 45
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 30
```

```
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ41
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 70
                  END IF
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ4
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
```

```
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ41 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130          CONTINUE
         ELSE IF (NUMQ4 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END
```

```
****************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class two of nine per year.
 *
****************************************************************
****

      SUBROUTINE SY400B
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(16, 365.0, ATRIB)

      SIZE = 45
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
```

```
                    AUX(J,70) = 1
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
                    GO TO 30
                 END IF
40            CONTINUE
30         CONTINUE
        ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
           QUOTA1 = NUMQ41
           QUOTA2 = SIZE - QUOTA1
           DO 50 I = 1, QUOTA1
              CALL RMOVE(1, 41, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 60 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
                    GO TO 50
                 END IF
60            CONTINUE
50         CONTINUE
           DO 70 I = 1,QUOTA2
              CALL RMOVE(1, 4, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 80 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
                    GO TO 70
                 END IF
80            CONTINUE
70         CONTINUE
        ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
           QUOTA2 = NUMQ4
           QUOTA1 = SIZE - QUOTA2
           DO 90 I = 1, QUOTA1
              CALL RMOVE(1, 41, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 100 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
```

```
                    GO TO 90
                 END IF
100           CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ41 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130          CONTINUE
         ELSE IF (NUMQ4 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 150
                  END IF
160           CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
```

```
      END IF
      RETURN
      END

**************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class three of nine per year.
 *
**************************************************************
****

      SUBROUTINE SY400C
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(17, 365.0, ATRIB)

      SIZE = 45

      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
```

```
               DO 30 I = 1,QUOTA2
                  CALL RMOVE(1, 4, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 40 J = 1, ROWS
                     IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 30
                     END IF
40                CONTINUE
30             CONTINUE
            ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
               QUOTA1 = NUMQ41
               QUOTA2 = SIZE - QUOTA1
               DO 50 I = 1, QUOTA1
                  CALL RMOVE(1, 41, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 60 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 50
                     END IF
60                CONTINUE
50             CONTINUE
               DO 70 I = 1,QUOTA2
                  CALL RMOVE(1, 4, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 80 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 70
                     END IF
80                CONTINUE
70             CONTINUE
            ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
               QUOTA2 = NUMQ4
               QUOTA1 = SIZE - QUOTA2
               DO 90 I = 1, QUOTA1
                  CALL RMOVE(1, 41, ATRIB)
                  ATRIB(70) = 1
```

```
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ41 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
      ELSE IF (NUMQ4 .EQ. 0) THEN
         QUOTA1 = SIZE
         DO 150 I = 1, QUOTA1
            CALL RMOVE(1, 41, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 150
```

```
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class four of nine per year.
 *
************************************************************
****

      SUBROUTINE SY400D
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(18, 365.0, ATRIB)

      SIZE = 45
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
```

```
                CALL FILEM(40, ATRIB)
                CALL SCHDL(34, 14.0, ATRIB)
                GO TO 10
             END IF
20        CONTINUE
10     CONTINUE
       DO 30 I = 1,QUOTA2
          CALL RMOVE(1, 4, ATRIB)
          ATRIB(70) = 1
          ATRIB(65) = TNOW
          DO 40 J = 1, ROWS
             IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                AUX(J,70) = 1
                CALL FILEM(40, ATRIB)
                CALL SCHDL(34, 14.0, ATRIB)
                GO TO 30
             END IF
40        CONTINUE
30     CONTINUE
    ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
       QUOTA1 = NUMQ41
       QUOTA2 = SIZE - QUOTA1
       DO 50 I = 1, QUOTA1
          CALL RMOVE(1, 41, ATRIB)
          ATRIB(70) = 1
          ATRIB(65) = TNOW
          DO 60 J = 1, ROWS
             IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                AUX(J,70) = 1
                CALL FILEM(40, ATRIB)
                CALL SCHDL(34, 14.0, ATRIB)
                GO TO 50
             END IF
60        CONTINUE
50     CONTINUE
       DO 70 I = 1,QUOTA2
          CALL RMOVE(1, 4, ATRIB)
          ATRIB(70) = 1
          ATRIB(65) = TNOW
          DO 80 J = 1, ROWS
             IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                AUX(J,70) = 1
                CALL FILEM(40, ATRIB)
                CALL SCHDL(34, 14.0, ATRIB)
                GO TO 70
             END IF
80        CONTINUE
70     CONTINUE
```

```
         ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ4
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110          CONTINUE
         ELSE IF (NUMQ41 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130          CONTINUE
         ELSE IF (NUMQ4 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
```

```
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 150
               END IF
160         CONTINUE
150      CONTINUE
      END IF
   ELSE
      CANCEL = CANCEL + 1
   END IF
   RETURN
   END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class five of nine per year.
 *
************************************************************
****

   SUBROUTINE SY400E
   INCLUDE '/usr/local/Slam/PARAM.INC'
   INCLUDE 'USER.ONE'

   INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

   REAL PERCNT

   CALL SCHDL(19, 365.0, ATRIB)

   SIZE = 45
   NUMQ4 = NNQ(4)
   NUMQ41 = NNQ(41)
   PERCNT = 0.70
   TYPE1 = NINT(PERCNT*SIZE)
   TYPE2 = SIZE - TYPE1
   CHECK = NUMQ4 + NUMQ41
   IF (CHECK .GE. SIZE) THEN
      IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
         QUOTA1 = TYPE1
         QUOTA2 = TYPE2
```

```
                  DO 10 I = 1, QUOTA1
                     CALL RMOVE(1, 41, ATRIB)
                     ATRIB(70) = 1
                     ATRIB(65) = TNOW
                     DO 20 J = 1, ROWS
                        IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                           AUX(J,70) = 1
                           CALL FILEM(40, ATRIB)
                           CALL SCHDL(34, 14.0, ATRIB)
                           GO TO 10
                        END IF
20                   CONTINUE
10                CONTINUE
                  DO 30 I = 1,QUOTA2
                     CALL RMOVE(1, 4, ATRIB)
                     ATRIB(70) = 1
                     ATRIB(65) = TNOW
                     DO 40 J = 1, ROWS
                        IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                           AUX(J,70) = 1
                           CALL FILEM(40, ATRIB)
                           CALL SCHDL(34, 14.0, ATRIB)
                           GO TO 30
                        END IF
40                   CONTINUE
30                CONTINUE
               ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
                  QUOTA1 = NUMQ41
                  QUOTA2 = SIZE - QUOTA1
                  DO 50 I = 1, QUOTA1
                     CALL RMOVE(1, 41, ATRIB)
                     ATRIB(70) = 1
                     ATRIB(65) = TNOW
                     DO 60 J = 1, ROWS
                        IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                           AUX(J,70) = 1
                           CALL FILEM(40, ATRIB)
                           CALL SCHDL(34, 14.0, ATRIB)
                           GO TO 50
                        END IF
60                   CONTINUE
50                CONTINUE
                  DO 70 I = 1,QUOTA2
                     CALL RMOVE(1, 4, ATRIB)
                     ATRIB(70) = 1
                     ATRIB(65) = TNOW
                     DO 80 J = 1, ROWS
                        IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                           AUX(J,70) = 1
```

```
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ4
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 41, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ41 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
```

```
130          CONTINUE
          ELSE IF (NUMQ4 .EQ. 0) THEN
             QUOTA1 = SIZE
             DO 150 I = 1, QUOTA1
                CALL RMOVE(1, 41, ATRIB)
                ATRIB(70) = 1
                ATRIB(65) = TNOW
                DO 160 J = 1, ROWS
                   IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                      AUX(J,70) = 1
                      CALL FILEM(40, ATRIB)
                      CALL SCHDL(34, 14.0, ATRIB)
                      GO TO 150
                   END IF
160             CONTINUE
150          CONTINUE
          END IF
       ELSE
          CANCEL = CANCEL + 1
       END IF
       RETURN
       END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class six of nine per year.
 *
************************************************************
****

       SUBROUTINE SY400F
       INCLUDE '/usr/local/Slam/PARAM.INC'
       INCLUDE 'USER.ONE'

       INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

       REAL PERCNT

       CALL SCHDL(20, 365.0, ATRIB)

       SIZE = 45
       NUMQ4 = NNQ(4)
       NUMQ41 = NNQ(41)
       PERCNT = 0.70
       TYPE1 = NINT(PERCNT*SIZE)
       TYPE2 = SIZE - TYPE1
       CHECK = NUMQ4 + NUMQ41
```

```
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ41
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
```

```
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 80 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ4
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 41, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ41 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
```

```
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130         CONTINUE
        ELSE IF (NUMQ4 .EQ. 0) THEN
           QUOTA1 = SIZE
           DO 150 I = 1, QUOTA1
              CALL RMOVE(1, 41, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 160 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(40, ATRIB)
                    CALL SCHDL(34, 14.0, ATRIB)
                    GO TO 150
                 END IF
160           CONTINUE
150        CONTINUE
        END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class seven of nine per year.
 *
************************************************************
****

      SUBROUTINE SY400G
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(21, 365.0, ATRIB)

      SIZE = 45
```

```
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ41
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
```

```
                  GO TO 50
               END IF
60          CONTINUE
50       CONTINUE
         DO 70 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 80 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 70
               END IF
80          CONTINUE
70       CONTINUE
      ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
         QUOTA2 = NUMQ4
         QUOTA1 = SIZE - QUOTA2
         DO 90 I = 1, QUOTA1
            CALL RMOVE(1, 41, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 100 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 90
               END IF
100         CONTINUE
90       CONTINUE
         DO 110 I = 1,QUOTA2
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 120 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 110
               END IF
120          CONTINUE
110       CONTINUE
      ELSE IF (NUMQ41 .EQ. 0) THEN
         QUOTA2 = SIZE
          DO 130 I = 1,QUOTA2
```

```
            CALL RMOVE(1, 4, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 140 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 130
               END IF
140          CONTINUE
130       CONTINUE
      ELSE IF (NUMQ4 .EQ. 0) THEN
         QUOTA1 = SIZE
         DO 150 I = 1, QUOTA1
            CALL RMOVE(1, 41, ATRIB)
            ATRIB(70) = 1
            ATRIB(65) = TNOW
            DO 160 J = 1, ROWS
               IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                  AUX(J,70) = 1
                  CALL FILEM(40, ATRIB)
                  CALL SCHDL(34, 14.0, ATRIB)
                  GO TO 150
               END IF
160         CONTINUE
150      CONTINUE
      END IF
   ELSE
      CANCEL = CANCEL + 1
   END IF
   RETURN
   END

*************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class eight of nine per year.
 *
*************************************************************
****

   SUBROUTINE SY400H
   INCLUDE '/usr/local/Slam/PARAM.INC'
   INCLUDE 'USER.ONE'

   INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2
```

```
      REAL PERCNT

      CALL SCHDL(22, 365.0, ATRIB)

      SIZE = 45
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
            QUOTA1 = NUMQ41
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
```

```
                  DO 60 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 50
                     END IF
60                CONTINUE
50             CONTINUE
               DO 70 I = 1,QUOTA2
                  CALL RMOVE(1, 4, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 80 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 70
                     END IF
80                CONTINUE
70             CONTINUE
            ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
               QUOTA2 = NUMQ4
               QUOTA1 = SIZE - QUOTA2
               DO 90 I = 1, QUOTA1
                  CALL RMOVE(1, 41, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 100 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 90
                     END IF
100               CONTINUE
90             CONTINUE
               DO 110 I = 1,QUOTA2
                  CALL RMOVE(1, 4, ATRIB)
                  ATRIB(70) = 1
                  ATRIB(65) = TNOW
                  DO 120 J = 1, ROWS
                     IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                        AUX(J,70) = 1
                        CALL FILEM(40, ATRIB)
                        CALL SCHDL(34, 14.0, ATRIB)
                        GO TO 110
                     END IF
```

```
120            CONTINUE
110         CONTINUE
         ELSE IF (NUMQ41 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 130
                  END IF
140            CONTINUE
130         CONTINUE
         ELSE IF (NUMQ4 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
****
*  The following routine is used to schedule training for
class*
*  SYS400 class nine of nine per year.
 *
************************************************************
****
```

```
      SUBROUTINE SY400I
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ4, NUMQ41, QUOTA1, QUOTA2

      REAL PERCNT

      CALL SCHDL(23, 365.0, ATRIB)

      SIZE = 45
      NUMQ4 = NNQ(4)
      NUMQ41 = NNQ(41)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ4 + NUMQ41
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ41 .GE. TYPE1) .AND. (NUMQ4 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 20 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 10
                  END IF
20             CONTINUE
10          CONTINUE
            DO 30 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 40 J = 1, ROWS
                  IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 30
                  END IF
40             CONTINUE
30          CONTINUE
         ELSE IF ((NUMQ41 .GT. 0) .AND. (NUMQ41 .LT. TYPE1))
THEN
```

```
            QUOTA1 = NUMQ41
            QUOTA2 = SIZE - QUOTA1
            DO 50 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 60 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 50
                  END IF
60             CONTINUE
50          CONTINUE
            DO 70 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 80 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 70
                  END IF
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ4 .GT. 0) .AND. (NUMQ4 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ4
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 41, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(40, ATRIB)
                     CALL SCHDL(34, 14.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 4, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
```

```
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(40, ATRIB)
                       CALL SCHDL(34, 14.0, ATRIB)
                       GO TO 110
                    END IF
120               CONTINUE
110            CONTINUE
           ELSE IF (NUMQ41 .EQ. 0) THEN
              QUOTA2 = SIZE
               DO 130 I = 1,QUOTA2
                 CALL RMOVE(1, 4, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 140 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(40, ATRIB)
                       CALL SCHDL(34, 14.0, ATRIB)
                       GO TO 130
                    END IF
140               CONTINUE
130            CONTINUE
           ELSE IF (NUMQ4 .EQ. 0) THEN
              QUOTA1 = SIZE
              DO 150 I = 1, QUOTA1
                 CALL RMOVE(1, 41, ATRIB)
                 ATRIB(70) = 1
                 ATRIB(65) = TNOW
                 DO 160 J = 1, ROWS
                    IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                       AUX(J,70) = 1
                       CALL FILEM(40, ATRIB)
                       CALL SCHDL(34, 14.0, ATRIB)
                       GO TO 150
                    END IF
160              CONTINUE
150           CONTINUE
           END IF
        ELSE
           CANCEL = CANCEL + 1
        END IF
        RETURN
        END


************************************************************
****
```

```
*  Subroutine GRD400 represents the graduation of individuals
*
*  from the numerous acquisition courses taken for APDP certi-
*
*  fication.  This routine updates their records to
 *
*  reflect the new course and call subroutine ACQLOG for a new
*
*  evaluated certification.
 *
************************************************************
****

      SUBROUTINE GRD400
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      REAL NUM, SCHOOL, TODAY, TIME

      TODAY = TNOW
      SCHOOL = 14
      NUM = NNQ(40)
      DO 10 I = 1, NUM
         CALL RMOVE(1, 40, ATRIB)
         TIME = TODAY - ATRIB(65)
         IF (TIME .GE. SCHOOL) THEN
            DO 20 J = 1, ROWS
               IF (ATRIB(7) .EQ. AUX(J,7)) THEN
                  AUX(J,44) = 103
                  ATRIB(44) = 103
                  AUX(J,70) = 0
                  ATRIB(70) = 0
                  AUX(J,54) = 0
                  ATRIB(54) = 0
                  GO TO 30
               END IF
20          CONTINUE
30          GD400 = GD400 + 1
            IF (ATRIB(56) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               IF (ATRIB(62) .EQ. 1) THEN
                  CALL FILEM(31, ATRIB)
               ELSE
                  CALL FILEM(3, ATRIB)
               END IF
```

```
            END IF
         ELSE
            CALL FILEM(40, ATRIB)
         END IF
10    CONTINUE

      RETURN
      END

************************************************************
****
*  The following subroutine, SPLCRS, schedule individual to
 *
*  attend one of the many reccomended acquisition specialty
 *
*  courses for APDP certification.
 *
************************************************************
****

      SUBROUTINE SPLCRS
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      INTEGER SIZE, CHECK, NUMQ3, NUMQ31, NEED, QUOTA1, QUOTA2

      REAL PERCNT

      IF (SPECL .LT. 12) THEN
         CALL SCHDL(24, 28.0, ATRIB)
         SPECL = SPECL + 1
      END IF

      SIZE = 850
      NUMQ3 = NNQ(3)
      NUMQ31 = NNQ(31)
      PERCNT = 0.70
      TYPE1 = NINT(PERCNT*SIZE)
      TYPE2 = SIZE - TYPE1
      CHECK = NUMQ3 + NUMQ31
      IF (CHECK .GE. SIZE) THEN
         IF ((NUMQ31 .GE. TYPE1) .AND. (NUMQ3 .GE. TYPE2))
THEN
            QUOTA1 = TYPE1
            QUOTA2 = TYPE2
            DO 10 I = 1, QUOTA1
               CALL RMOVE(1, 31, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
```

```
      DO 20 J = 1, ROWS
         IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
            AUX(J,70) = 1
            CALL FILEM(30, ATRIB)
            CALL SCHDL(35, 7.0, ATRIB)
            GO TO 10
         END IF
20    CONTINUE
10    CONTINUE
      DO 30 I = 1,QUOTA2
         CALL RMOVE(1, 3, ATRIB)
         ATRIB(70) = 1
         ATRIB(65) = TNOW
         DO 40 J = 1, ROWS
            IF (REAL(AUX(J,7)) .EQ. ATRIB(7)) THEN
               AUX(J,70) = 1
               CALL FILEM(30, ATRIB)
               CALL SCHDL(35, 7.0, ATRIB)
               GO TO 30
            END IF
40       CONTINUE
30    CONTINUE
   ELSE IF ((NUMQ31 .GT. 0) .AND. (NUMQ31 .LT. TYPE1))
THEN
      QUOTA1 = NUMQ31
      QUOTA2 = SIZE - QUOTA1
      DO 50 I = 1, QUOTA1
         CALL RMOVE(1, 31, ATRIB)
         ATRIB(70) = 1
         ATRIB(65) = TNOW
         DO 60 J = 1, ROWS
            IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
               AUX(J,70) = 1
               CALL FILEM(30, ATRIB)
               CALL SCHDL(35, 7.0, ATRIB)
               GO TO 50
            END IF
60       CONTINUE
50    CONTINUE
      DO 70 I = 1,QUOTA2
         CALL RMOVE(1, 3, ATRIB)
         ATRIB(70) = 1
         ATRIB(65) = TNOW
         DO 80 J = 1, ROWS
            IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
               AUX(J,70) = 1
               CALL FILEM(30, ATRIB)
               CALL SCHDL(35, 7.0, ATRIB)
               GO TO 70
            END IF
```

```
80             CONTINUE
70          CONTINUE
         ELSE IF ((NUMQ3 .GT. 0) .AND. (NUMQ3 .LT. TYPE2))
THEN
            QUOTA2 = NUMQ3
            QUOTA1 = SIZE - QUOTA2
            DO 90 I = 1, QUOTA1
               CALL RMOVE(1, 31, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 100 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(30, ATRIB)
                     CALL SCHDL(35, 7.0, ATRIB)
                     GO TO 90
                  END IF
100            CONTINUE
90          CONTINUE
            DO 110 I = 1,QUOTA2
               CALL RMOVE(1, 3, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 120 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(30, ATRIB)
                     CALL SCHDL(35, 14.0, ATRIB)
                     GO TO 110
                  END IF
120             CONTINUE
110         CONTINUE
         ELSE IF (NUMQ31 .EQ. 0) THEN
            QUOTA2 = SIZE
             DO 130 I = 1,QUOTA2
               CALL RMOVE(1, 3, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 140 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(30, ATRIB)
                     CALL SCHDL(35, 7.0, ATRIB)
                     GO TO 130
                  END IF
140             CONTINUE
130         CONTINUE
         ELSE IF (NUMQ3 .EQ. 0) THEN
            QUOTA1 = SIZE
            DO 150 I = 1, QUOTA1
```

```
80            CONTINUE
70         CONTINUE
        ELSE IF ((NUMQ3 .GT. 0) .AND. (NUMQ3 .LT. TYPE2))
THEN
           QUOTA2 = NUMQ3
           QUOTA1 = SIZE - QUOTA2
           DO 90 I = 1, QUOTA1
              CALL RMOVE(1, 31, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 100 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(30, ATRIB)
                    CALL SCHDL(35, 7.0, ATRIB)
                    GO TO 90
                 END IF
100           CONTINUE
90         CONTINUE
           DO 110 I = 1,QUOTA2
              CALL RMOVE(1, 3, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 120 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(30, ATRIB)
                    CALL SCHDL(35, 14.0, ATRIB)
                    GO TO 110
                 END IF
120            CONTINUE
110         CONTINUE
        ELSE IF (NUMQ31 .EQ. 0) THEN
           QUOTA2 = SIZE
            DO 130 I = 1,QUOTA2
              CALL RMOVE(1, 3, ATRIB)
              ATRIB(70) = 1
              ATRIB(65) = TNOW
              DO 140 J = 1, ROWS
                 IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                    AUX(J,70) = 1
                    CALL FILEM(30, ATRIB)
                    CALL SCHDL(35, 7.0, ATRIB)
                    GO TO 130
                 END IF
140            CONTINUE
130         CONTINUE
        ELSE IF (NUMQ3 .EQ. 0) THEN
           QUOTA1 = SIZE
           DO 150 I = 1, QUOTA1
```

```
               CALL RMOVE(1, 31, ATRIB)
               ATRIB(70) = 1
               ATRIB(65) = TNOW
               DO 160 J = 1, ROWS
                  IF ((AUX(J,7)) .EQ. ATRIB(7)) THEN
                     AUX(J,70) = 1
                     CALL FILEM(30, ATRIB)
                     CALL SCHDL(35, 14.0, ATRIB)
                     GO TO 150
                  END IF
160            CONTINUE
150         CONTINUE
         END IF
      ELSE
         CANCEL = CANCEL + 1
      END IF
      RETURN
      END


************************************************************
***
*  Subroutine GRD999 represents the graduation of individuals
*
*  from the numerous specialty  courses taken for APDP certi-
*
*   fication.  This routine updates updates their records to
 *
*  reflect the new course and call subroutine ACQLOG for a new
*
*   evaluated certification.
 *
************************************************************
****

      SUBROUTINE GRD999
      INCLUDE '/usr/local/Slam/PARAM.INC'
      INCLUDE 'USER.ONE'

      REAL NUM, SCHOOL, TODAY, TIME

      TODAY = TNOW
      SCHOOL = 7
      NUM = NNQ(30)
      DO 10 I = 1, NUM
         CALL RMOVE(1, 30, ATRIB)
         TIME = TODAY - ATRIB(65)
         IF (TIME .GE. SCHOOL) THEN
            DO 20 J = 1, ROWS
```

```
         IF ((ATRIB(7)) .EQ. AUX(J,7)) THEN
            IF (ATRIB(43) .EQ. 0) THEN
               AUX(J,43) = 999
               ATRIB(43) = 999
            ELSE IF (ATRIB(46) .EQ. 0) THEN
               AUX(J,46) = 888
               ATRIB(46) = 888
            ELSE IF (ATRIB(47) .EQ. 0) THEN
               AUX(J,47) = 777
               ATRIB(47) = 777
            END IF
            AUX(J,70) = 0
            ATRIB(70) = 0
            IF (ATRIB(53) .EQ. 1) THEN
               AUX(J,53) = 0
               ATRIB(43) = 0
            ELSE IF (ATRIB(56) .EQ. 1) THEN
               AUX(J,56) = 0
               ATRIB(56) = 0
            ELSE IF (ATRIB(57) .EQ. 1) THEN
               AUX(J,57) = 0
               ATRIB(57) = 0
            END IF
         GO TO 30
         END IF
20    CONTINUE
30    GDSPL = GDSPL + 1
      IF (ATRIB(55) .EQ. 1) THEN
         IF (ATRIB(62) .EQ. 1) THEN
            CALL FILEM(25, ATRIB)
         ELSE
            CALL FILEM(22, ATRIB)
         END IF
      ELSE IF (ATRIB(54) .EQ. 1) THEN
         IF (ATRIB(62) .EQ. 1) THEN
            CALL FILEM(41, ATRIB)
         ELSE
            CALL FILEM(4, ATRIB)
         END IF
      ELSE IF (ATRIB(56) .EQ. 1) THEN
         IF (ATRIB(62) .EQ. 1) THEN
            CALL FILEM(31, ATRIB)
         ELSE
            CALL FILEM(3, ATRIB)
         END IF
      ELSE IF (ATRIB(57) .EQ. 1) THEN
         IF (ATRIB(62) .EQ. 1) THEN
            CALL FILEM(31, ATRIB)
         ELSE
            CALL FILEM(3, ATRIB)
```

```
                 END IF
              END IF
           ELSE
              CALL FILEM(30, ATRIB)
           END IF
10      CONTINUE
        RETURN
        END
```

Appendix G

SLAM Simulation Output

1991 TRAINING BASELINE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	165.793	119.839	0.00	507.00	10220.000	72.00
SYS100 DEMAND	210.213	35.283	0.00	286.00	10220.000	286.00
SYS200 DEMAND	224.653	58.292	0.00	392.00	10220.000	286.00
SYS225 DEMAND	323.709	116.276	0.00	676.00	10220.000	360.00
SYS400 DEMAND	317.709	116.439	0.00	668.00	10220.000	357.00
SPECIAL COURSE	1049.171	266.690	0.00	1831.00	10220.000	1175.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		143.660	73.905	286	0	356.101
2		0.866	0.872	4	0	121.301
3		40.451	16.030	86	0	314.621
4		27.017	49.405	172	0	336.319
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.072	0.366	2	0	7.000
11		61.251	72.819	254	0	355.070
12		0.195	0.735	3	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	269	0	0.000
15		67.024	49.467	772	487	0.040
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		0.436	1.246	4	0	21.000
21		13.732	31.002	129	0	286.996
22		1.997	3.695	17	0	190.776
23		0.000	0.000	0	0	0.000

24		0.000	0.000	0	0	0.000
25		1.223	3.747	32	0	142.000
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000
28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.041	1.570	60	0	7.000
31		76.336	23.286	127	0	335.695
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.211	0.610	2	0	14.000
41		9.959	22.810	94	0	317.075
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	28.205	2.712	98	28	188.155

50% TRAINING INCREASE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	169.935	119.655	0.00	507.00	10220.000	78.00
SYS100 DEMAND	155.905	42.102	0.00	270.00	10220.000	143.00
SYS200 DEMAND	166.453	66.503	0.00	392.00	10220.000	142.00
SYS225 DEMAND	303.931	117.796	0.00	676.00	10220.000	284.00
SYS400 DEMAND	293.074	112.634	0.00	668.00	10220.000	278.00
SPECIAL COURSE	994.804	296.660	0.00	1831.00	10220.000	918.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		100.624	33.562	152	0	351.101
2		1.949	2.300	11	0	110.050
3		62.374	23.982	106	0	322.438
4		18.548	45.061	172	0	321.295
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.149	0.748	4	0	7.000
11		48.985	71.013	254	0	348.625
12		0.226	1.135	6	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	264	0	0.000
15		66.799	48.590	772	432	0.042
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		0.690	2.246	8	0	21.000
21		9.255	24.239	129	0	228.476
22		2.257	6.078	34	0	197.120
23		0.000	0.000	0	0	0.000
24		0.000	0.000	0	0	0.000
25		1.649	4.938	26	0	203.096
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000

28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.000	0.000	0	0	0.000
31		101.935	34.108	185	0	338.458
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.259	0.979	4	0	14.000
41		4.550	16.475	94	0	271.959
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	28.575	3.293	47	28	165.365

100% TRAINING INCREASE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	166.043	119.810	0.00	507.00	10220.000	72.00
SYS100 DEMAND	123.837	49.018	0.00	270.00	10220.000	114.00
SYS200 DEMAND	131.599	69.561	0.00	392.00	10220.000	114.00
SYS225 DEMAND	286.076	117.121	0.00	676.00	10220.000	267.00
SYS400 DEMAND	269.612	108.451	0.00	668.00	10220.000	257.00
SPECIAL COURSE	957.274	315.951	0.00	1831.00	10220.000	857.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		78.072	24.325	118	0	335.531
2		3.296	3.835	17	0	100.856
3		78.363	24.539	133	0	304.053
4		13.708	37.537	172	0	306.558
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.303	1.521	8	0	7.000
11		36.245	66.627	254	0	340.149
12		0.237	1.645	12	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	269	0	0.000
15		68.175	46.393	772	388	0.043
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		1.151	4.134	16	0	21.000
21		6.079	17.771	129	0	174.520
22		4.325	8.002	59	0	244.182
23		0.000	0.000	0	0	0.000
24		0.000	0.000	0	0	0.000
25		1.703	6.214	40	0	193.356
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000

28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.000	0.000	0	0	0.000
31		115.711	48.787	213	0	338.068
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.307	1.536	8	0	14.000
41		2.706	12.741	94	0	212.723
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	29.249	5.405	64	28	132.325

150% TRAINING INCREASE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	174.041	117.020	0.00	507.00	10220.000	85.00
SYS100 DEMAND	75.560	63.531	0.00	270.00	10220.000	22.00
SYS200 DEMAND	88.285	79.497	0.00	392.00	10220.000	36.00
SYS225 DEMAND	306.895	104.423	0.00	676.00	10220.000	292.00
SYS400 DEMAND	290.895	90.435	0.00	668.00	10220.000	284.00
SPECIAL COURSE	1018.945	288.657	0.00	1831.00	10220.000	931.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		30.642	17.641	76	0	258.807
2		11.374	9.779	44	0	136.115
3		130.610	39.015	214	0	306.647
4		8.954	28.308	172	0	265.249
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.579	2.986	16	0	7.000
11		29.058	62.295	254	0	328.148
12		0.248	2.366	24	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	265	0	0.000
15		77.056	50.698	772	436	0.045
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		1.907	7.575	32	0	21.000
21		5.233	13.349	129	0	141.489
22		5.333	11.424	91	0	252.343
23		0.000	0.000	0	0	0.000
24		0.000	0.000	0	0	0.000
25		2.017	9.612	79	0	212.484
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000

28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.000	0.000	0	0	0.000
31		132.302	55.305	255	0	337.948
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.329	2.270	16	0	14.000
41		1.506	9.378	94	0	163.787
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	30.314	9.073	84	28	101.443

200% TRAINING INCREASE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	177.006	119.212	0.00	507.00	10220.000	79.00
SYS100 DEMAND	60.845	54.454	0.00	270.00	10220.000	32.00
SYS200 DEMAND	92.605	72.461	0.00	392.00	10220.000	44.00
SYS225 DEMAND	299.678	102.457	0.00	676.00	10220.000	314.00
SYS400 DEMAND	285.107	85.656	0.00	668.00	10220.000	309.00
SPECIAL COURSE	1008.438	282.608	0.00	1831.00	10220.000	1006.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		20.150	16.208	72	0	190.501
2		29.106	21.013	89	0	213.543
3		118.140	48.237	238	0	314.587
4		4.985	20.766	172	0	236.944
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.658	4.540	32	0	7.000
11		18.450	53.040	254	0	303.156
12		0.248	3.352	48	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	263	0	0.000
15		73.282	49.701	772	460	0.046
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		1.823	10.598	64	0	21.000
21		5.262	14.615	129	0	114.917
22		8.269	18.229	140	0	292.426
23		0.000	0.000	0	0	0.000
24		0.000	0.000	0	0	0.000
25		1.558	8.652	79	0	164.175
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000

28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.000	0.000	0	0	0.000
31		147.013	76.829	320	0	345.237
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.351	3.332	32	0	14.000
41		0.673	6.360	94	0	73.170
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	30.330	12.846	131	28	98.622

250% TRAINING INCREASE

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
BACKLOG	0.000	0.000	0.00	0.00	10220.000	0.00
BACKLOG CHECK	160.543	120.686	0.00	507.00	10220.000	66.00
SYS100 DEMAND	66.200	47.470	0.00	270.00	10220.000	52.00
SYS200 DEMAND	128.711	67.378	0.00	392.00	10220.000	59.00
SYS225 DEMAND	296.358	103.318	0.00	676.00	10220.000	299.00
SYS400 DEMAND	278.894	86.792	0.00	668.00	10220.000	293.00
SPECIAL COURSE	981.980	289.041	0.00	1831.00	10220.000	938.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1		33.685	22.782	91	0	247.311
2		56.164	41.977	153	0	292.405
3		85.006	54.261	209	0	325.377
4		4.192	14.235	172	0	177.029
5		0.000	0.000	0	0	0.000
6		0.000	0.000	0	0	0.000
7		0.000	0.000	0	0	0.000
8		0.000	0.000	0	0	0.000
9		0.000	0.000	0	0	0.000
10		0.570	6.012	64	0	7.000
11		12.326	41.489	254	0	273.267
12		0.248	4.739	96	0	14.000
13		0.000	0.000	273	0	0.000
14		0.000	0.000	271	0	0.000
15		77.872	49.036	772	469	0.050
16		0.000	0.000	0	0	0.000
17		0.000	0.000	0	0	0.000
18		0.000	0.000	0	0	0.000
19		0.000	0.000	0	0	0.000
20		1.297	12.731	128	0	21.000
21		10.352	26.211	130	0	166.867
22		9.077	16.273	140	0	278.595
23		0.000	0.000	0	0	0.000
24		0.000	0.000	0	0	0.000
25		1.417	8.278	79	0	149.320
26		0.000	0.000	0	0	0.000
27		0.000	0.000	0	0	0.000

28		0.000	0.000	0	0	0.000
29		0.000	0.000	0	0	0.000
30		0.000	0.000	0	0	0.000
31		132.347	75.729	305	0	345.664
32		0.000	0.000	0	0	0.000
33		0.000	0.000	0	0	0.000
34		0.000	0.000	0	0	0.000
35		0.000	0.000	0	0	0.000
36		0.000	0.000	0	0	0.000
37		0.000	0.000	0	0	0.000
38		0.000	0.000	0	0	0.000
39		0.000	0.000	0	0	0.000
40		0.351	4.724	64	0	14.000
41		0.304	4.082	94	0	33.032
42		0.000	0.000	0	0	0.000
43		0.000	0.000	0	0	0.000
44		0.000	0.000	0	0	0.000
45		0.000	0.000	0	0	0.000
46	CALENDAR	29.716	15.993	177	28	110.075

Appendix H

Course Quotas For Training

The following course quotas were provided by AFIT/LSA and used throughout the analysis of this thesis effort:

SYS100: 30 per course offering (2 per year)

SYS200: 72 per course offering (5 per year)

SYS225: 48 per course offering (7 per year)

SYS400: 30 per course offering (9 per year)

SPECIALTY COURSE: 650 per course offering. (Group emulation once each month)

Appendix I

Simulation Results Using 1991 Baseline Capability

Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	408	247	131	347	187	566	466	558	277	0	1545
1994	366	223	106	299	160	491	389	490	249	0	1392
1995	336	223	108	281	142	462	355	461	236	0	1346
1996	294	207	102	242	120	405	339	413	199	0	1231
1997	262	188	96	205	103	350	276	364	172	0	1127
1998	234	174	88	175	92	309	247	327	171	0	1042
1999	216	166	82	163	71	271	206	291	143	0	980
2000	197	159	68	157	81	241	194	260	136	0	921
2001	174	141	70	138	67	204	157	217	117	0	817
2002	152	128	58	127	61	186	157	187	86	0	743
2003	129	112	56	111	53	164	131	166	82	0	647
2004	112	100	51	100	49	147	123	150	65	0	584
2005	105	102	55	102	53	145	118	148	75	0	570
2006	100	97	48	97	45	143	111	146	72	0	546
2007	87	84	41	84	36	130	100	133	66	0	501
2008	84	77	34	77	35	122	88	125	62	0	469
2009	78	70	40	70	36	112	88	115	48	0	433
2010	74	61	29	61	32	105	87	108	60	0	408
2011	73	59	32	59	37	102	78	105	56	0	399
2012	72	53	29	53	25	99	84	102	45	0	388
2013	71	52	28	52	26	98	75	101	50	0	385
2014	69	52	22	52	34	96	78	99	45	0	379
2015	69	50	27	50	21	93	72	96	48	0	370
2016	69	49	27	49	21	94	71	97	56	0	373
2017	69	49	25	49	22	94	75	97	42	0	373
2018	69	49	20	49	21	94	79	97	55	0	373
2019	69	48	27	48	31	94	80	97	45	0	373

Simulation Results Using 50% Increase

Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	418	243	115	330	180	558	458	563	293	0	1585
1994	375	216	108	265	136	471	375	496	247	0	1431
1995	338	212	103	226	116	417	317	451	230	0	1368
1996	298	185	100	182	95	350	292	386	176	0	1250
1997	267	166	80	170	83	290	232	327	166	0	1139
1998	244	158	82	160	81	267	216	285	146	0	1065
1999	222	137	61	140	80	245	194	249	125	0	968
2000	199	129	65	130	50	230	171	235	126	0	905
2001	177	114	58	115	59	210	167	214	104	0	821
2002	156	98	51	99	50	189	151	192	87	0	736
2003	130	87	41	88	45	171	139	175	80	0	654
2004	115	83	41	85	39	160	133	165	79	0	611
2005	108	76	37	78	37	155	125	160	88	0	578
2006	98	69	34	71	33	148	123	153	79	0	539
2007	92	61	40	63	35	138	112	143	72	0	499
2008	90	56	32	58	27	135	104	140	68	0	480
2009	89	55	25	56	27	133	102	138	64	0	474
2010	83	56	30	56	28	128	91	133	60	0	455
2011	80	55	30	56	27	122	98	127	68	0	437
2012	77	51	23	53	30	116	95	121	58	0	417
2013	77	51	22	55	24	115	94	120	52	0	414
2014	76	53	32	51	15	113	88	118	56	0	408
2015	76	49	18	48	25	111	89	116	51	0	402
2016	76	51	22	50	21	113	84	118	61	0	408
2017	76	53	32	52	28	114	89	119	70	0	411
2018	76	50	26	50	26	112	97	117	53	0	405
2019	76	49	28	49	26	110	92	115	60	0	399

Simulation Results Using 100% Increase

Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	418	237	120	294	160	533	440	567	287	0	1585
1994	370	207	113	212	109	418	333	477	228	0	1434
1995	334	183	101	188	100	331	253	394	193	0	1349
1996	301	159	82	165	83	304	246	325	149	0	1252
1997	261	130	61	133	71	269	217	280	145	0	1116
1998	228	114	48	117	59	250	198	259	119	0	1024
1999	212	98	46	102	53	241	198	249	138	0	971
2000	197	88	46	92	41	229	182	238	126	0	911
2001	172	77	31	78	42	208	159	218	98	0	822
2002	152	65	32	66	35	190	148	199	98	0	749
2003	136	59	33	59	30	176	139	184	99	0	681
2004	124	56	29	56	26	161	124	169	92	0	629
2005	116	54	30	54	15	152	129	160	77	0	591
2006	106	50	23	50	24	144	115	152	78	0	543
2007	92	46	25	46	27	125	105	132	62	0	465
2008	87	48	20	48	28	125	102	132	65	0	457
2009	83	47	16	47	23	120	98	127	72	0	440
2010	77	46	21	46	21	112	85	119	62	0	412
2011	74	51	32	51	28	110	84	117	55	0	404
2012	72	44	19	44	19	106	83	114	59	0	390
2013	71	46	30	46	22	105	73	113	59	0	387
2014	71	44	20	44	18	108	89	116	57	0	396
2015	71	40	20	40	25	105	87	113	51	0	387
2016	71	40	17	40	17	105	86	113	63	0	387
2017	71	45	21	45	19	104	32	112	50	0	384
2018	71	37	16	37	23	108	84	116	58	0	396
2019	71	39	18	39	17	109	85	117	55	0	399

Simulation Results Using 150% Increase

Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	418	226	119	222	124	469	387	574	286	0	1585
1994	377	182	95	208	105	345	275	432	204	0	1450
1995	342	165	84	185	101	340	265	352	178	0	1387
1996	299	123	56	141	67	309	237	321	159	0	1269
1997	269	105	56	118	62	287	242	296	145	0	1161
1998	246	82	34	90	46	272	218	280	128	0	1085
1999	218	57	28	64	32	245	195	252	135	0	957
2000	196	50	26	52	28	238	193	245	127	0	908
2001	175	39	20	40	21	219	170	228	119	0	834
2002	158	39	22	40	20	199	151	206	106	0	754
2003	143	33	15	33	17	185	147	191	100	0	689
2004	132	34	21	35	19	171	133	176	91	0	631
2005	116	31	13	32	15	157	132	162	76	0	575
2006	108	35	20	35	17	146	118	151	69	0	518
2007	103	34	17	34	15	144	119	150	80	0	512
2008	97	37	20	37	20	139	114	145	70	0	495
2009	91	33	15	34	16	133	108	139	65	0	473
2010	86	33	13	34	19	125	97	131	59	0	448
2011	86	30	17	30	21	124	95	130	73	0	443
2012	84	28	16	28	14	119	80	125	73	0	428
2013	84	33	15	41	17	121	102	127	59	0	434
2014	84	27	13	35	13	117	95	123	59	0	422
2015	84	25	9	31	12	122	99	128	70	0	437
2016	84	38	19	39	16	118	92	124	66	0	425
2017	84	23	10	26	9	114	88	120	64	0	413
2018	84	28	13	31	18	117	92	123	57	0	422
2019	84	22	9	26	17	119	92	125	62	0	428

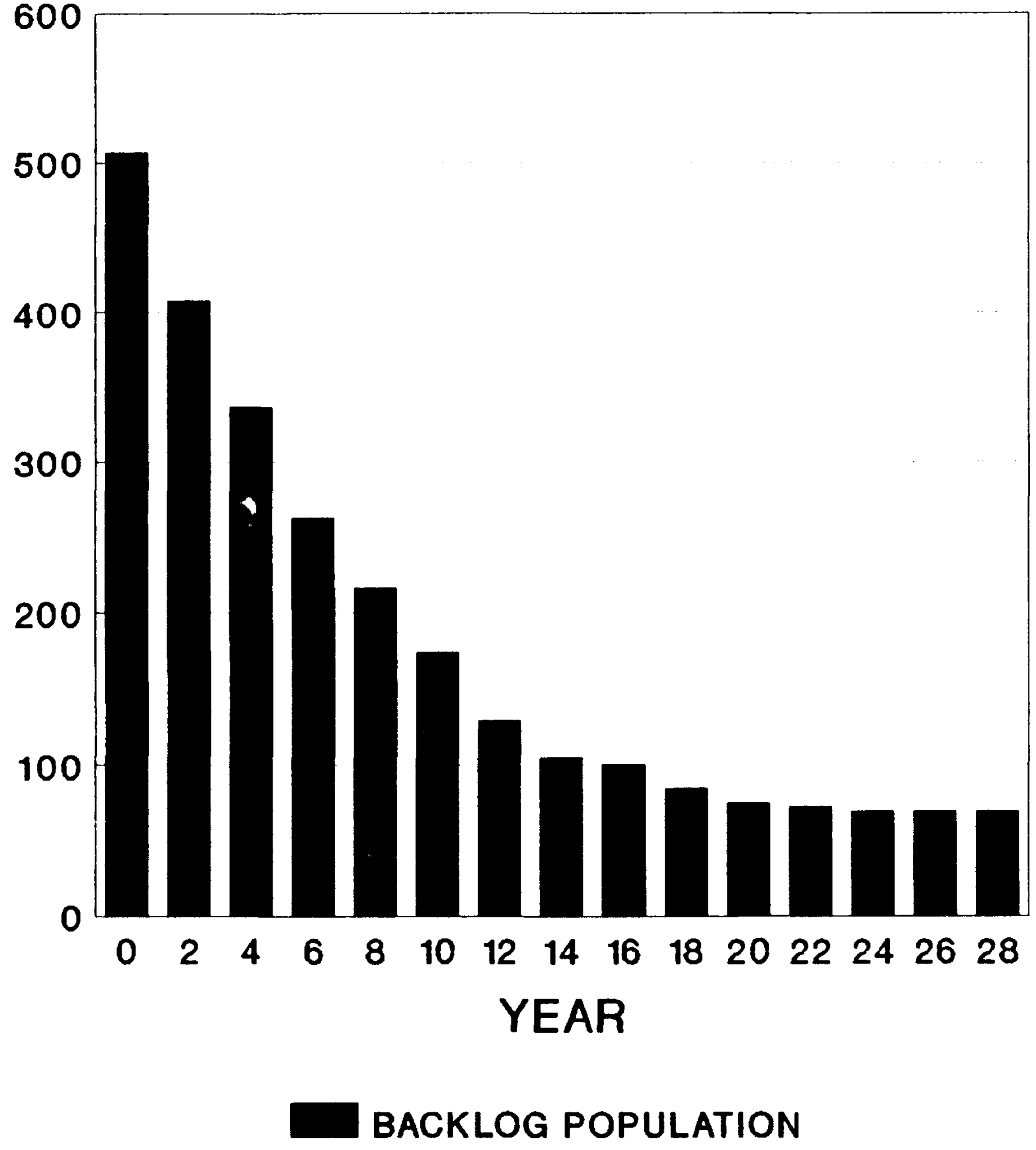
Simulation Results Using 200% Increase

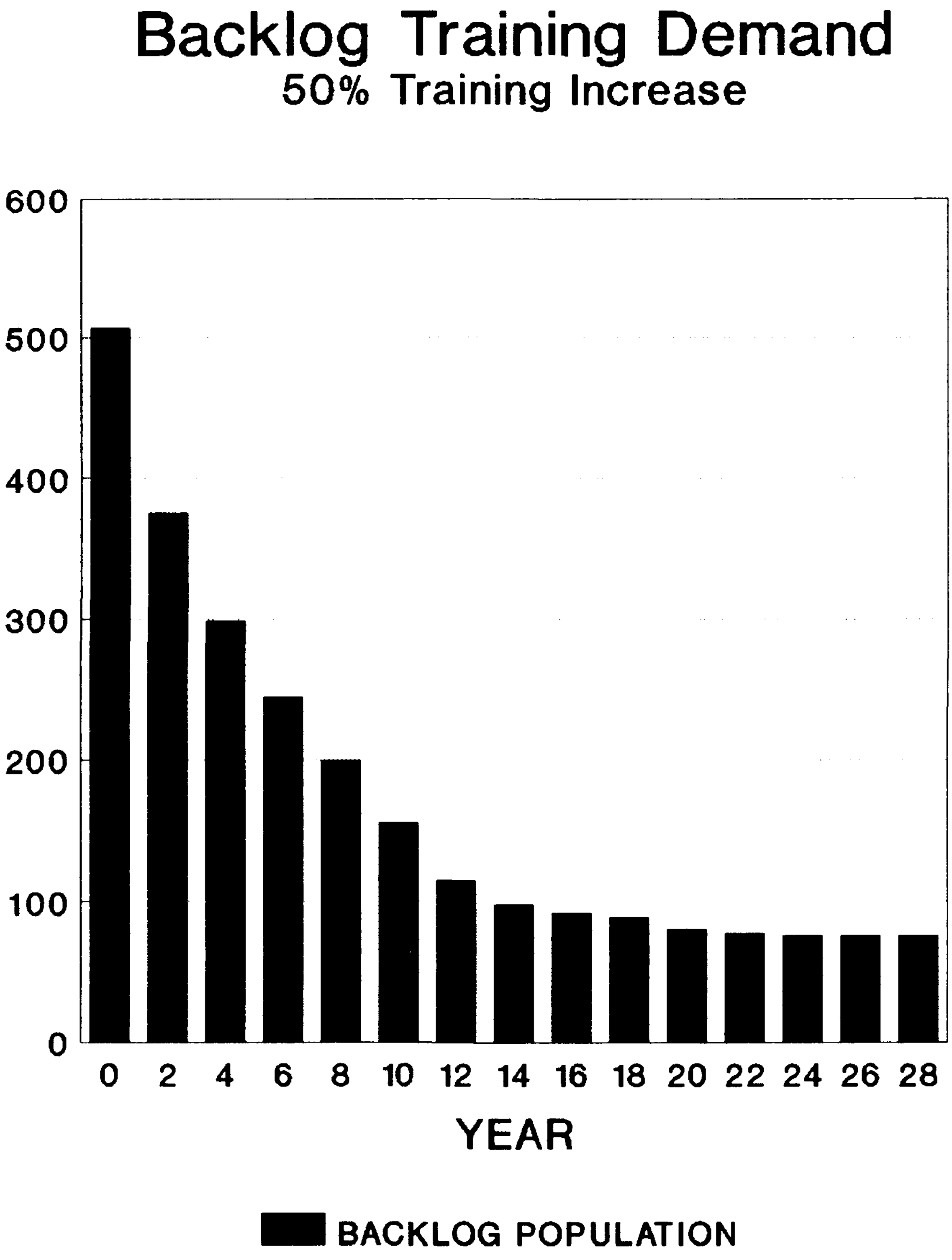
Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	418	209	113	246	132	378	312	574	282	0	1585
1994	375	130	63	174	93	347	280	360	169	0	1433
1995	344	82	41	122	67	337	267	350	188	0	1373
1996	320	41	15	71	36	322	251	333	179	0	1305
1997	276	37	14	38	21	284	228	293	148	0	1129
1998	258	31	13	49	23	275	226	286	138	0	1074
1999	235	35	15	63	27	262	205	272	140	0	999
2000	211	31	16	33	21	242	199	251	116	0	918
2001	185	45	27	45	23	220	169	228	109	0	820
2002	169	41	24	51	25	197	148	206	115	0	738
2003	153	41	17	60	29	186	147	193	90	0	694
2004	140	40	22	44	27	171	135	179	89	0	642
2005	126	36	15	56	28	161	131	170	86	0	598
2006	113	32	18	32	15	151	125	160	89	0	560
2007	105	38	14	45	22	136	111	144	76	0	499
2008	100	35	20	52	25	130	109	138	68	0	475
2009	93	27	11	32	22	123	101	130	62	0	450
2010	88	25	10	43	20	118	92	124	59	0	433
2011	83	43	21	48	21	110	84	116	51	0	407
2012	80	21	16	27	12	108	83	114	68	0	401
2013	79	32	18	36	20	105	70	110	50	0	390
2014	78	25	12	45	23	105	90	110	53	0	390
2015	78	23	11	31	18	105	85	110	58	0	390
2016	78	44	18	46	28	110	86	115	57	0	405
2017	78	28	16	28	9	109	90	114	60	0	402
2018	78	32	19	48	22	111	83	116	62	0	408
2019	78	30	22	34	12	109	92	114	63	0	402

Simulation Results Using 250% Increase

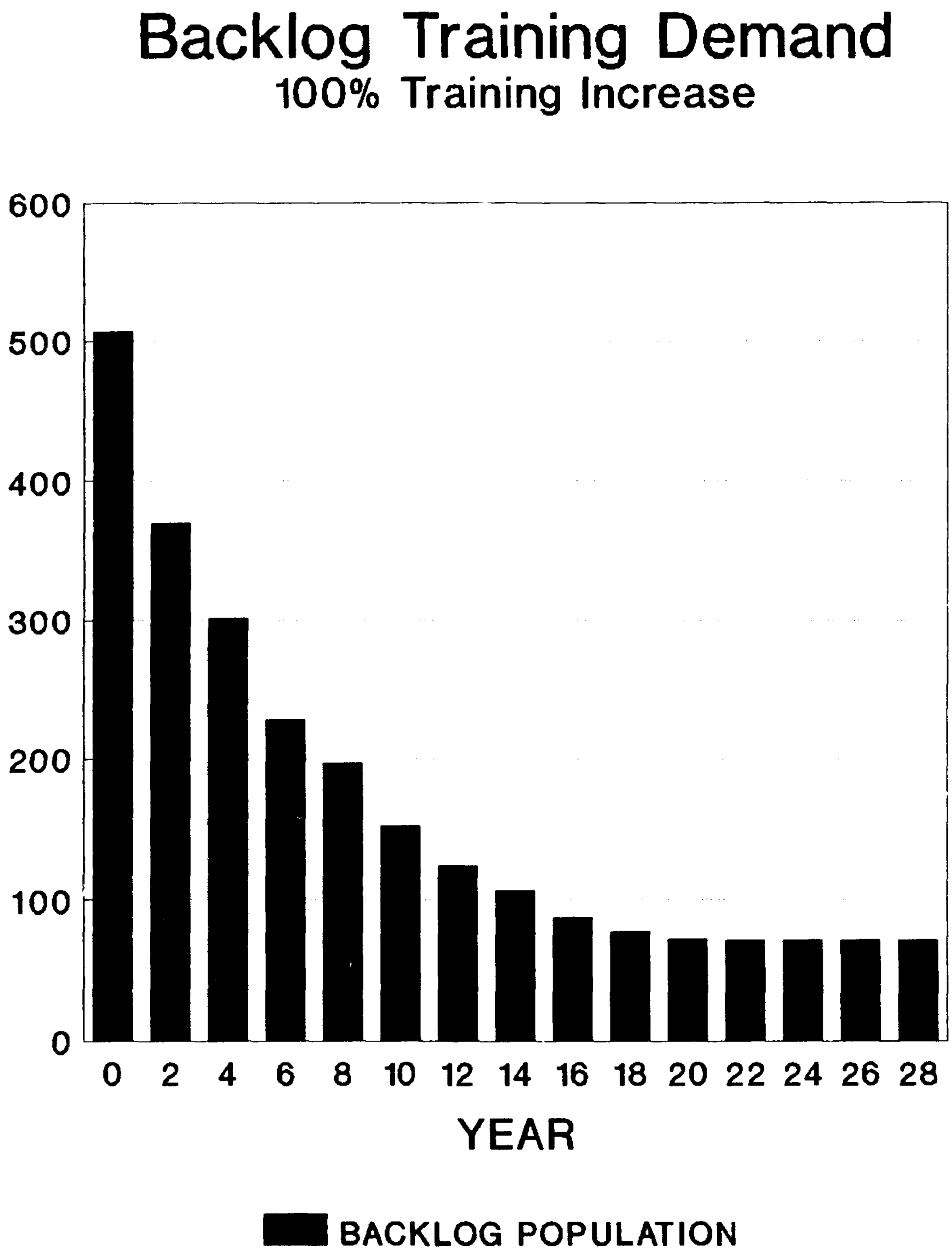
Year	BACK LOG	MAX 100	MIN 100	MAX 200	MIN 200	MAX 225	MIN 225	MAX 400	MIN 400	SPECIAL	
1992	507	270	131	392	198	668	523	676	338	0	1831
1993	418	160	71	246	132	378	312	574	299	0	1585
1994	371	33	14	117	63	341	274	359	171	0	1430
1995	326	56	25	129	67	318	251	334	182	0	1323
1996	295	59	31	128	70	300	236	314	148	0	1234
1997	250	37	18	41	21	269	216	284	166	0	1093
1998	226	47	17	52	24	251	202	265	131	0	1006
1999	205	36	21	57	27	230	184	244	115	0	913
2000	185	47	25	63	34	219	176	232	117	0	860
2001	165	57	22	63	32	203	164	217	108	0	788
2002	149	33	19	68	35	192	148	206	98	0	739
2003	127	48	26	63	31	170	127	183	84	0	645
2004	118	56	28	62	33	159	129	169	80	0	595
2005	107	36	19	63	32	149	117	159	74	0	551
2006	94	50	28	57	32	131	101	141	71	0	483
2007	90	22	13	60	27	130	112	141	64	0	473
2008	86	34	18	60	29	126	102	136	69	0	447
2009	80	44	20	62	34	122	95	132	70	0	435
2010	74	31	19	31	15	111	97	118	63	0	393
2011	69	45	18	47	31	106	88	113	60	0	377
2012	67	34	13	50	19	99	78	105	53	0	354
2013	66	48	24	53	22	98	73	104	46	0	351
2014	65	13	6	53	20	96	73	102	50	0	345
2015	65	39	18	51	24	94	74	100	57	0	339
2016	65	47	22	52	30	95	65	101	50	0	342
2017	65	50	22	51	31	94	80	100	59	0	339
2018	65	19	10	17	6	94	76	100	43	0	339
2019	65	31	12	31	14	93	71	99	49	0	336

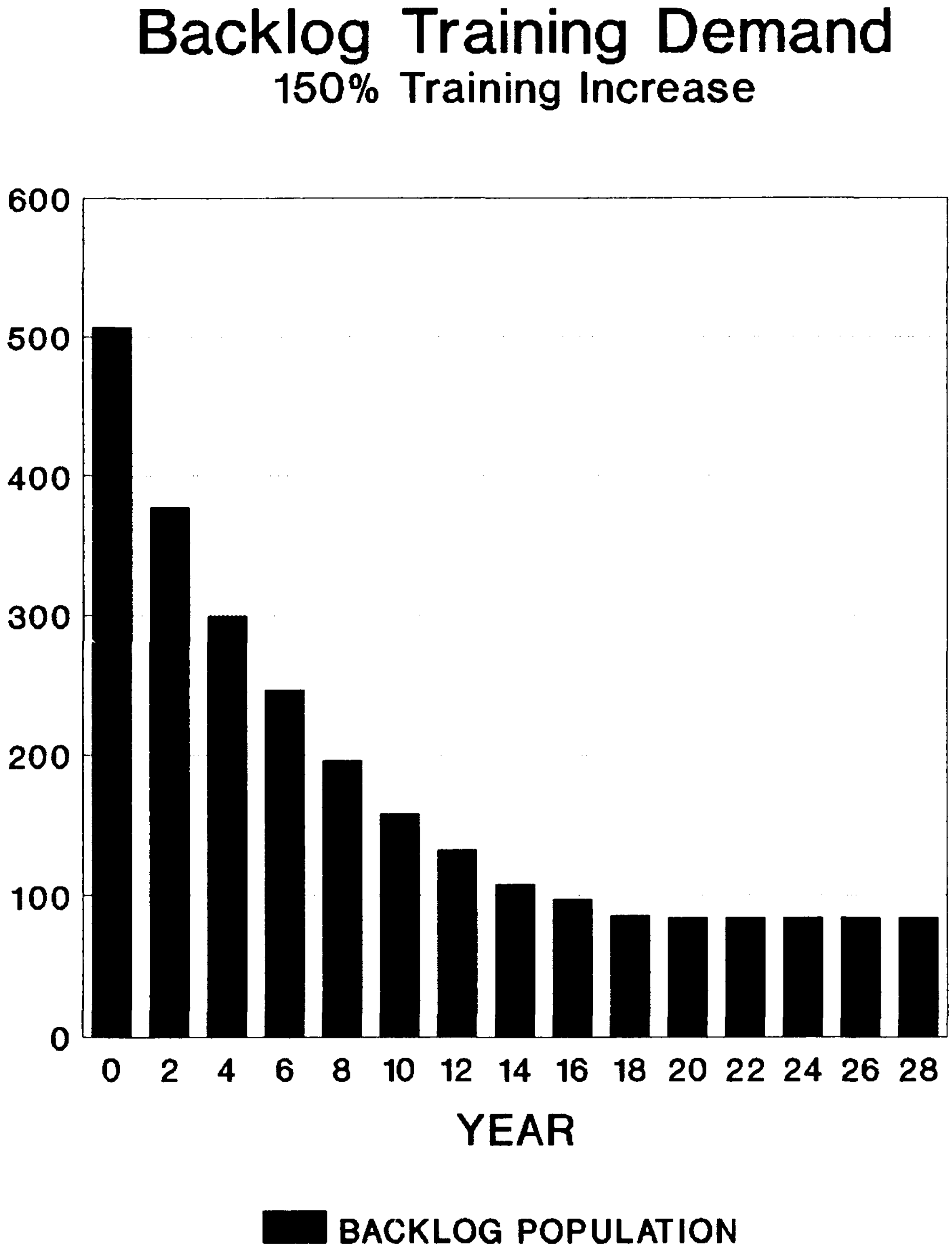
Backlog Training Demand
Based on 1991 Training Capability

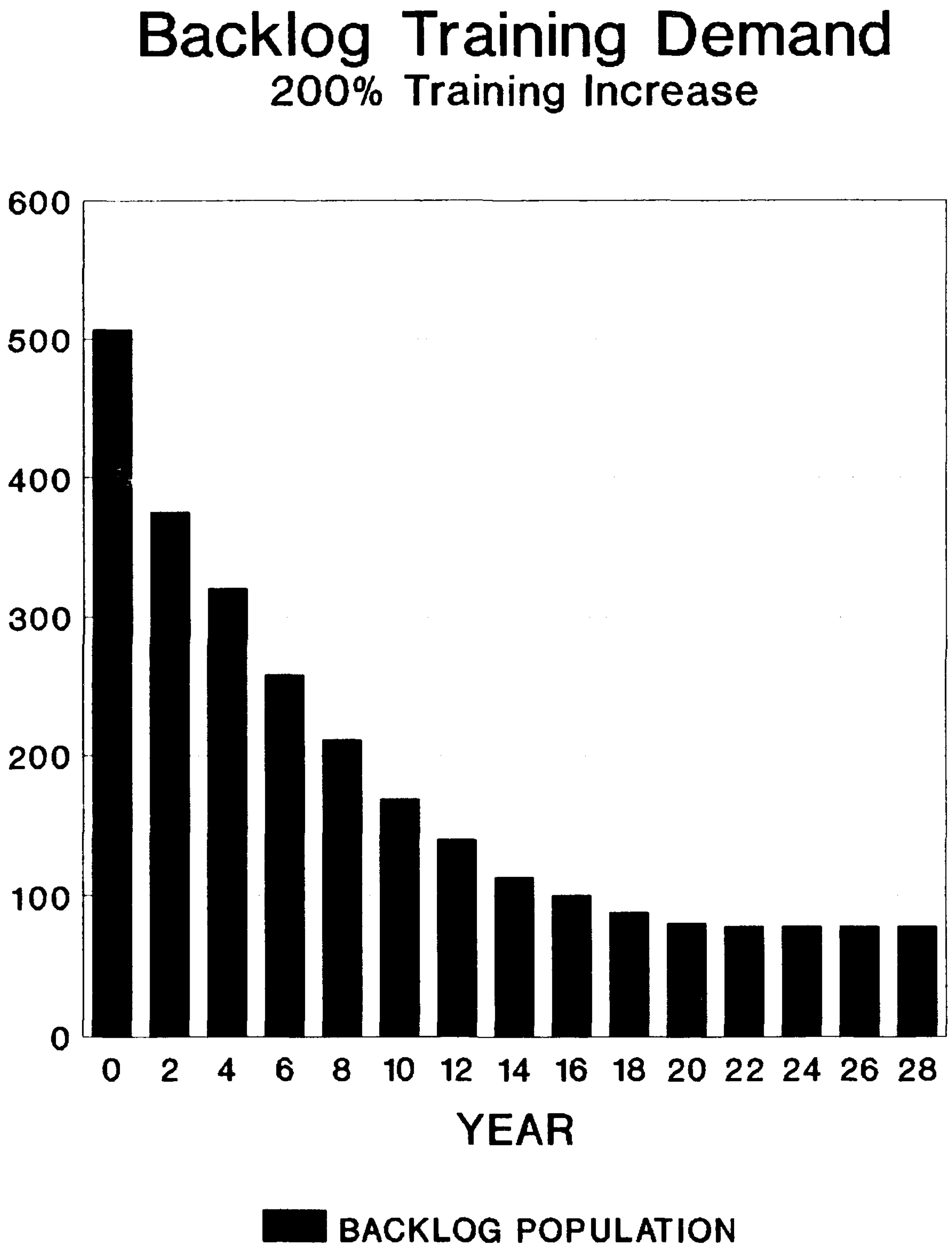




Backlog Training Demand
50% Training Increase
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
BACKLOG POPULATION

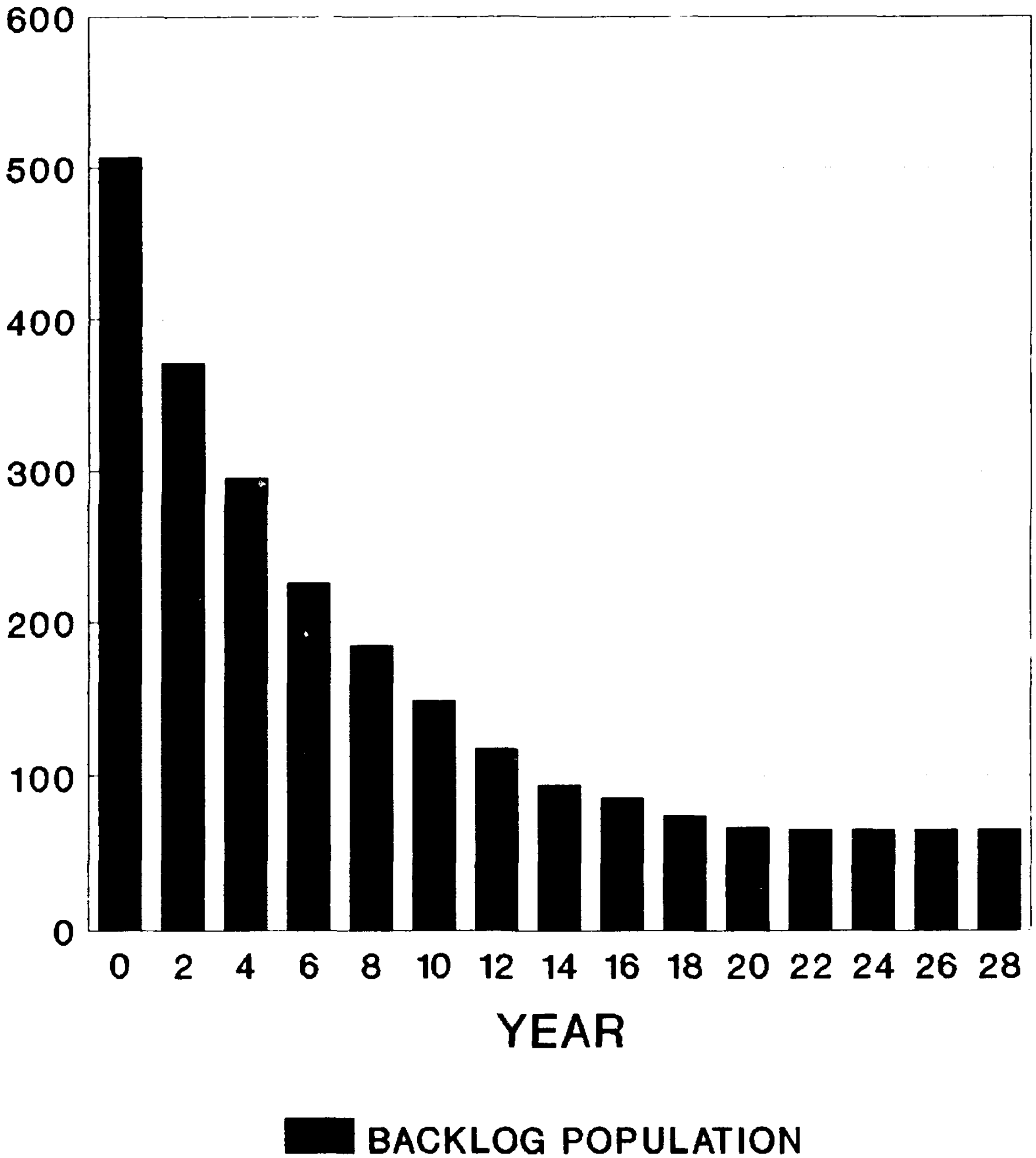

Backlog Training Demand
100% Training Increase
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
BACKLOG POPULATION

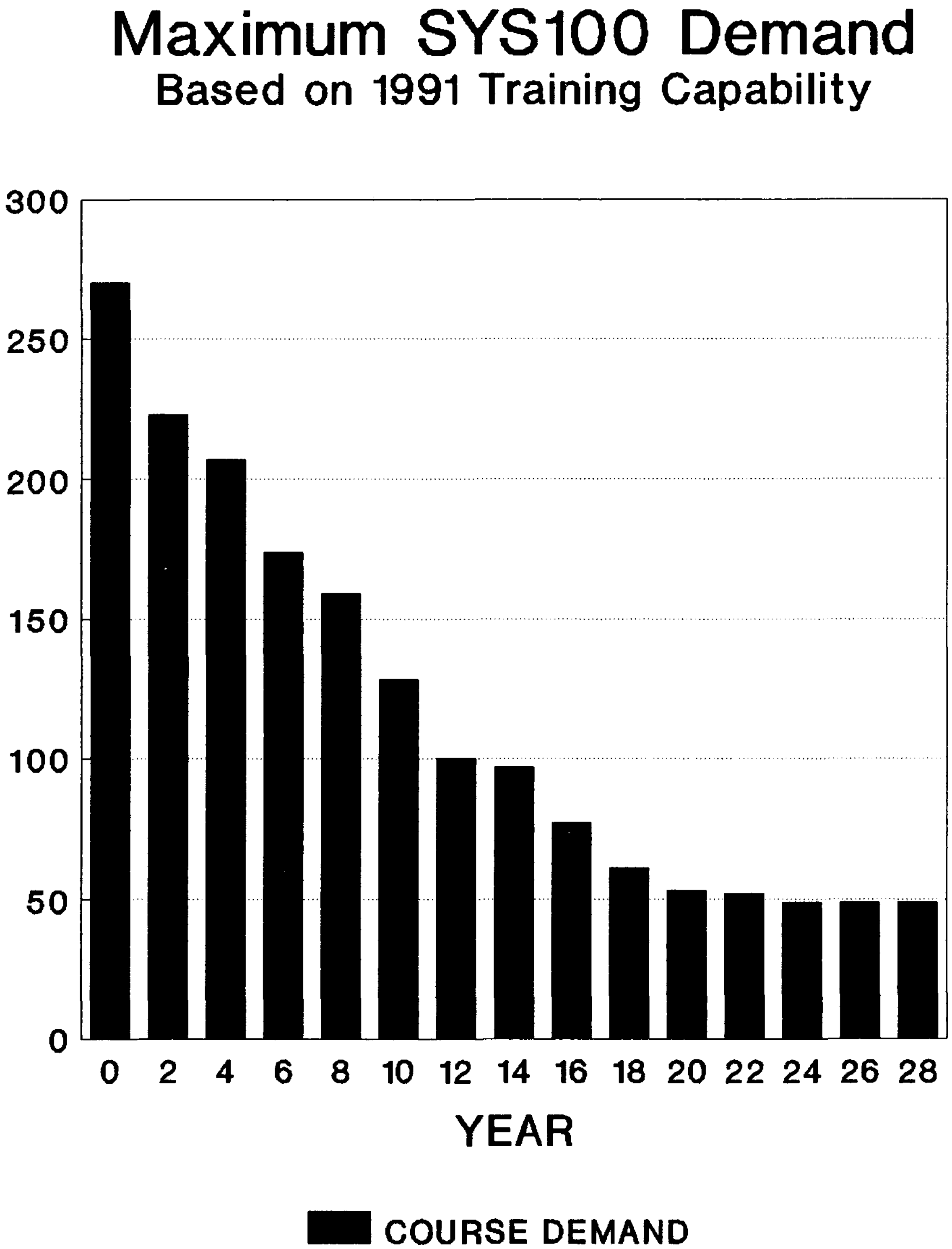

Backlog Training Demand
150% Training Increase
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
BACKLOG POPULATION

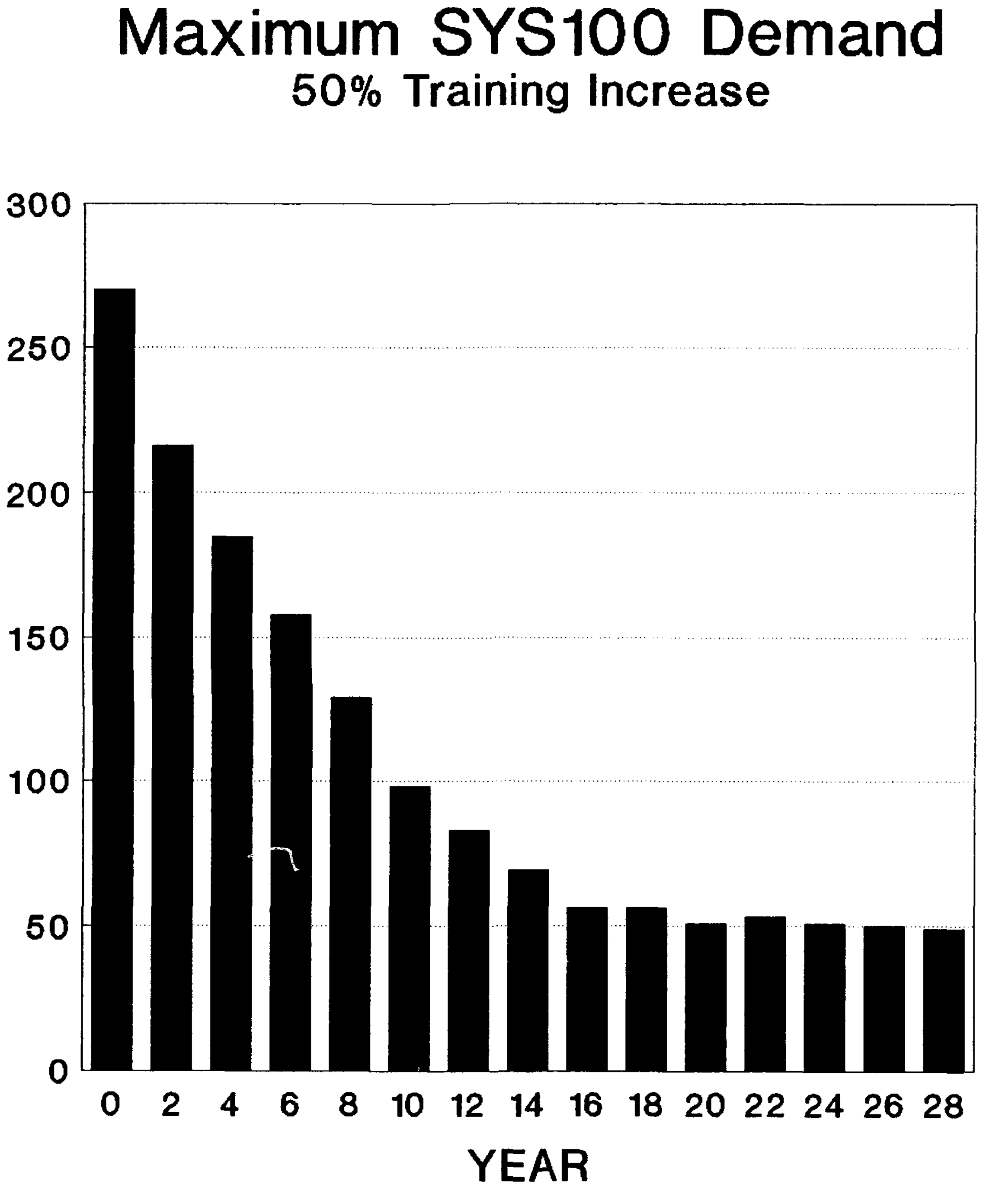

Backlog Training Demand
200% Training Increase
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
BACKLOG POPULATION

Backlog Training Demand

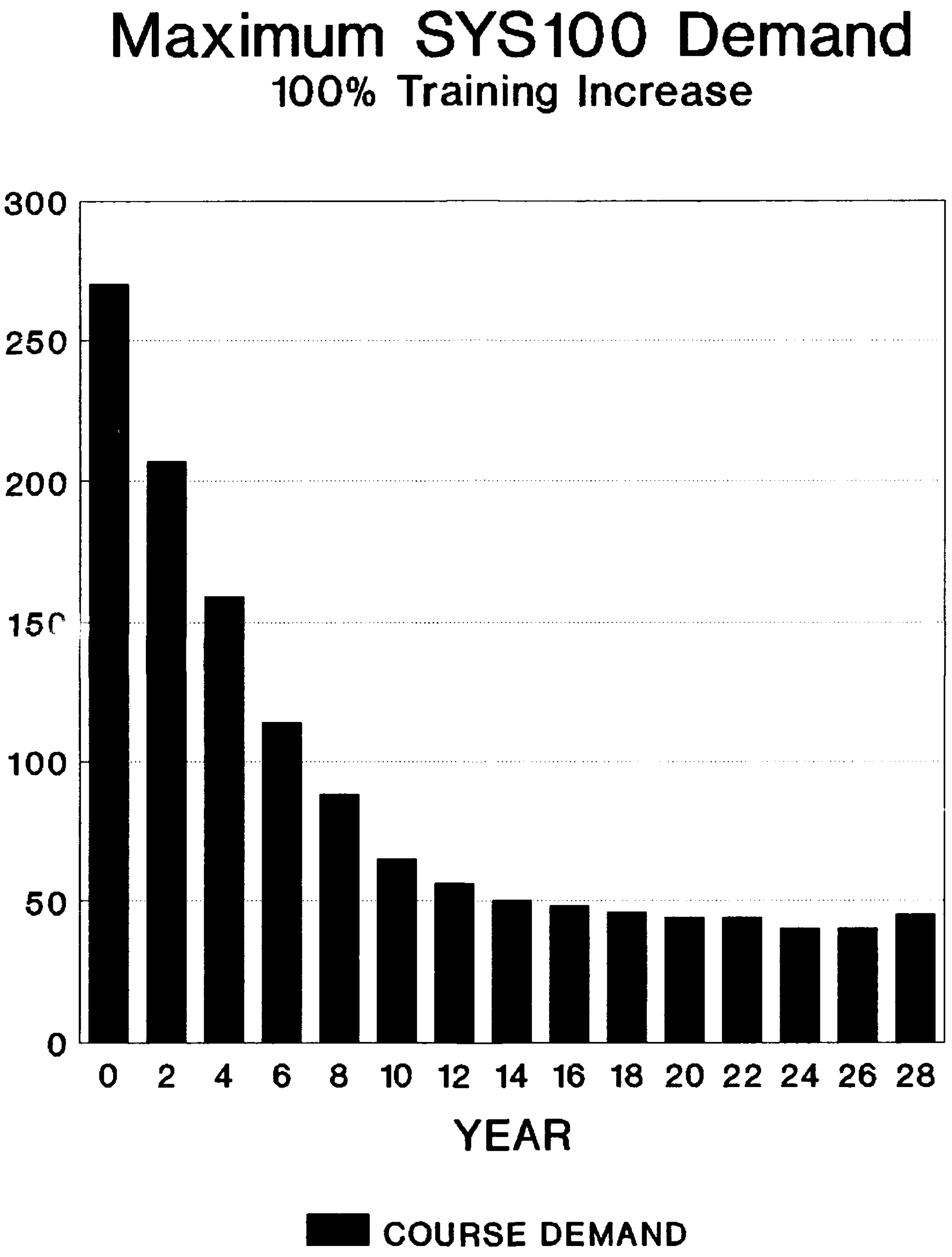
250% Training Increase




Maximum SYS100 Demand
Based on 1991 Training Capability
300
250
200
150
100
50
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

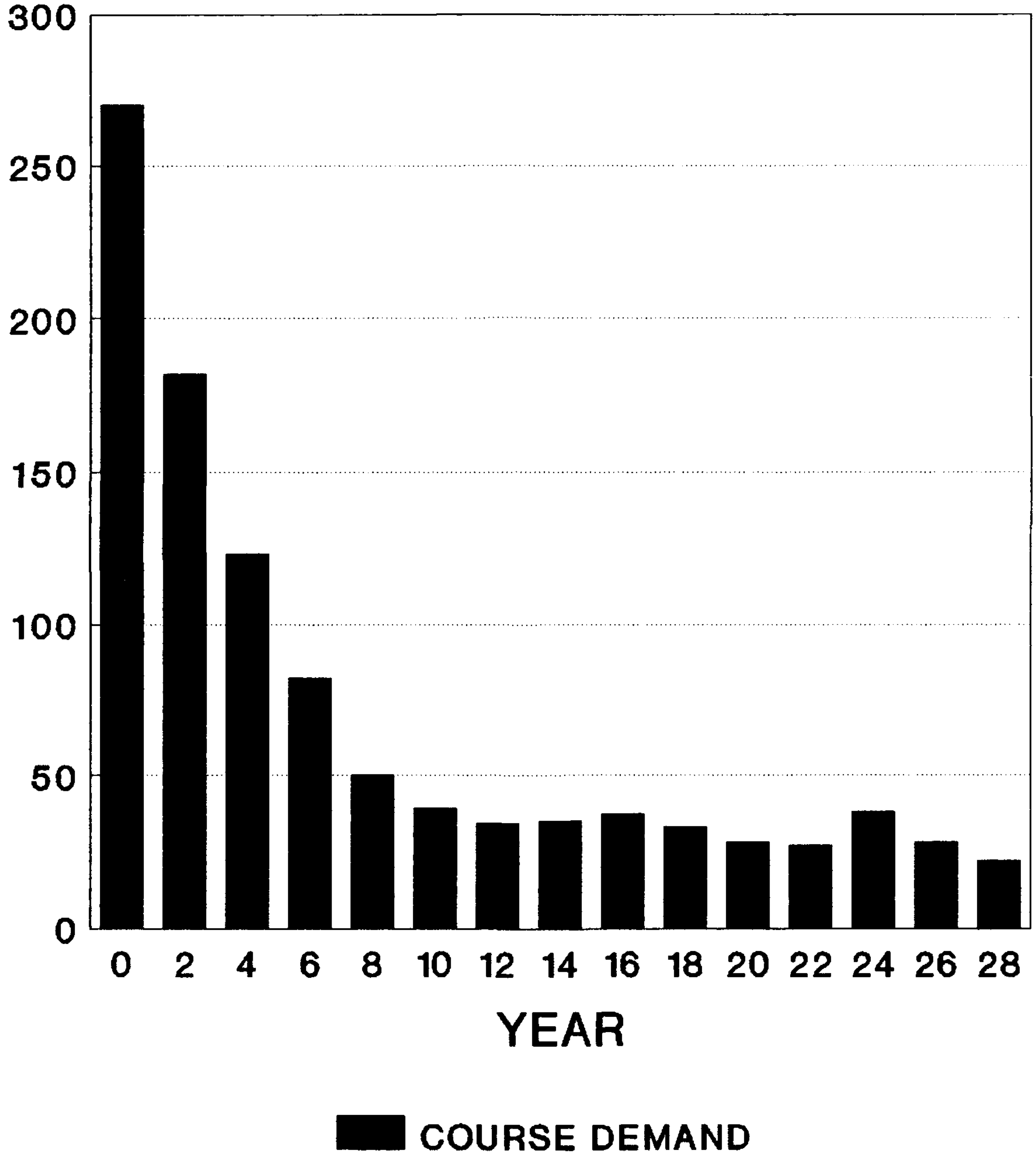


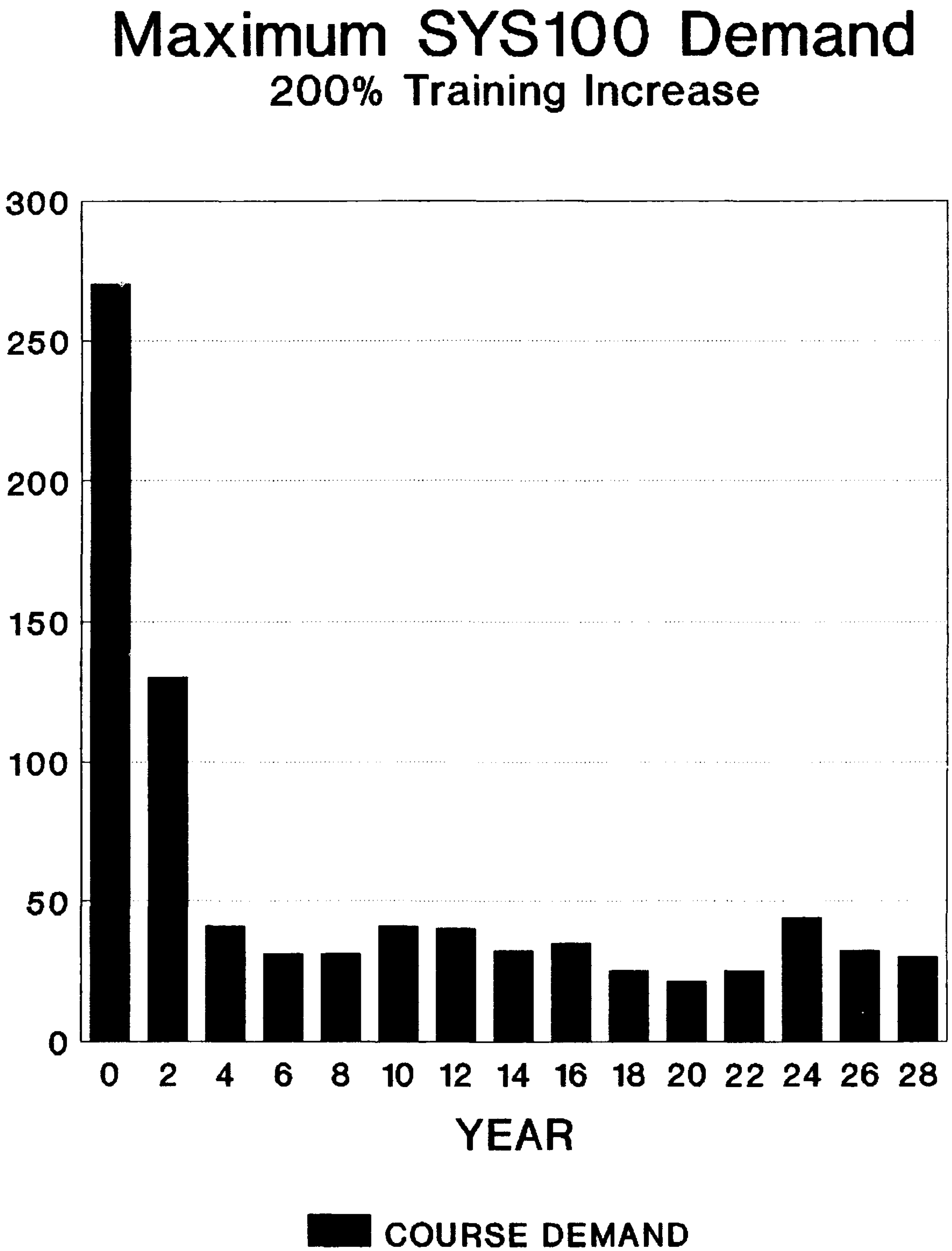
Maximum SYS100 Demand
50% Training Increase
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND

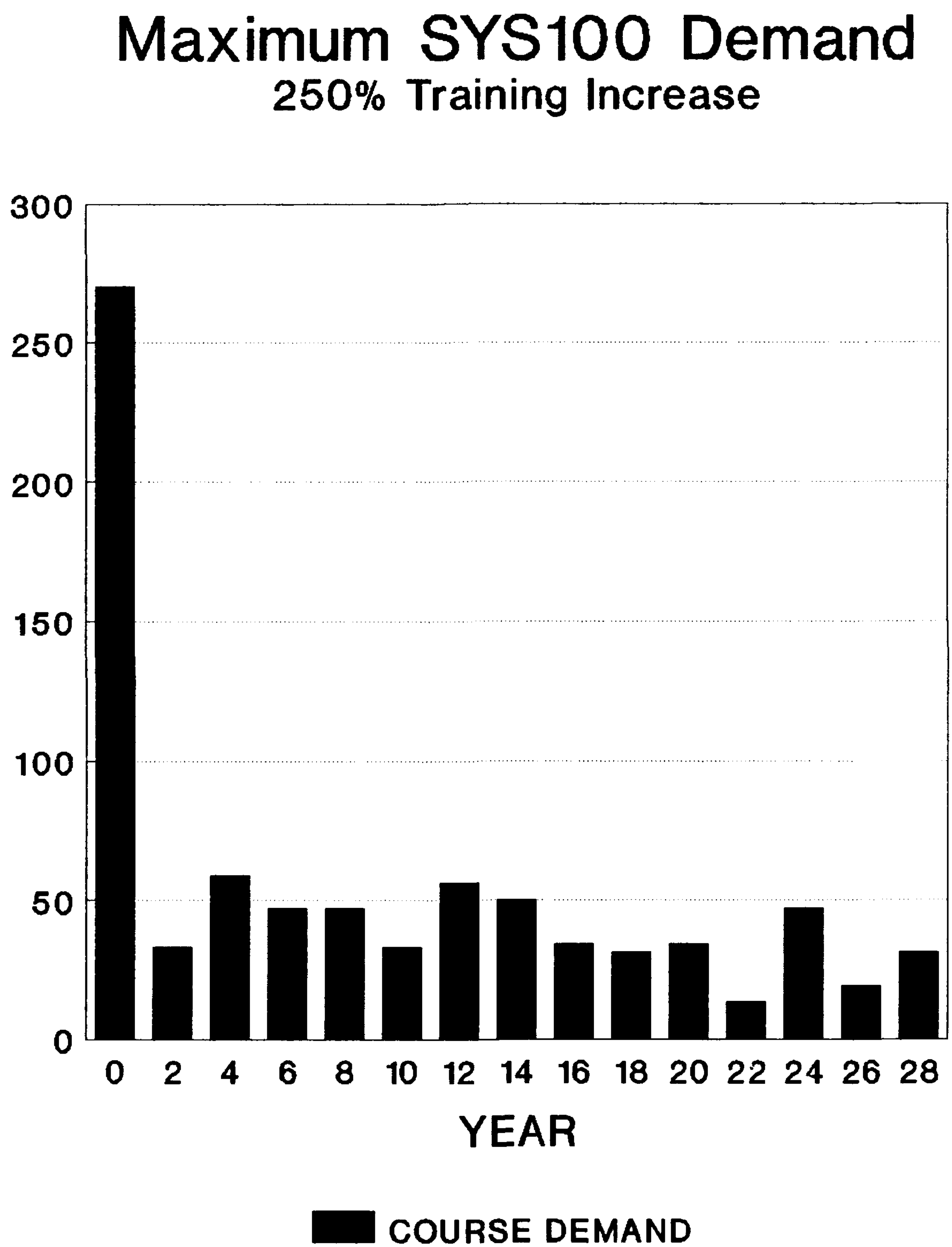


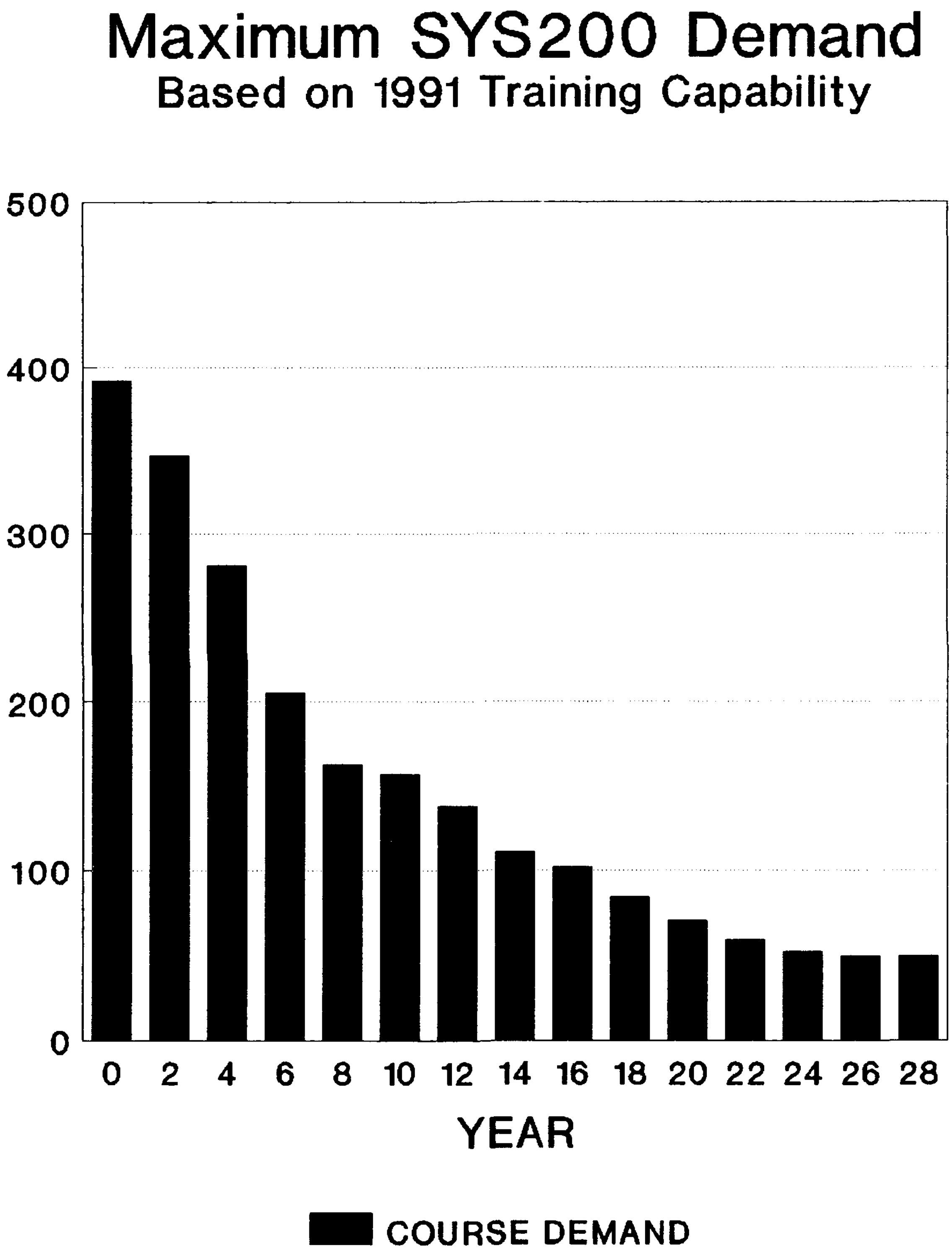
Maximum SYS100 Demand
100% Training Increase
300
250
200
150
100
50
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

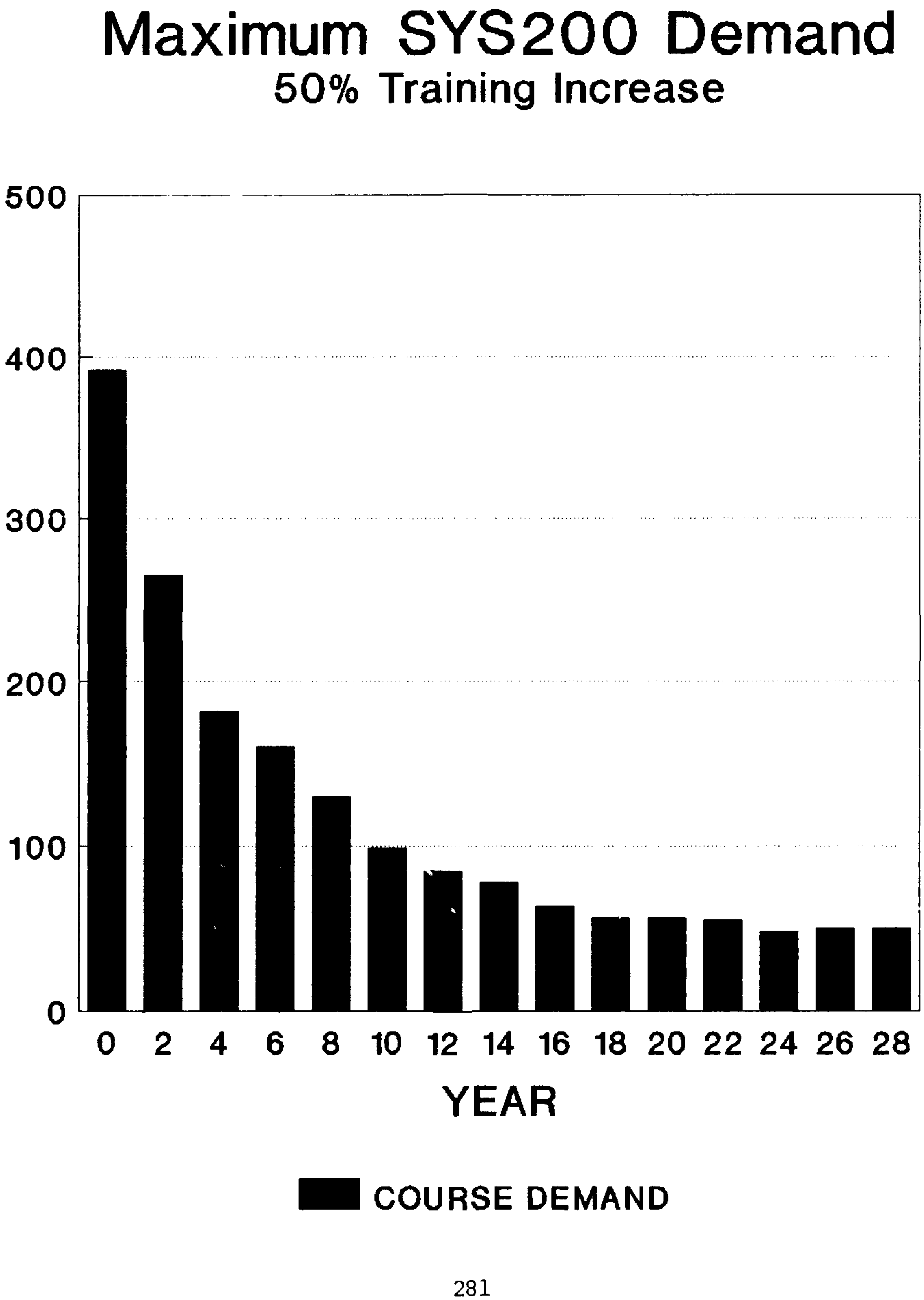
Maximum SYS100 Demand
150% Training Increase




Maximum SYS100 Demand
200% Training Increase
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND

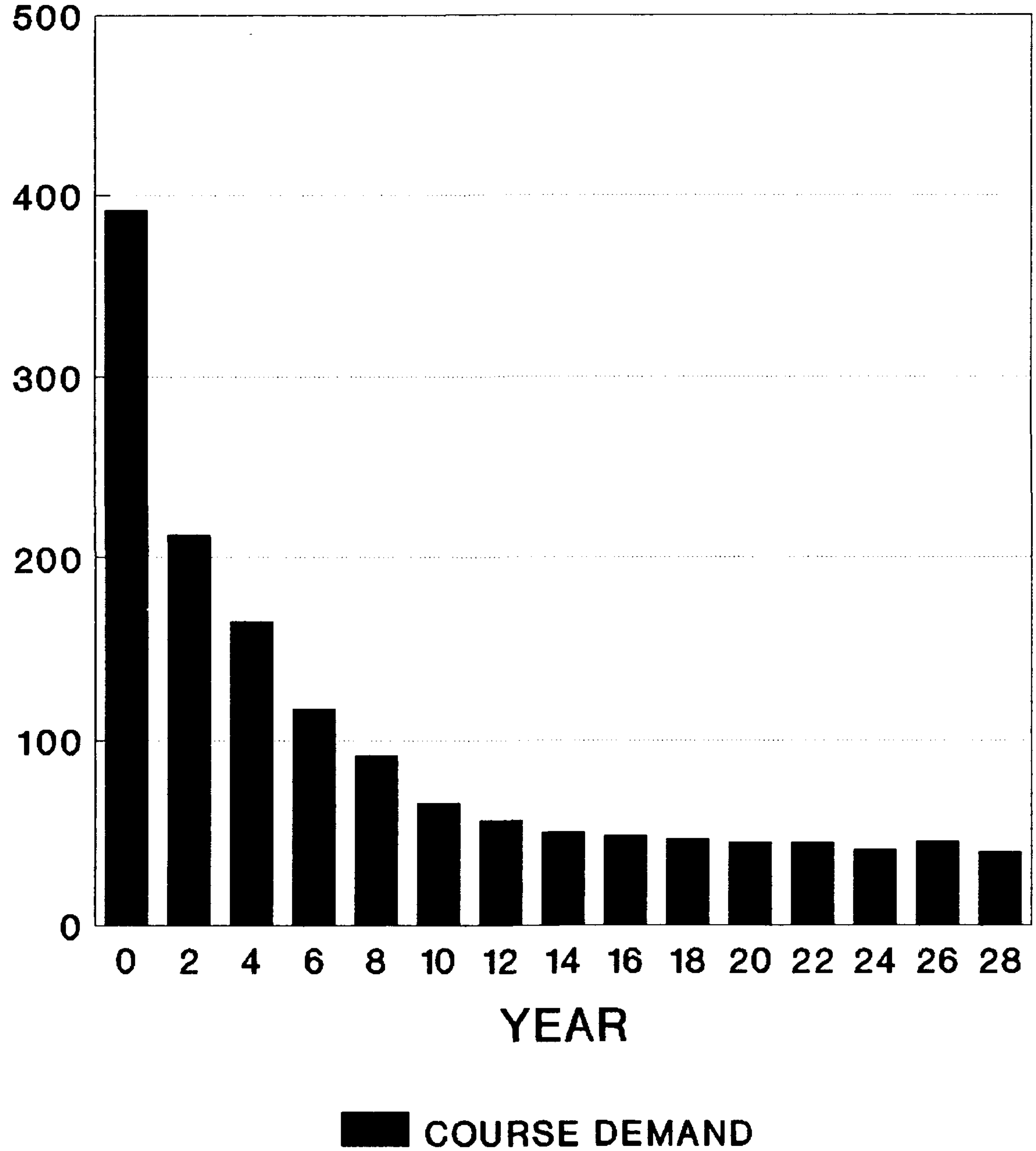

Maximum SYS100 Demand
250% Training Increase
300
250
200
150
100
50
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

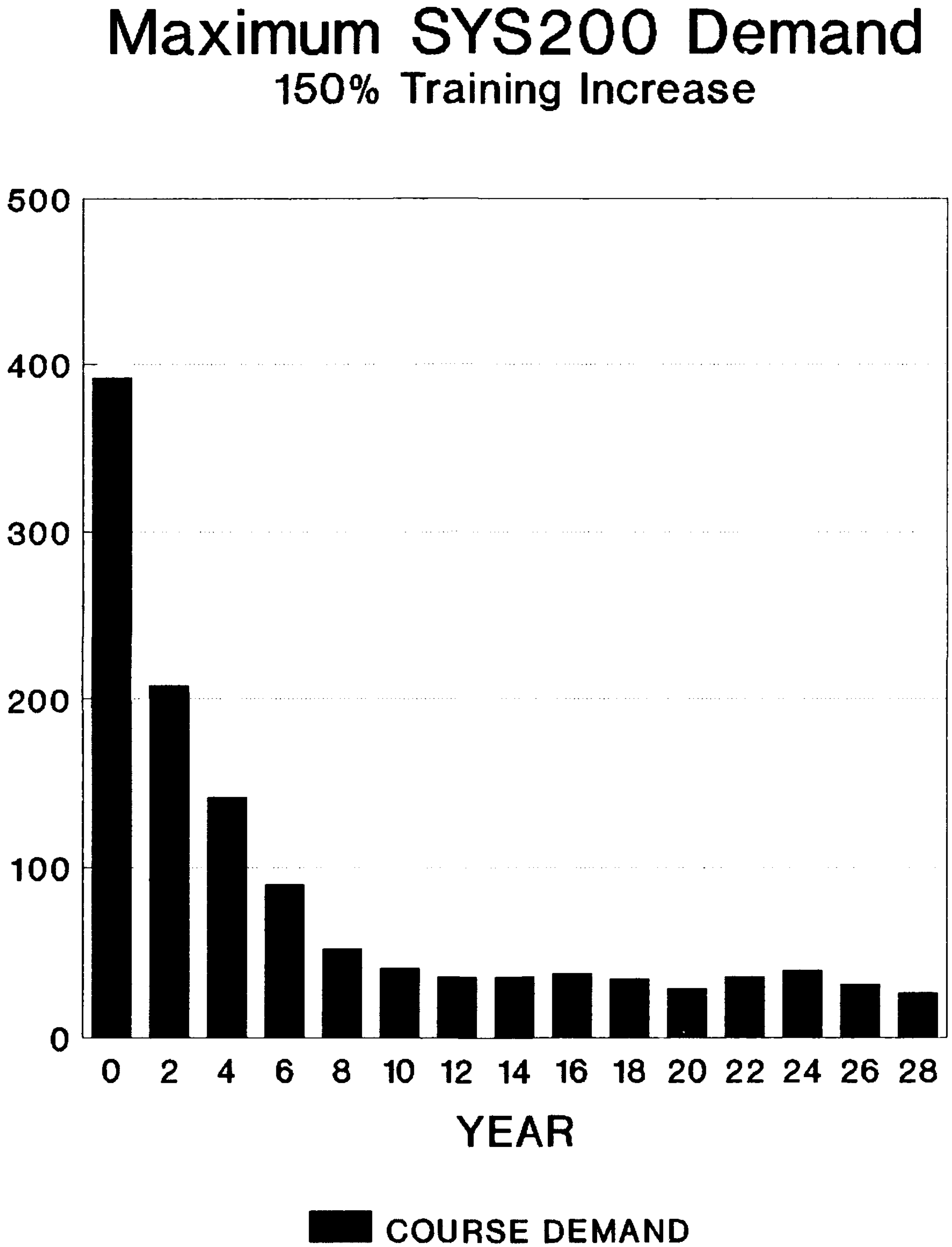

Maximum SYS200 Demand
Based on 1991 Training Capability
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND


Maximum SYS200 Demand
50% Training Increase
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

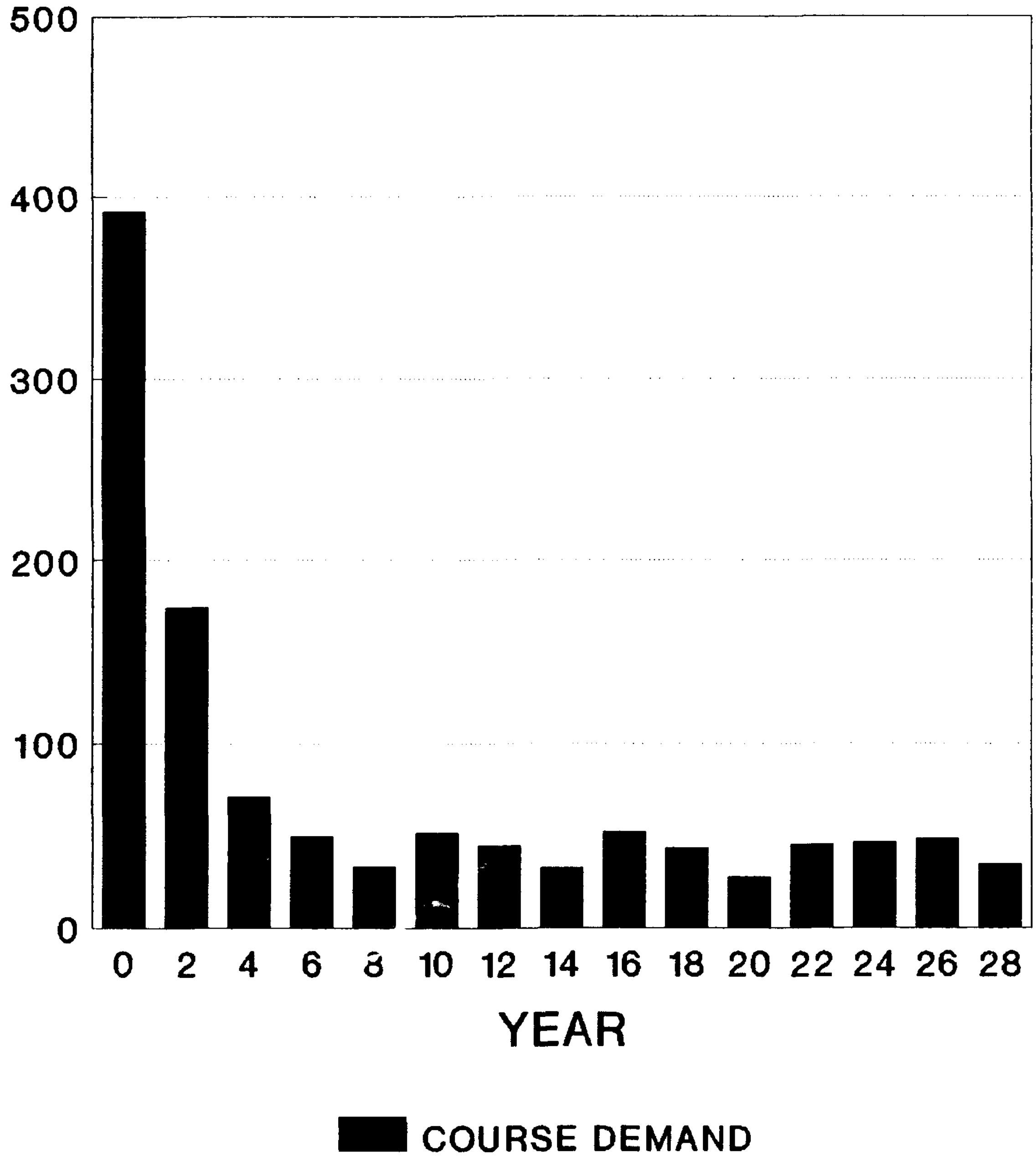
Maximum SYS200 Demand

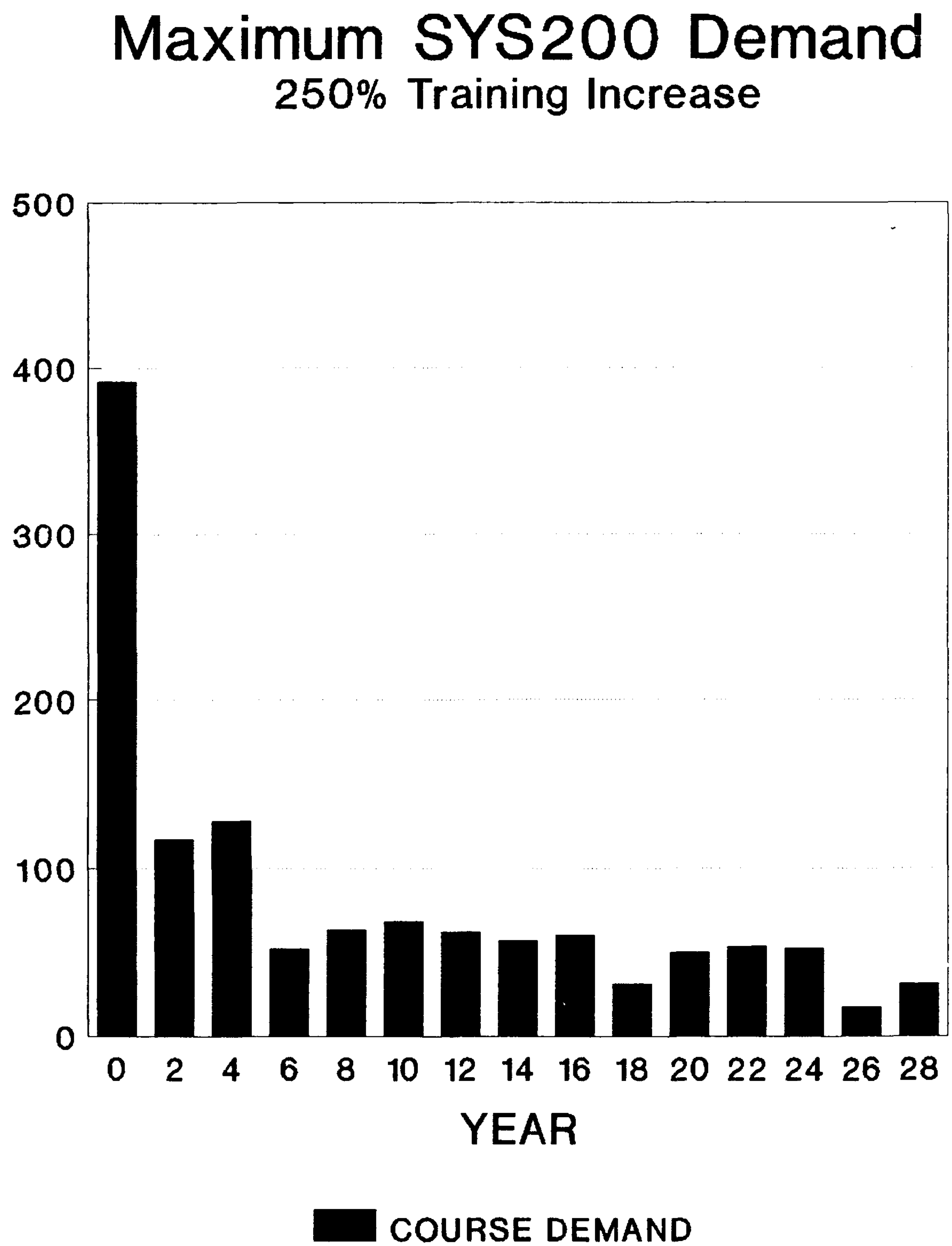
100% Training Increase

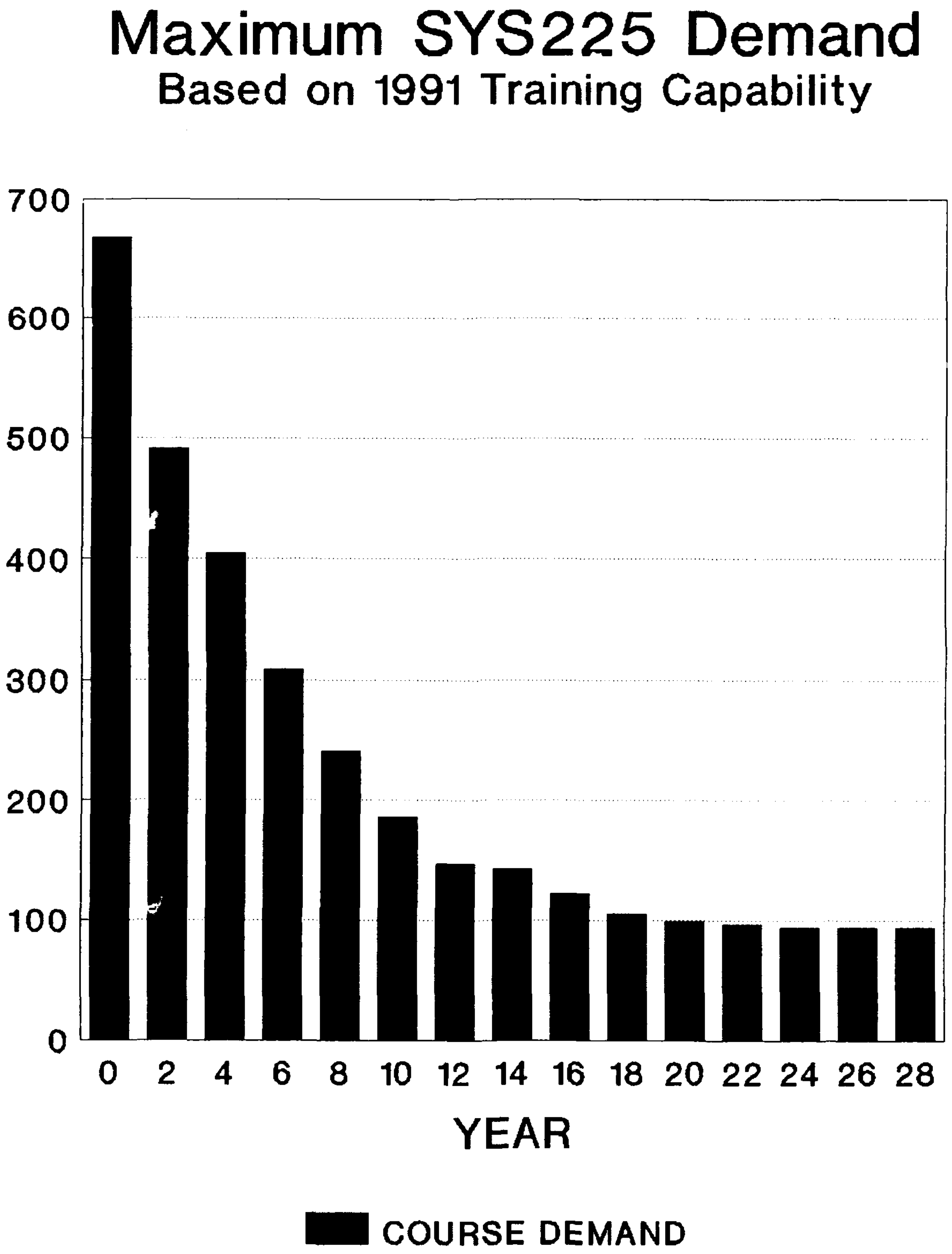



Maximum SYS200 Demand
150% Training Increase
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

Maximum SYS200 Demand
200% Training Increase



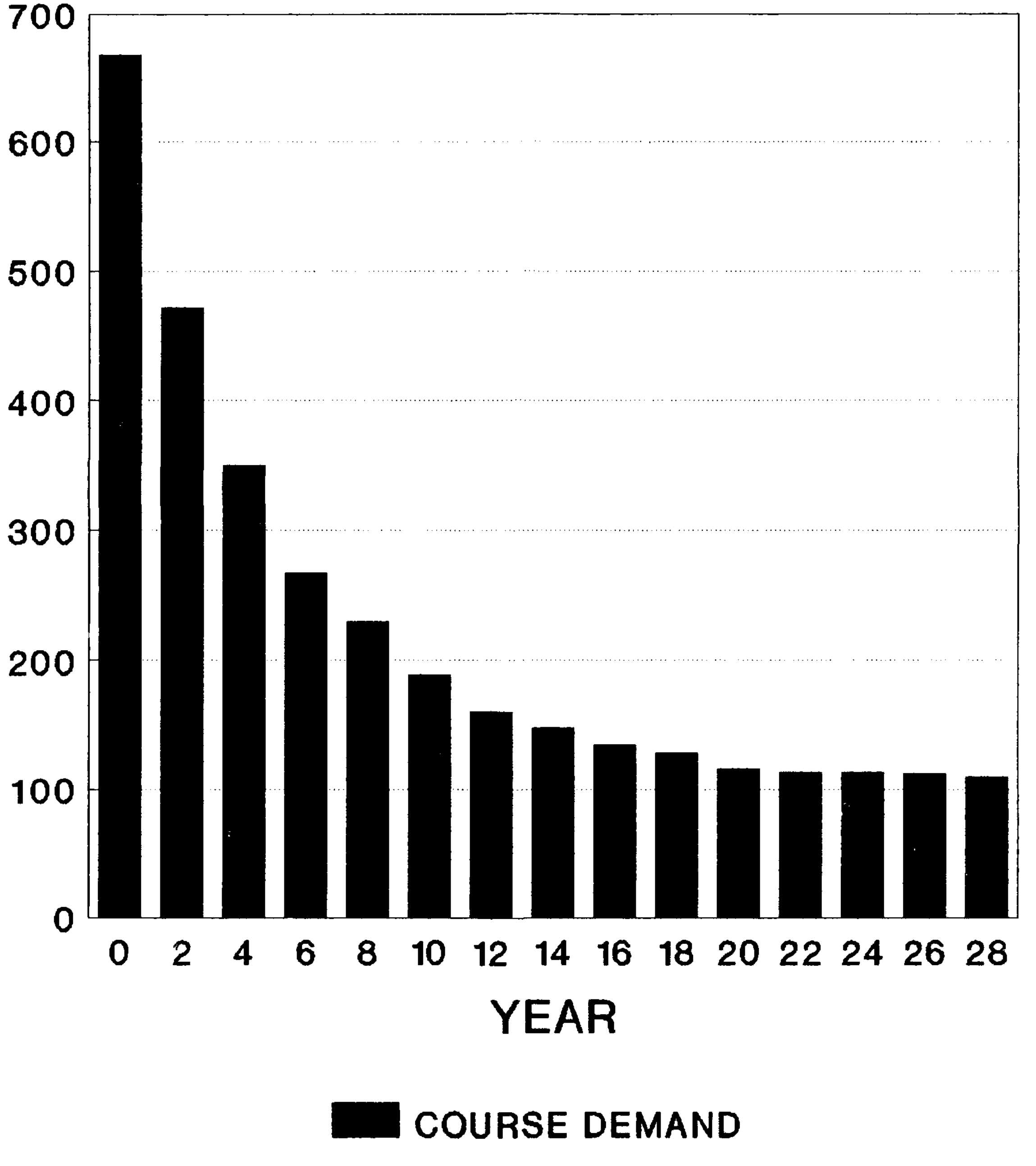

Maximum SYS200 Demand
250% Training Increase
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND



Maximum SYS225 Demand
Based on 1991 Training Capability
700
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

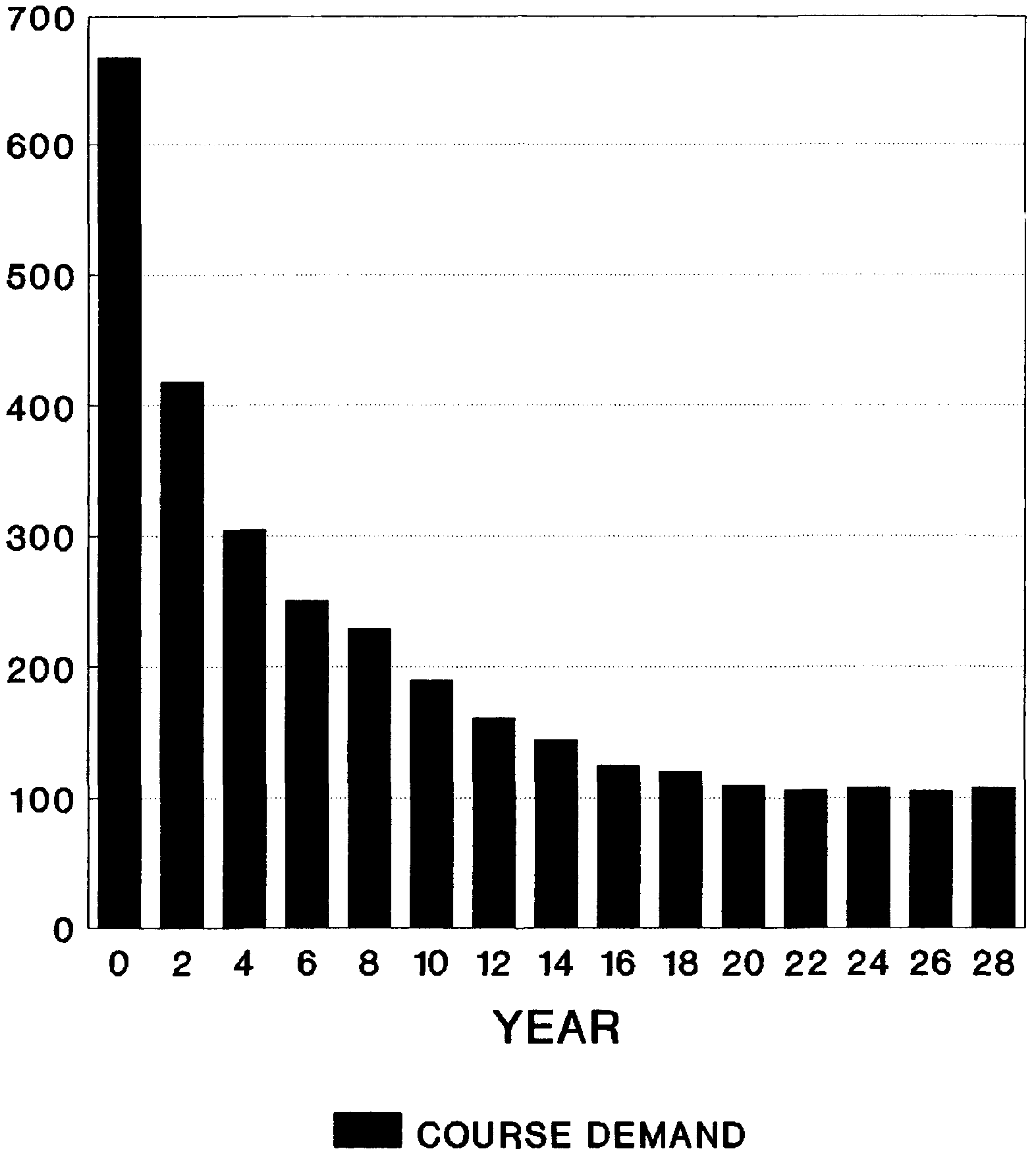
Maximum SYS225 Demand

50% Training Increase



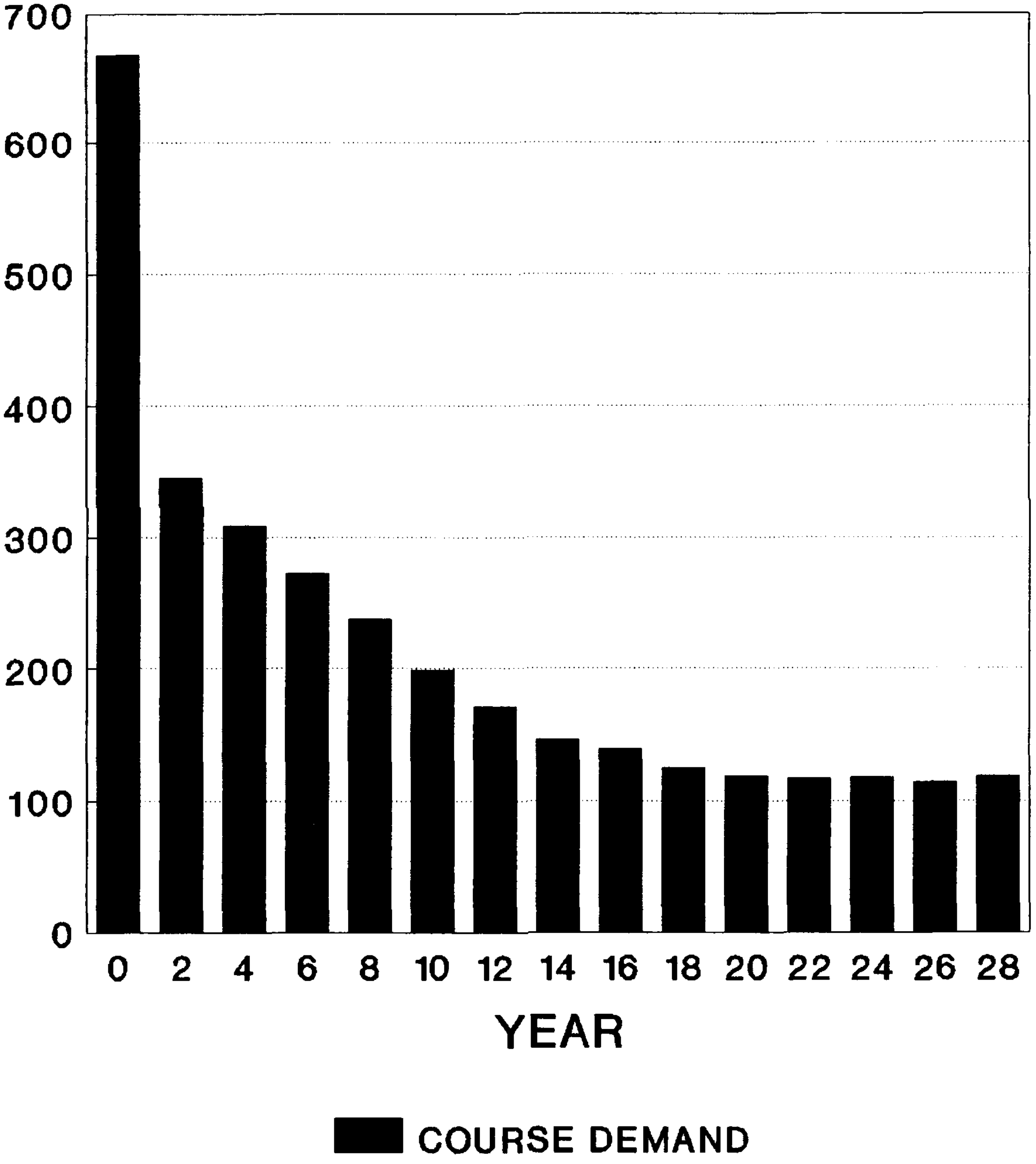
Maximum SYS225 Demand

100% Training Increase

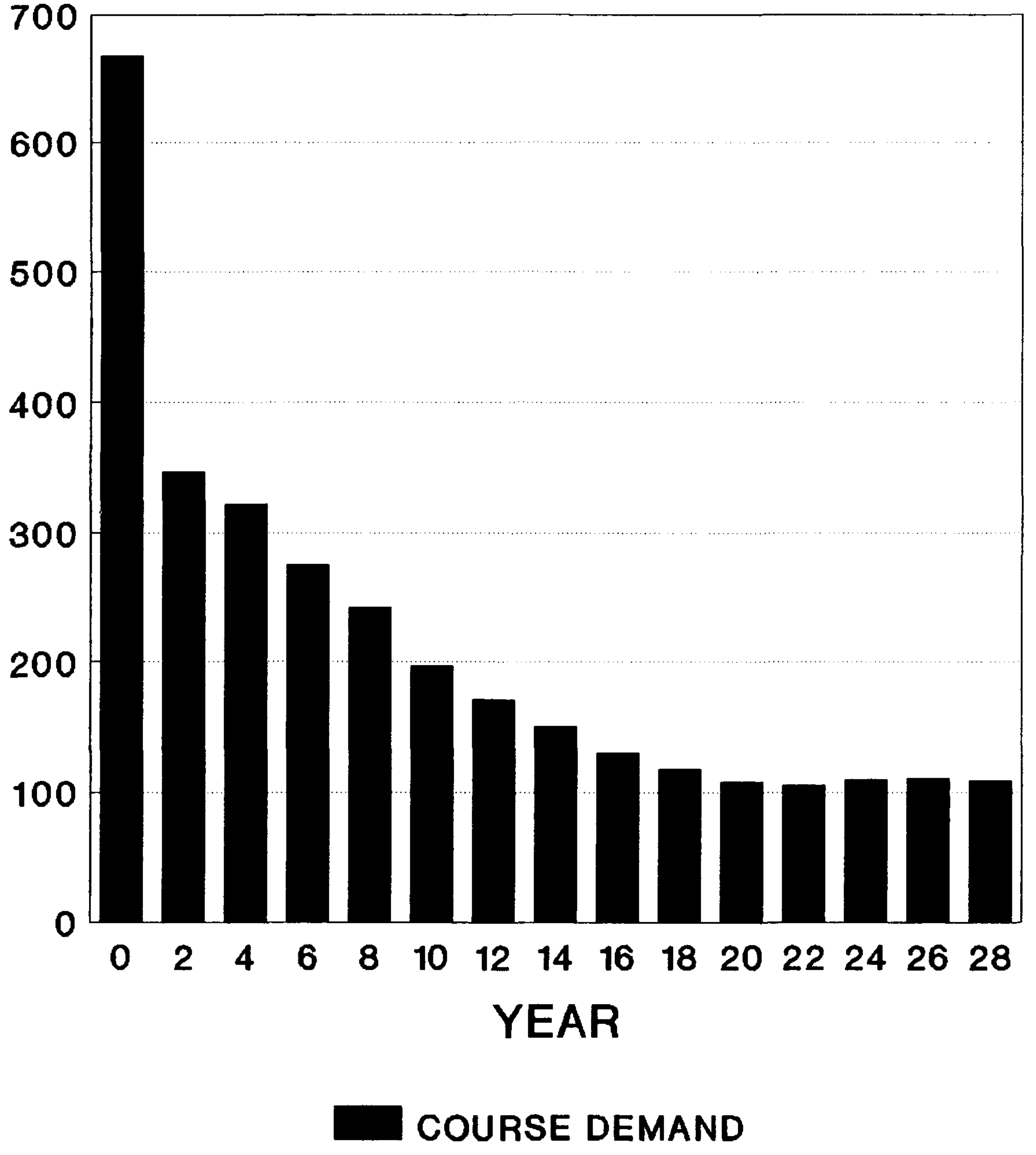


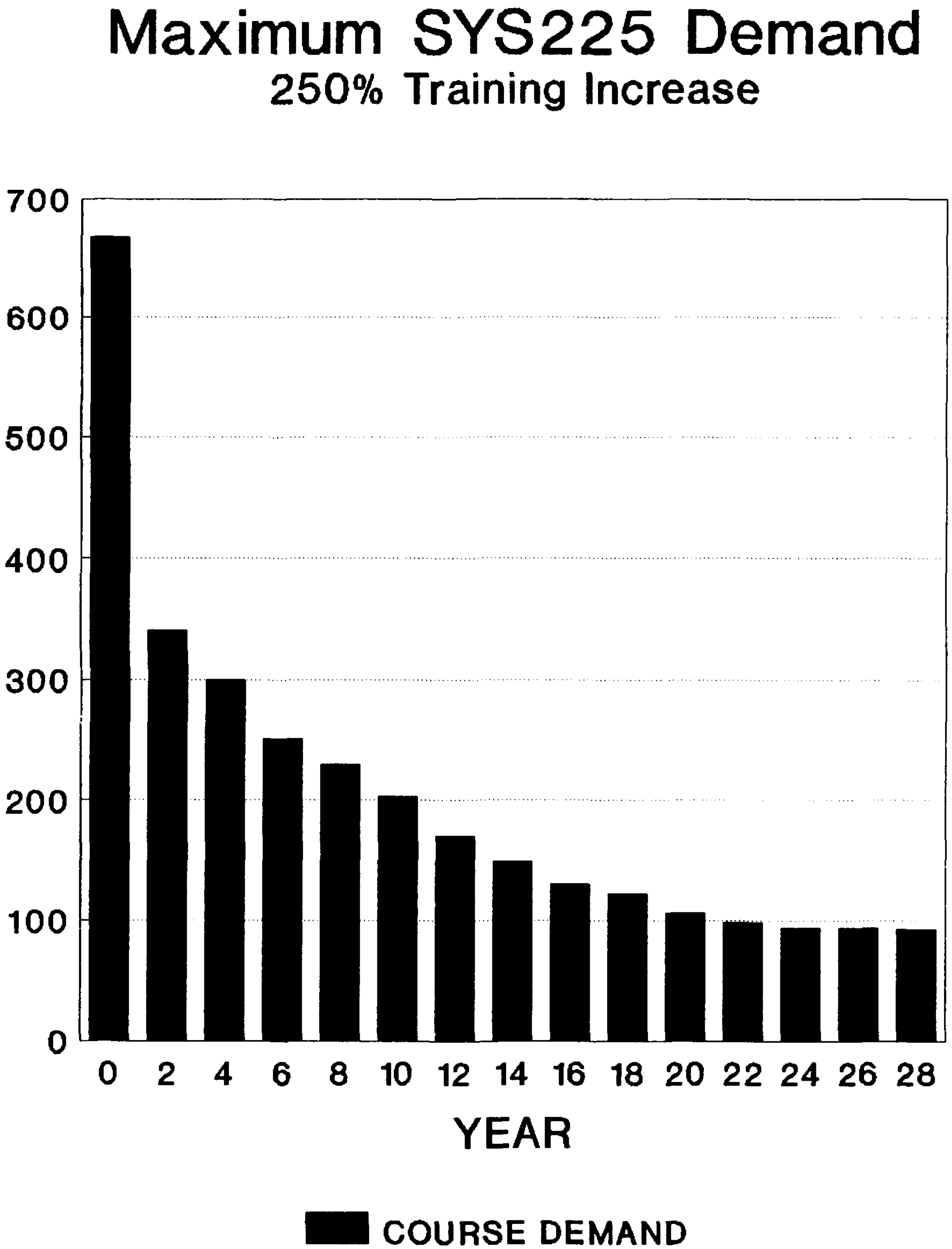
Maximum SYS225 Demand

150% Training Increase

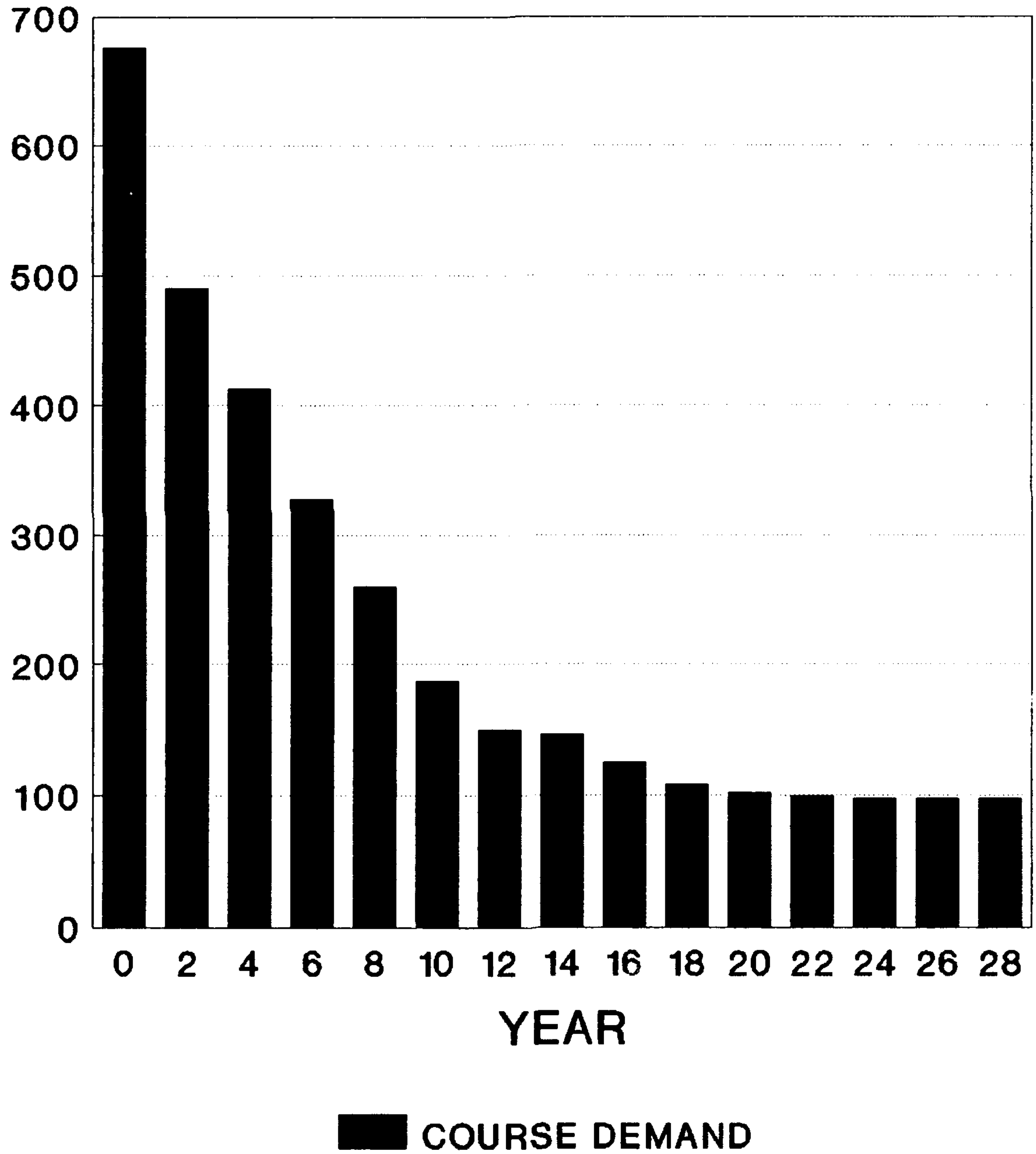


Maximum SYS225 Demand
200% Training Increase

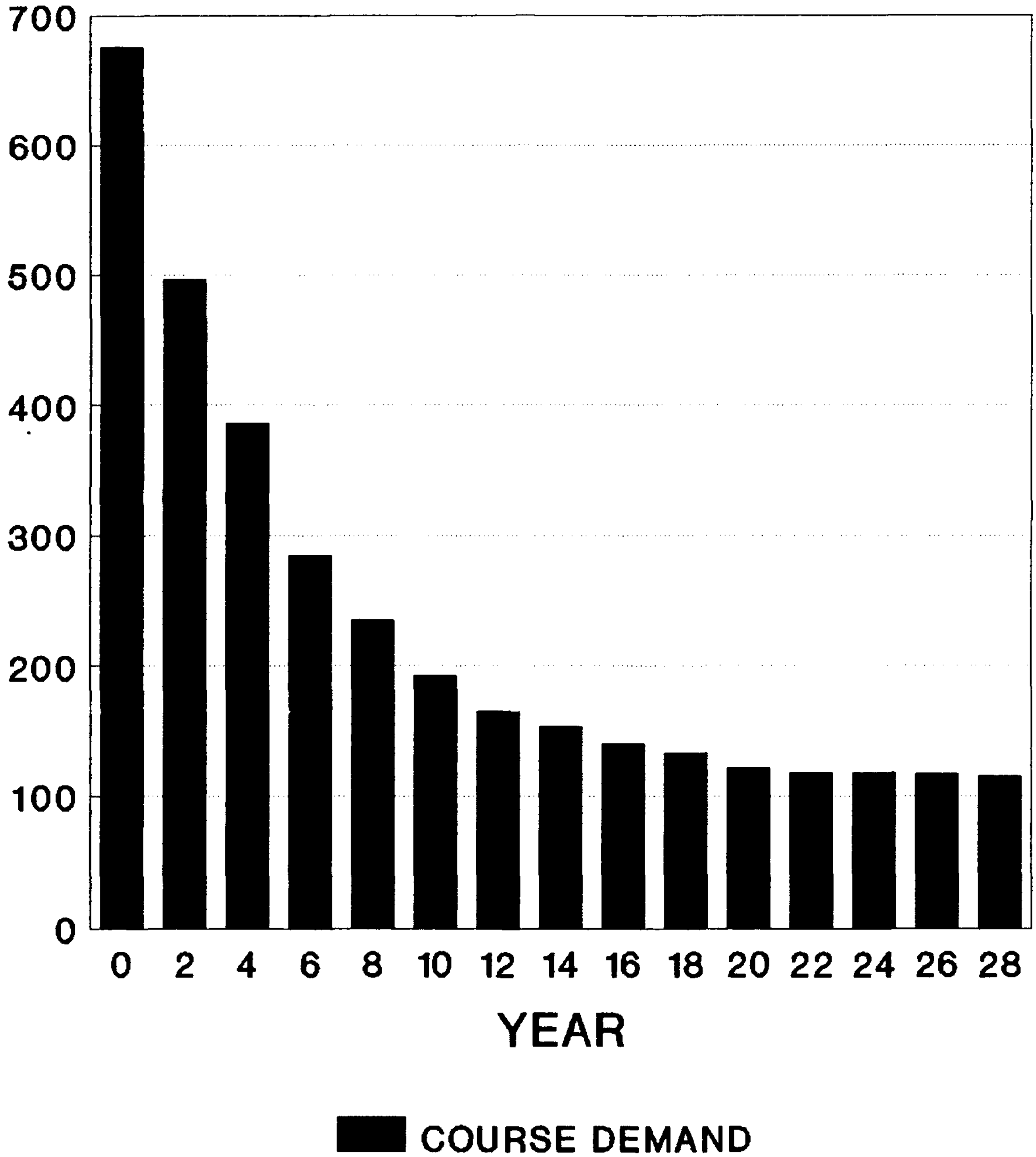


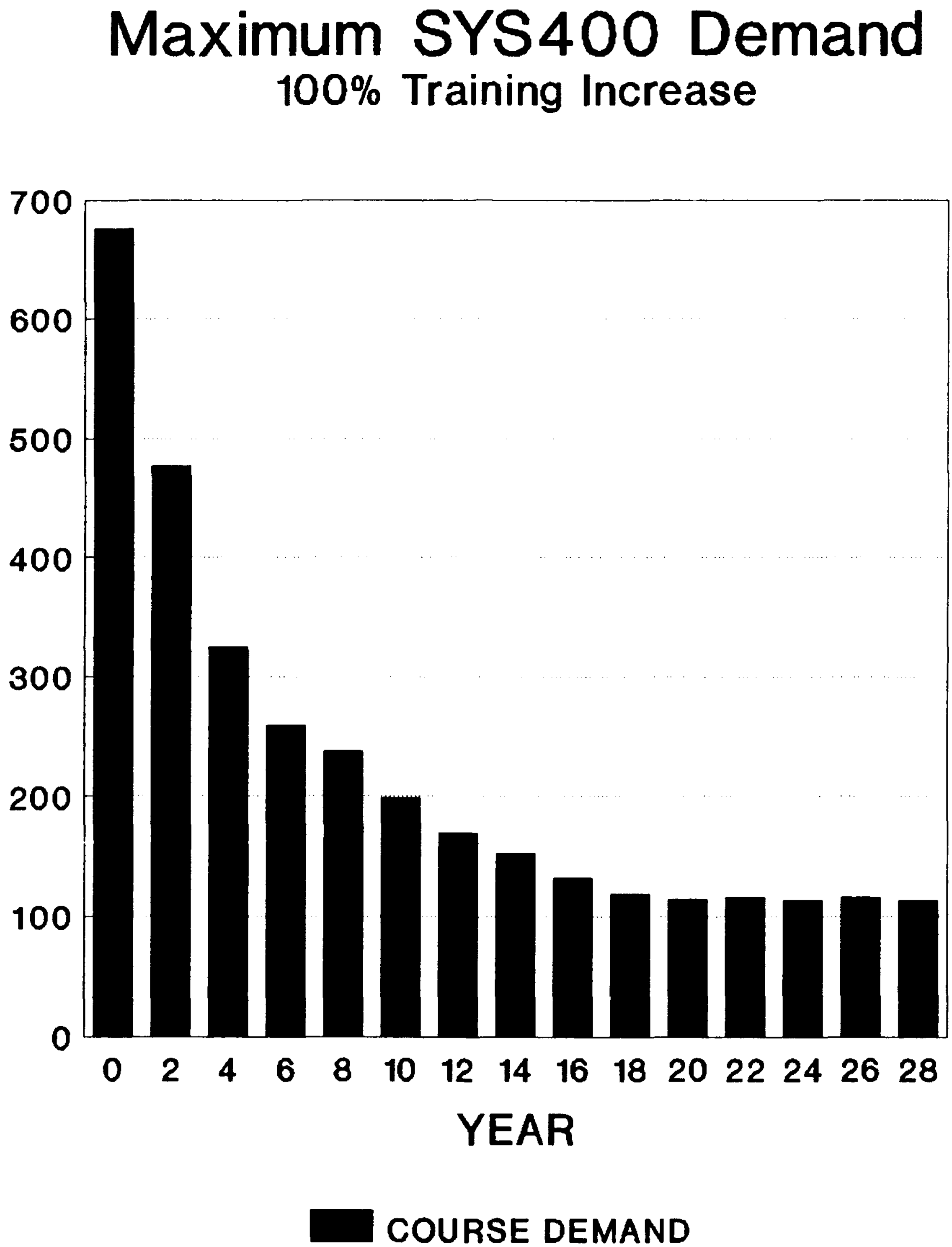

Maximum SYS225 Demand
250% Training Increase
700
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

Maximum SYS400 Demand
Based on 1991 Training Capability



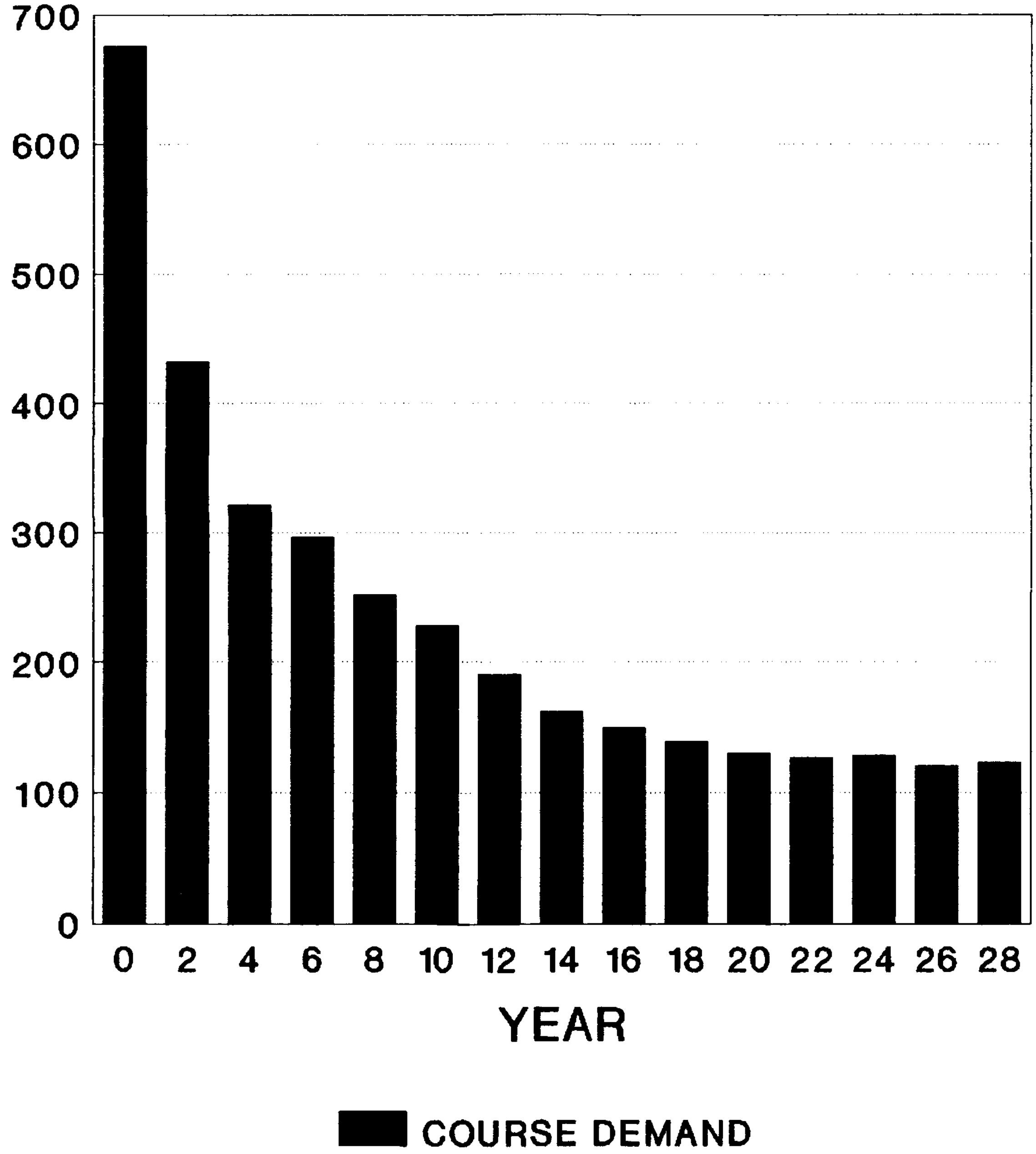
Maximum SYS400 Demand
50% Training Increase



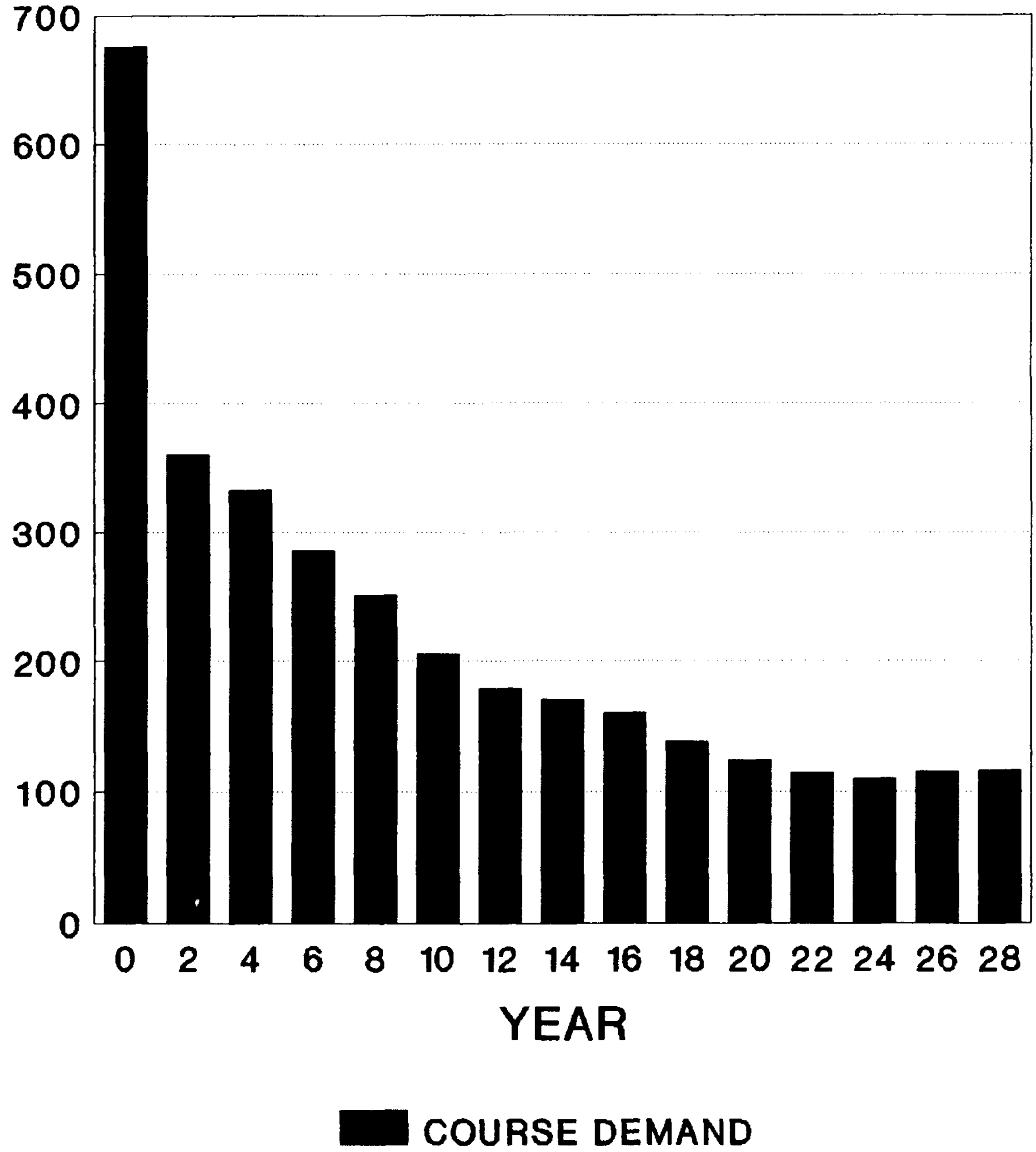

Maximum SYS400 Demand
100% Training Increase
700
600
500
400
300
200
100
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

Maximum SYS400 Demand

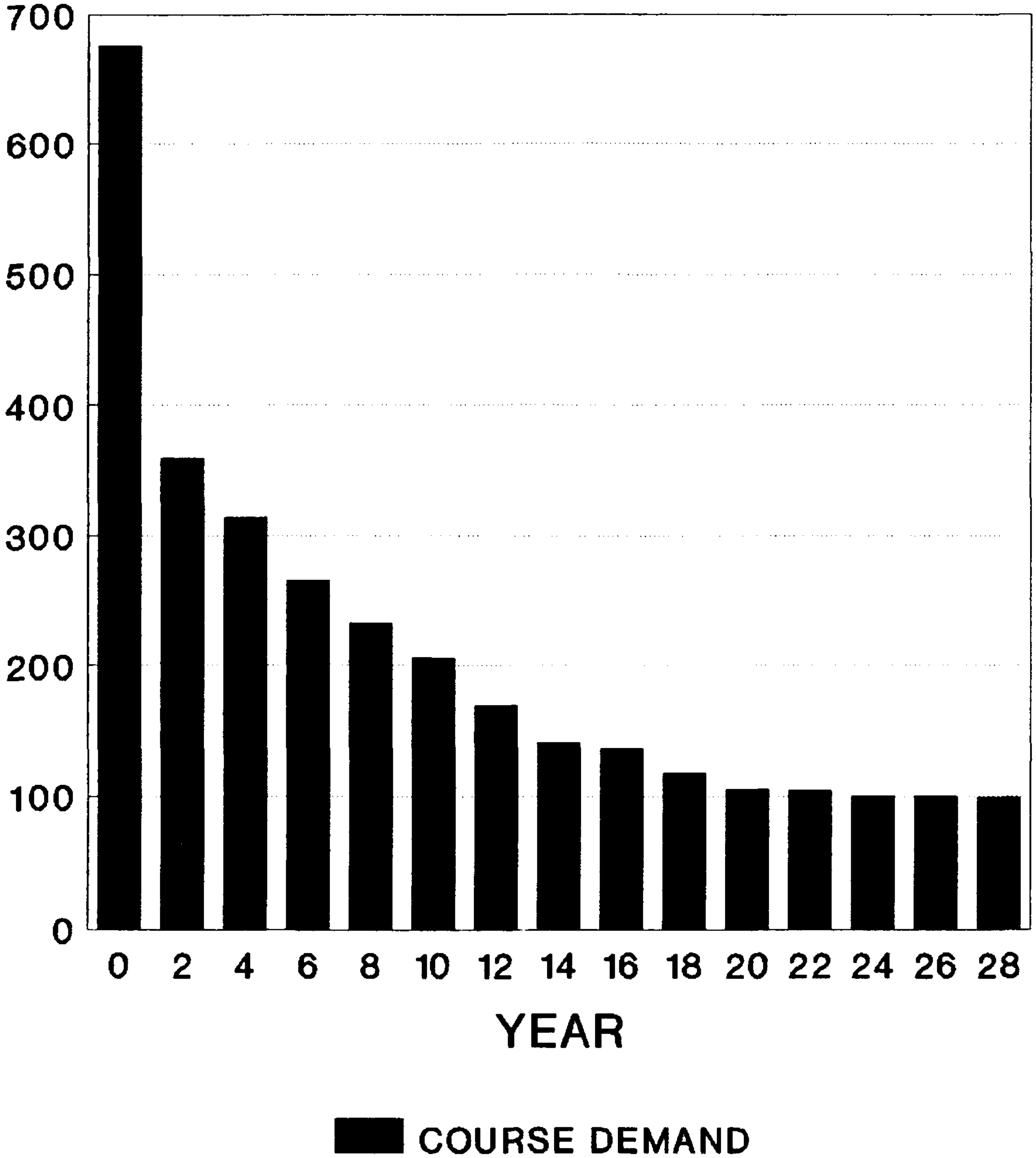
150% Training Increase



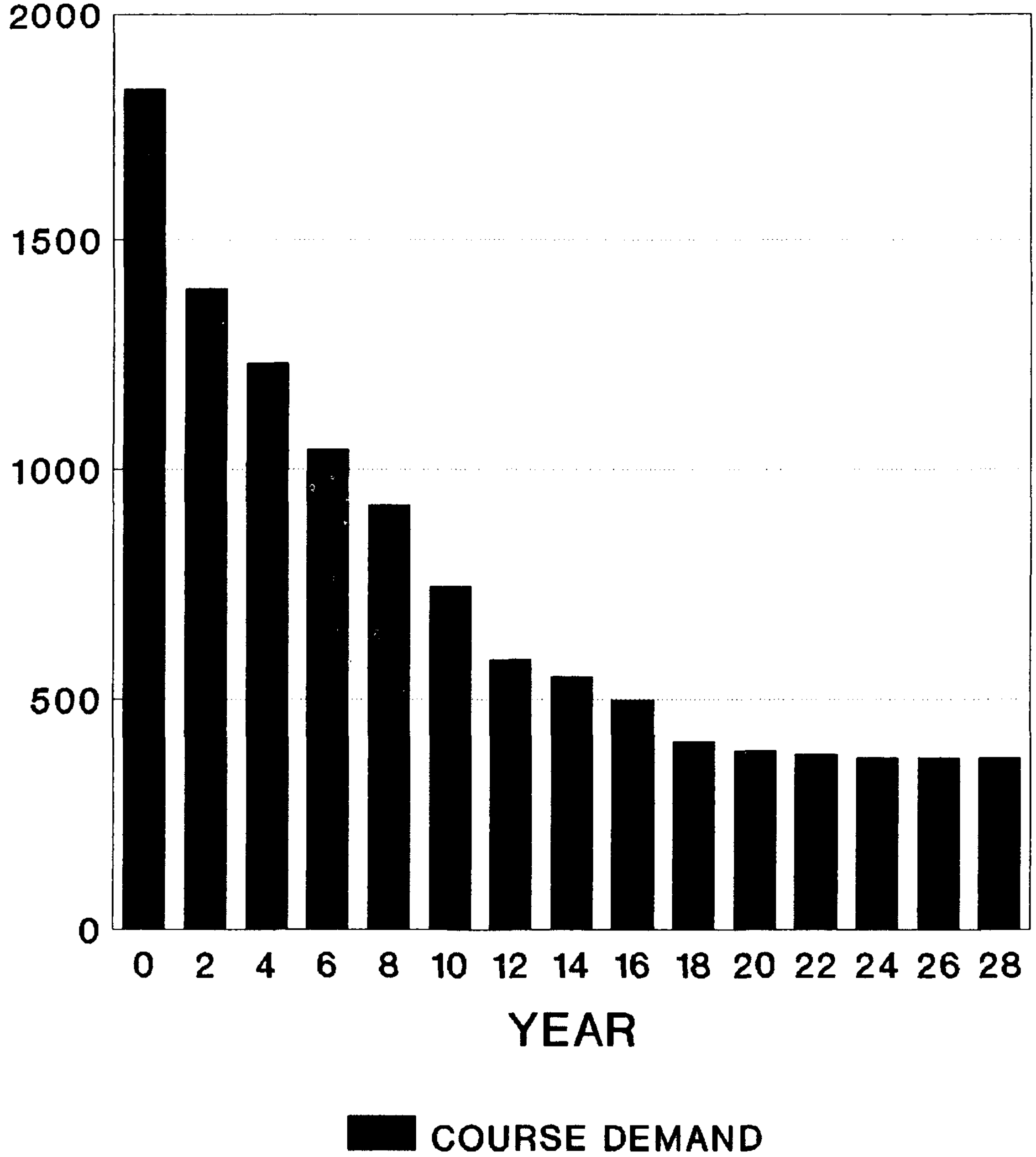
Maximum SYS400 Demand
200% Training Increase

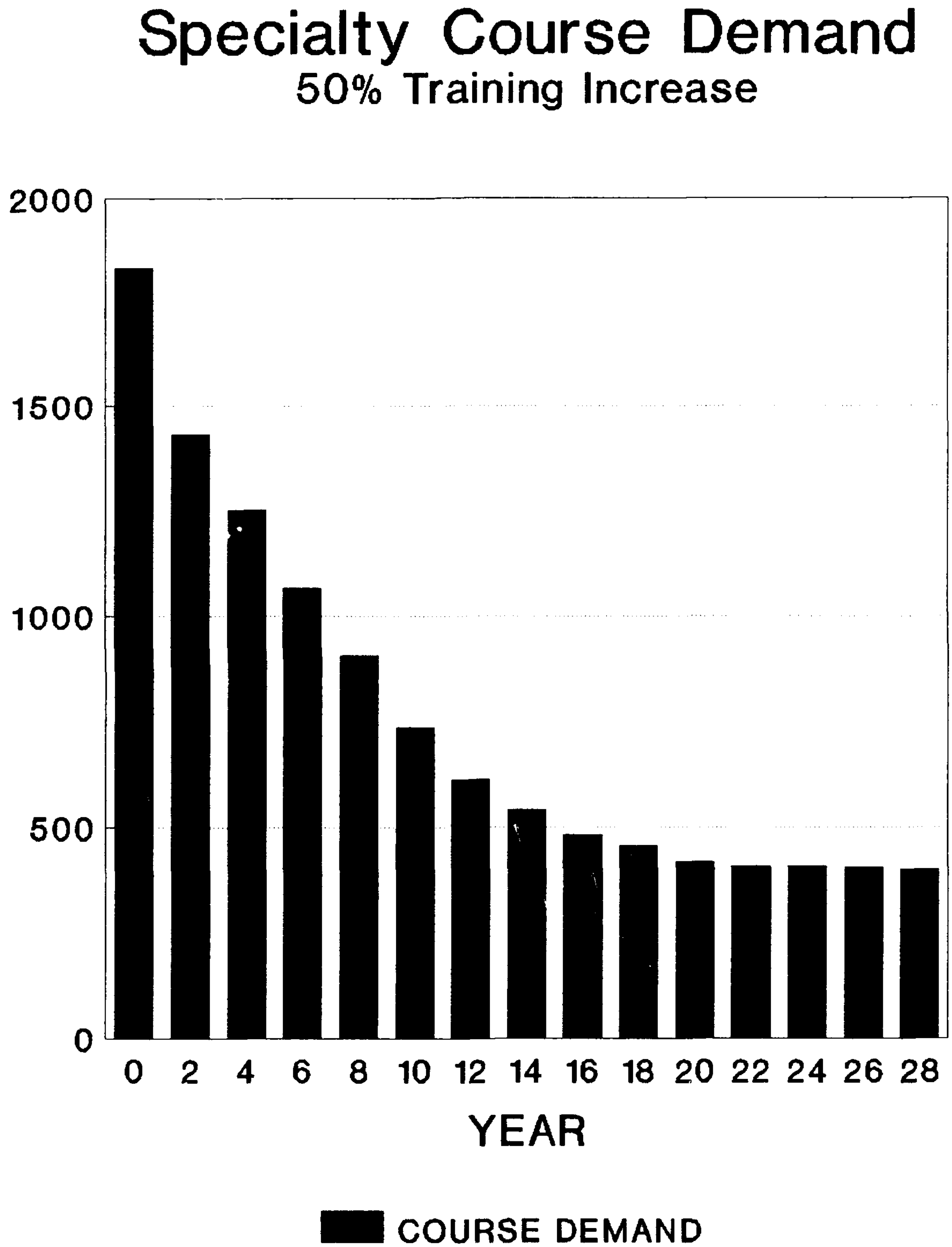


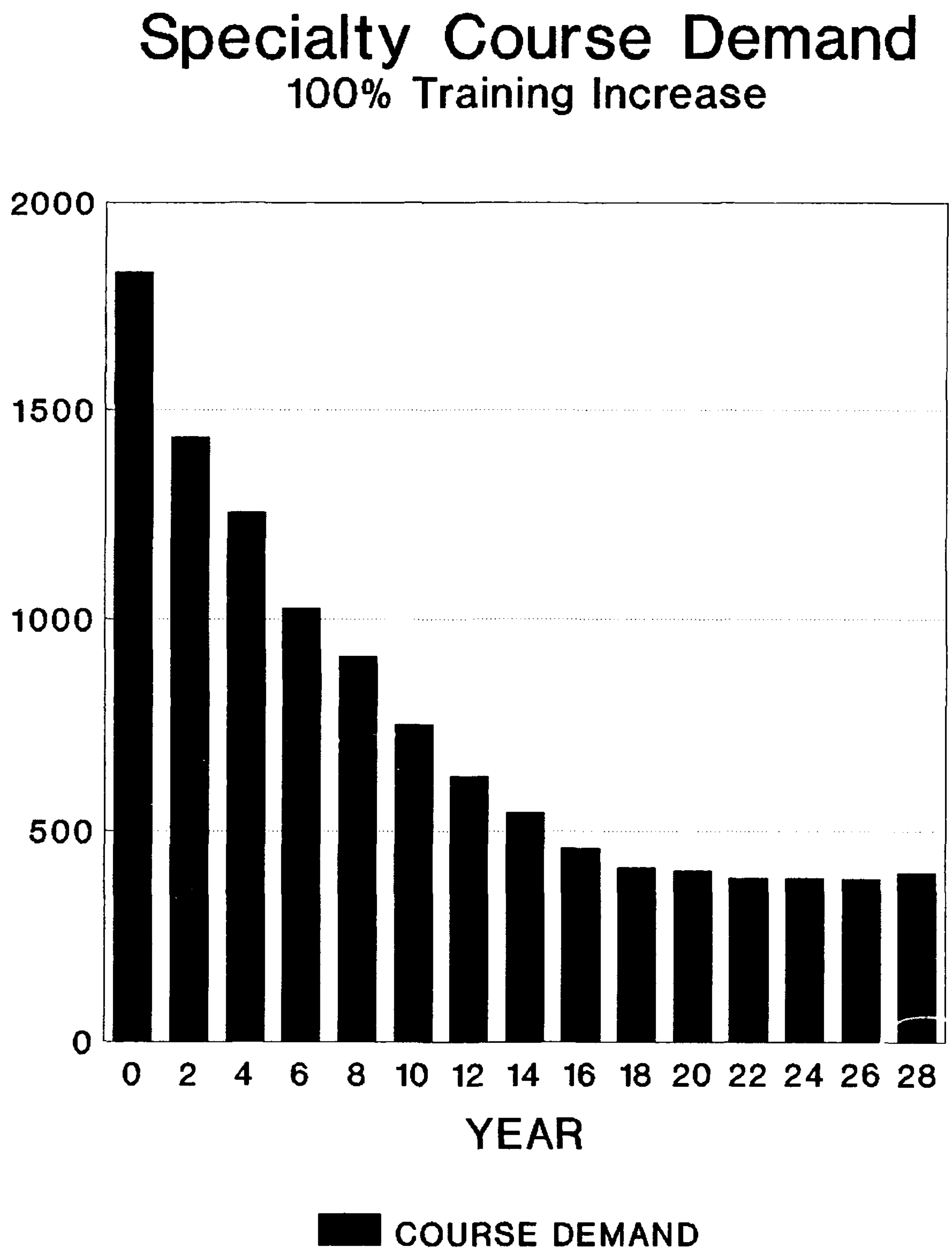
Maximum SYS400 Demand
250% Training Increase

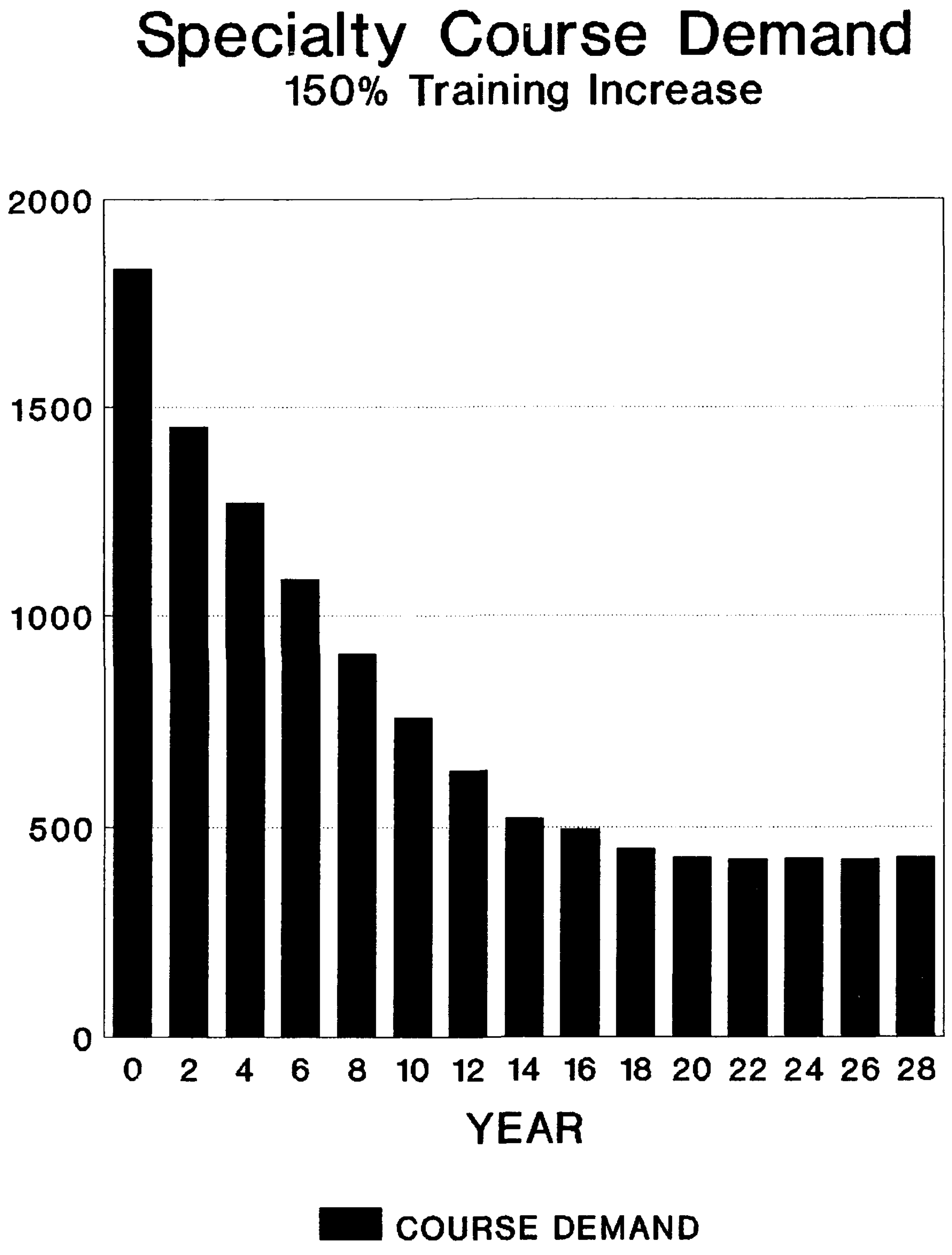


Specialty Course Demand
Based on 1991 Training Capability

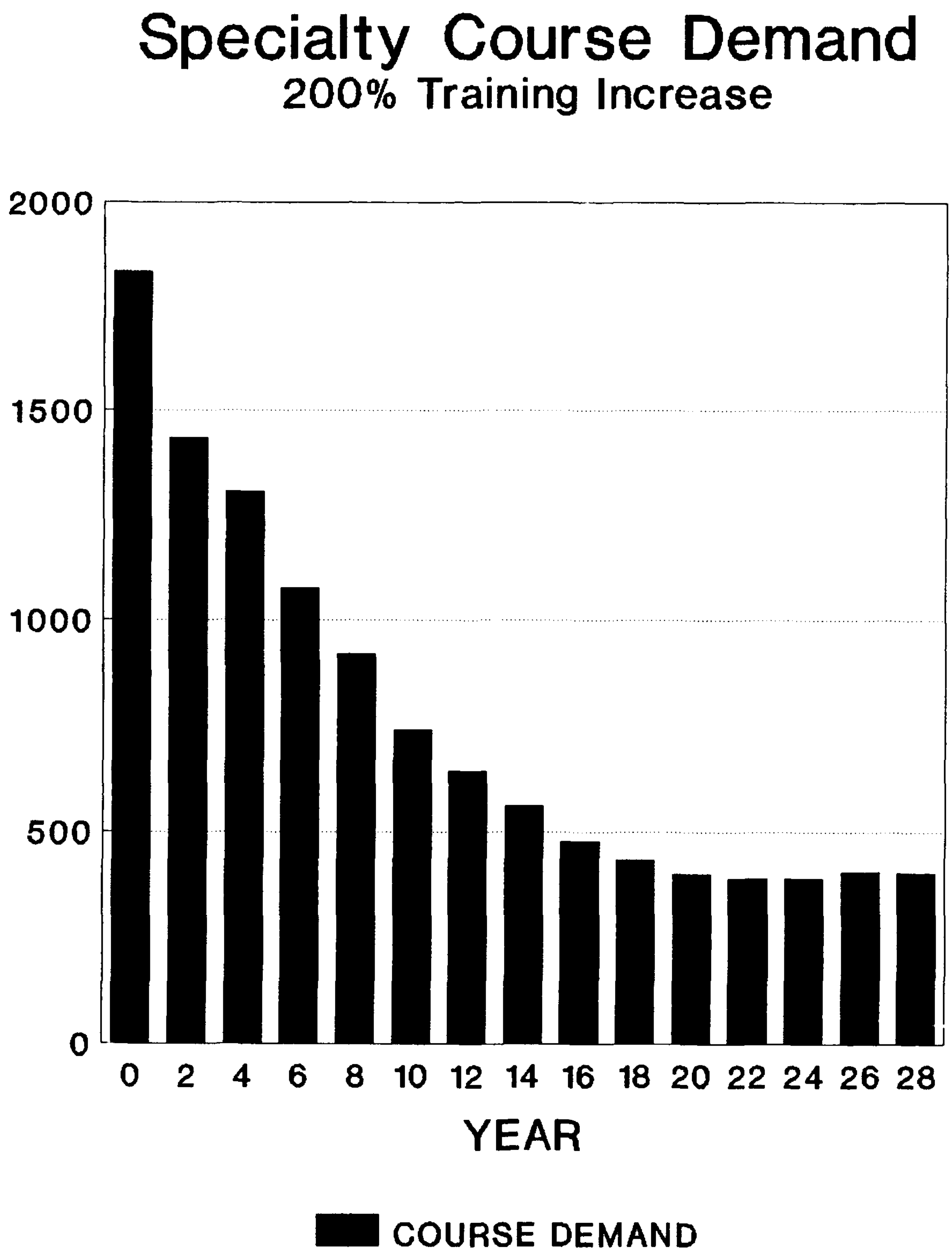



Specialty Course Demand
50% Training Increase
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND


Specialty Course Demand
100% Training Increase
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND

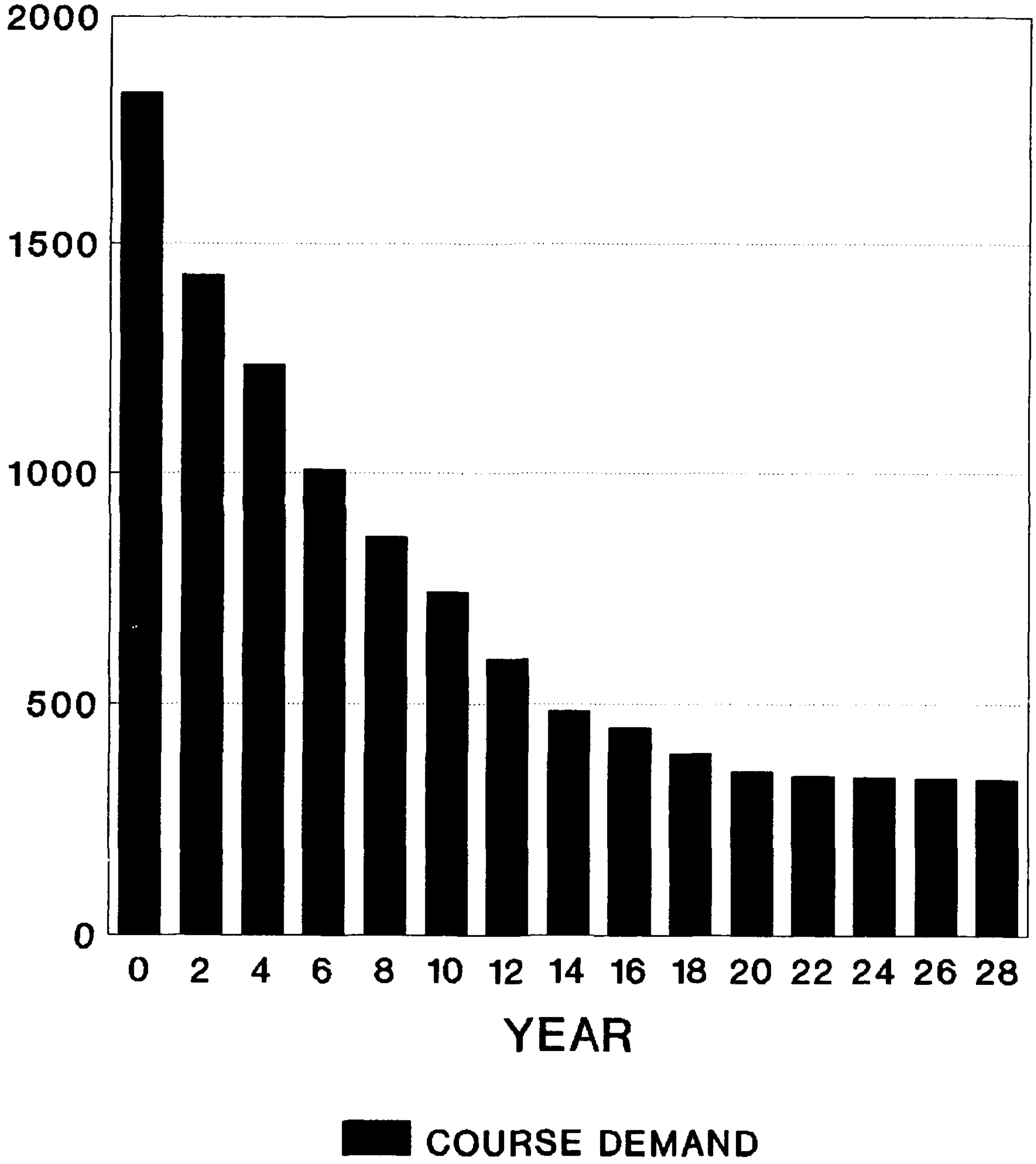


Specialty Course Demand
150% Training Increase
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND

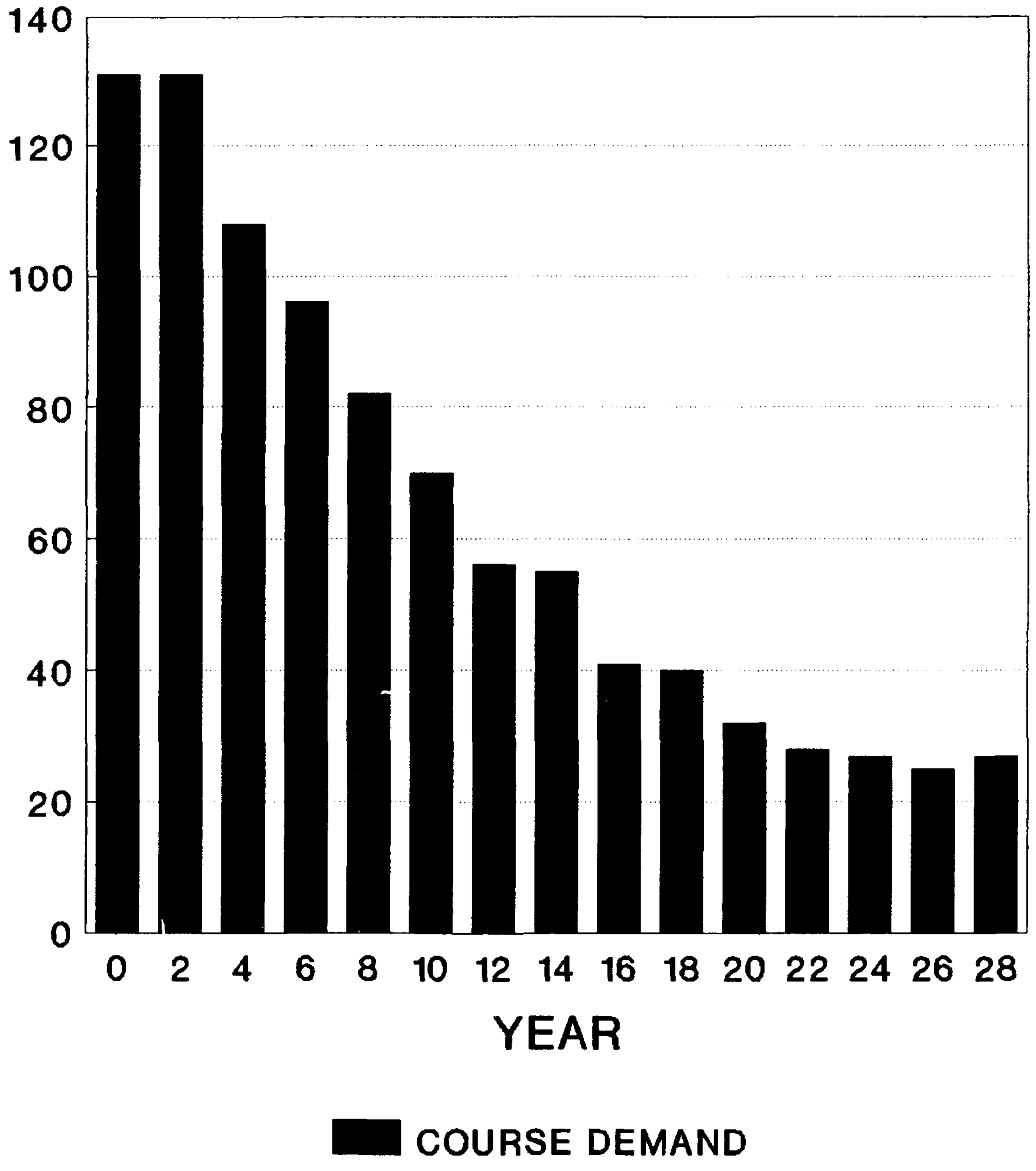


Specialty Course Demand
200% Training Increase
2000
1500
1000
500
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

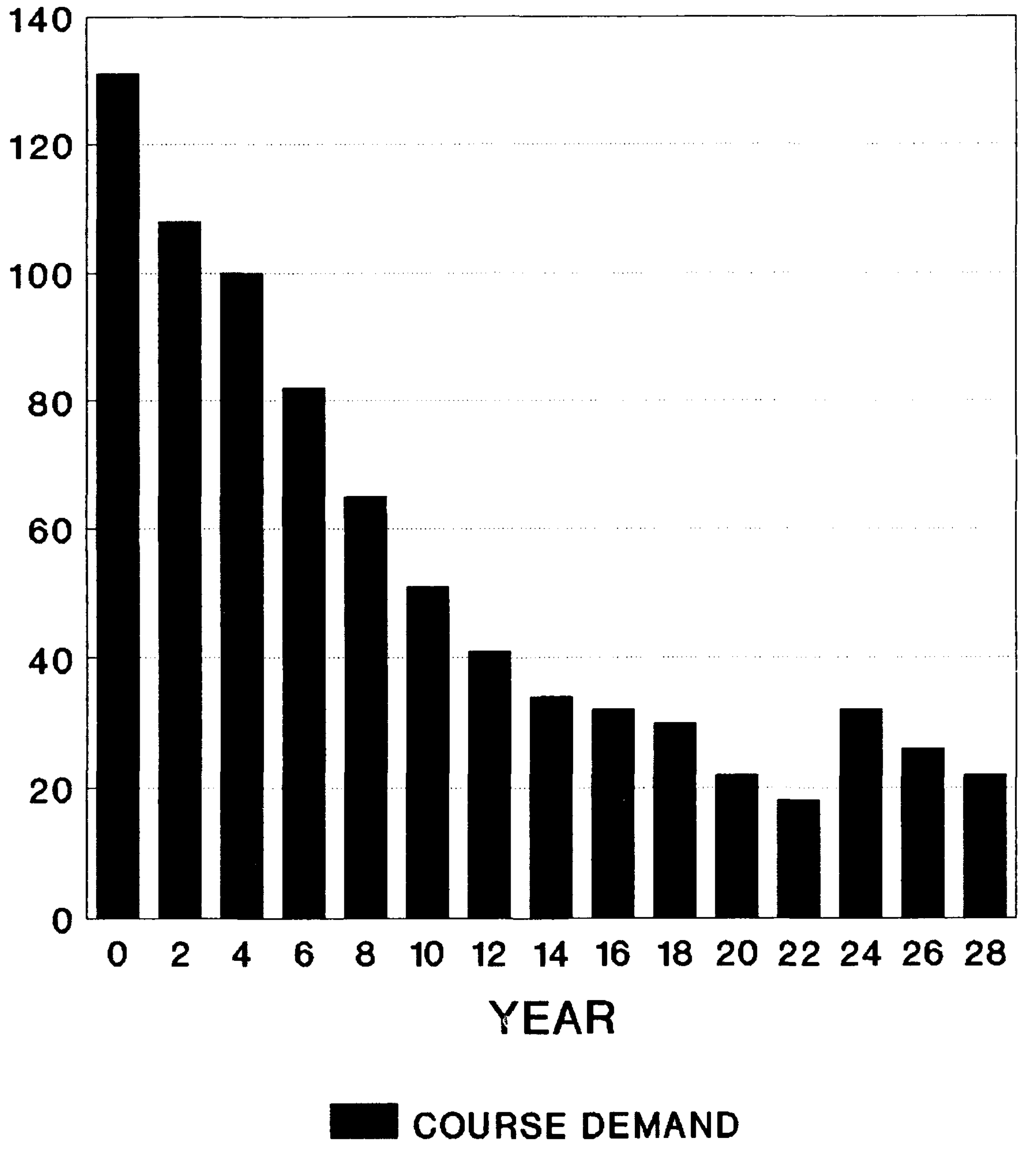
Specialty Course Demand
250% Training Increase



Minimum SYS100 Demand
Based on 1991 Training Capability

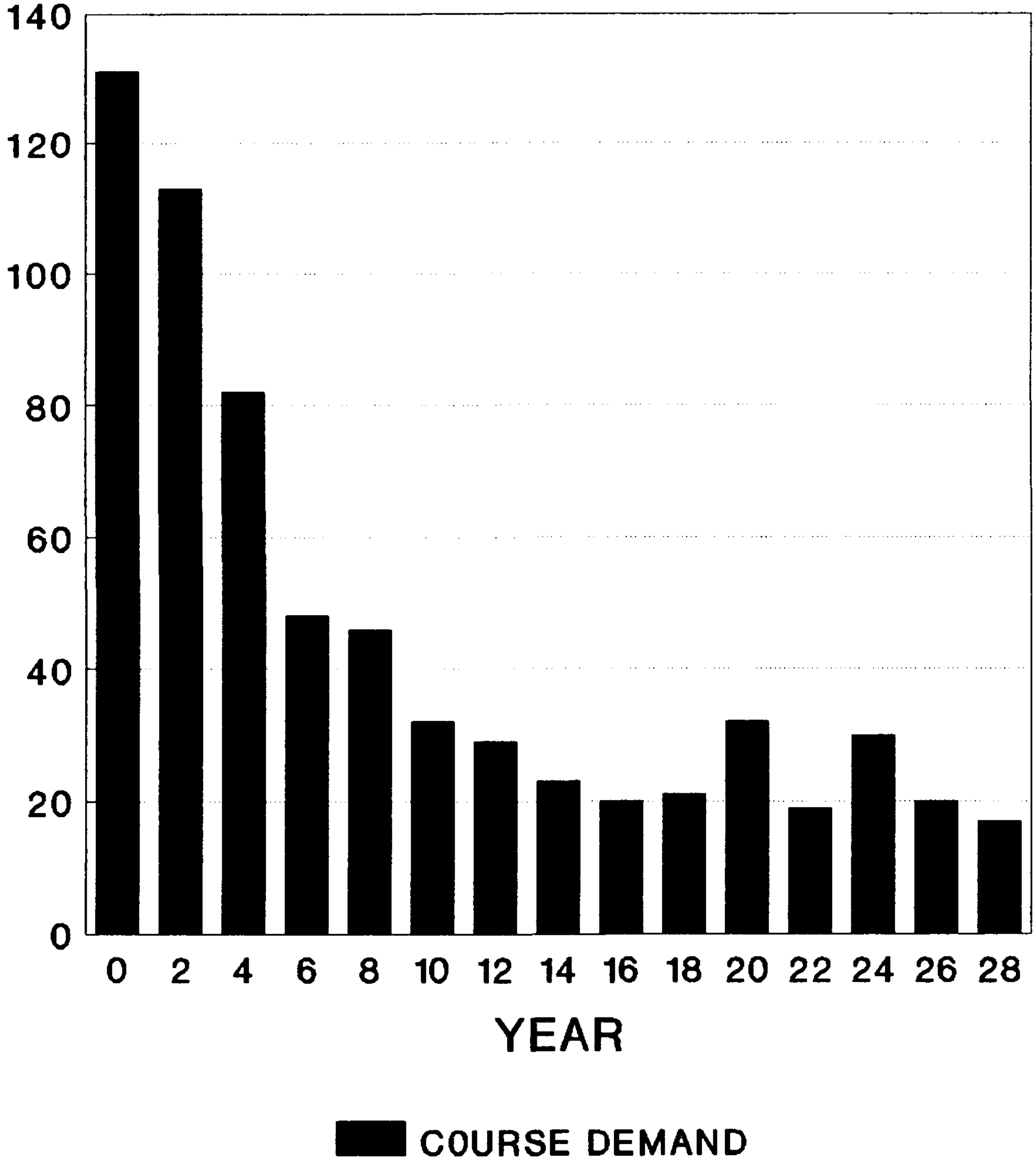


Minimum SYS100 Demand
50% Training Increase



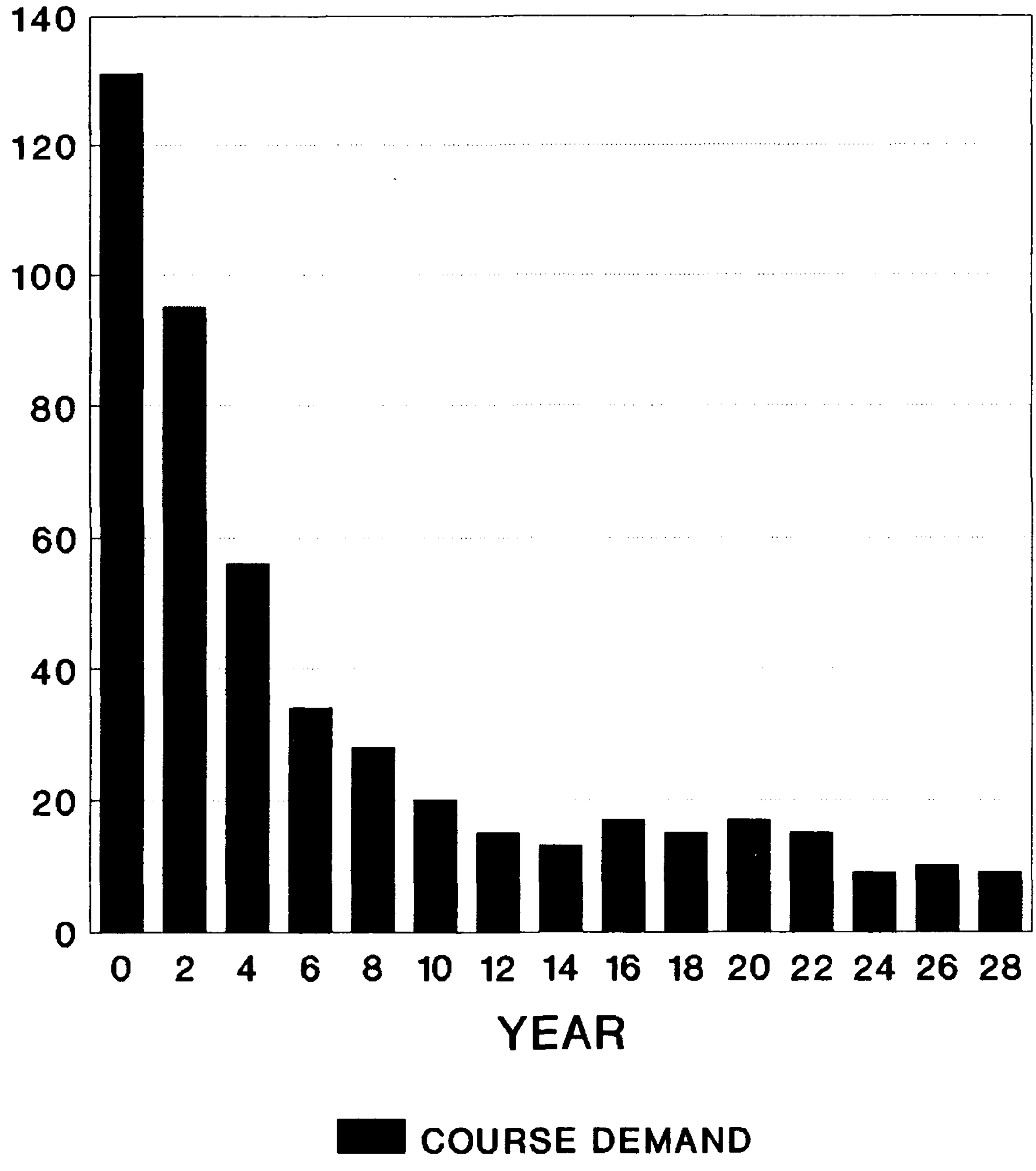
Minimum SYS100 Demand

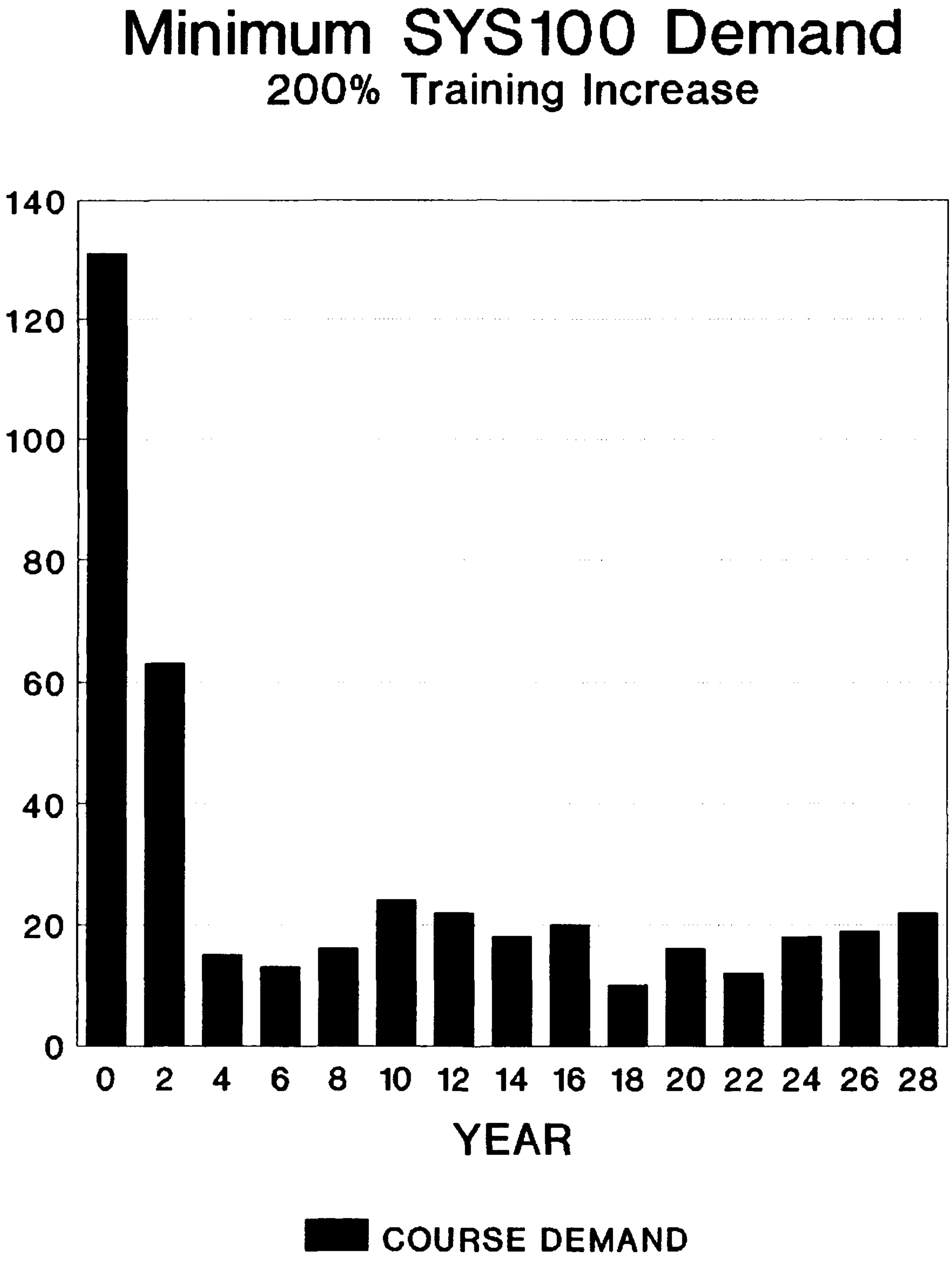
100% Training Increase



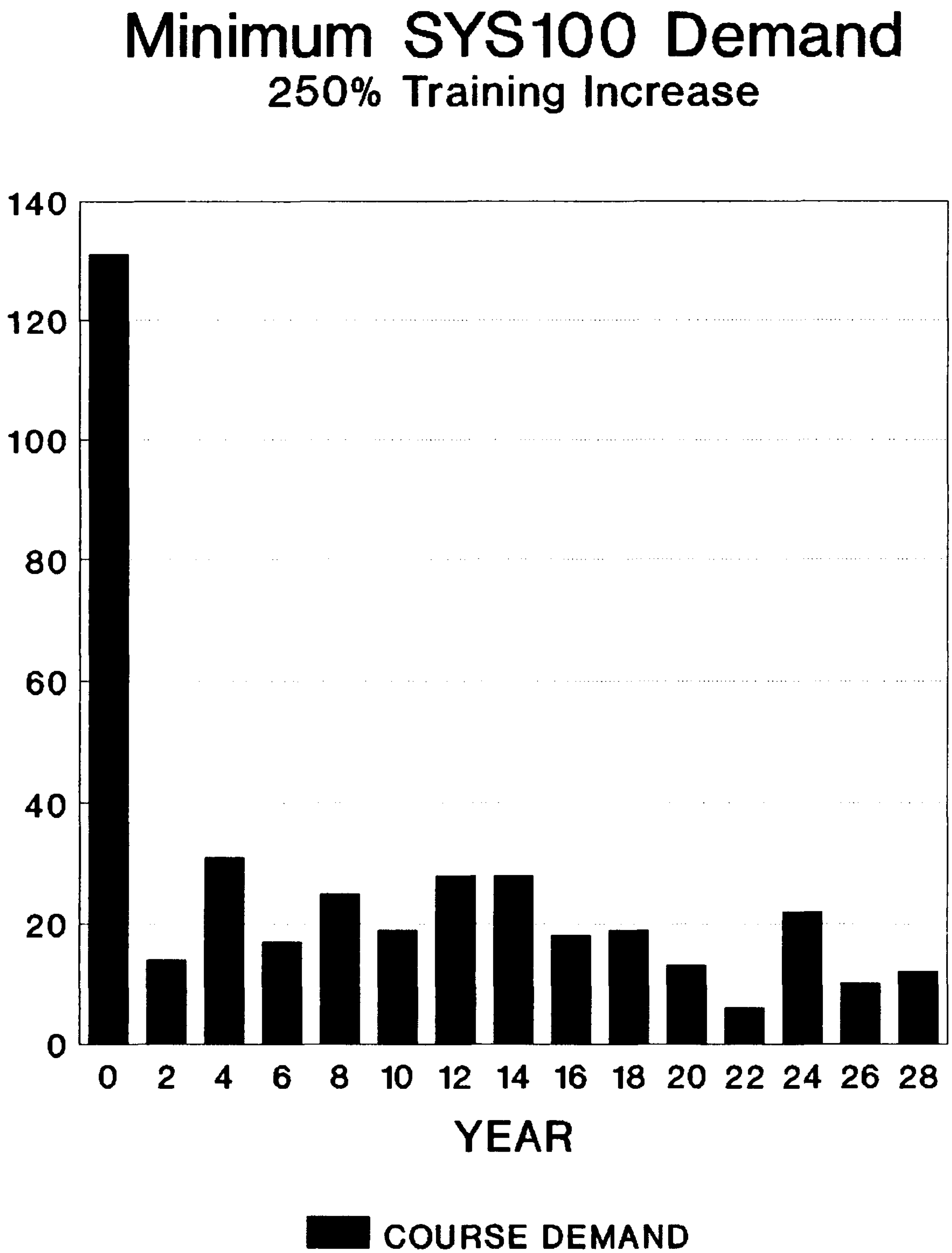
Minimum SYS100 Demand

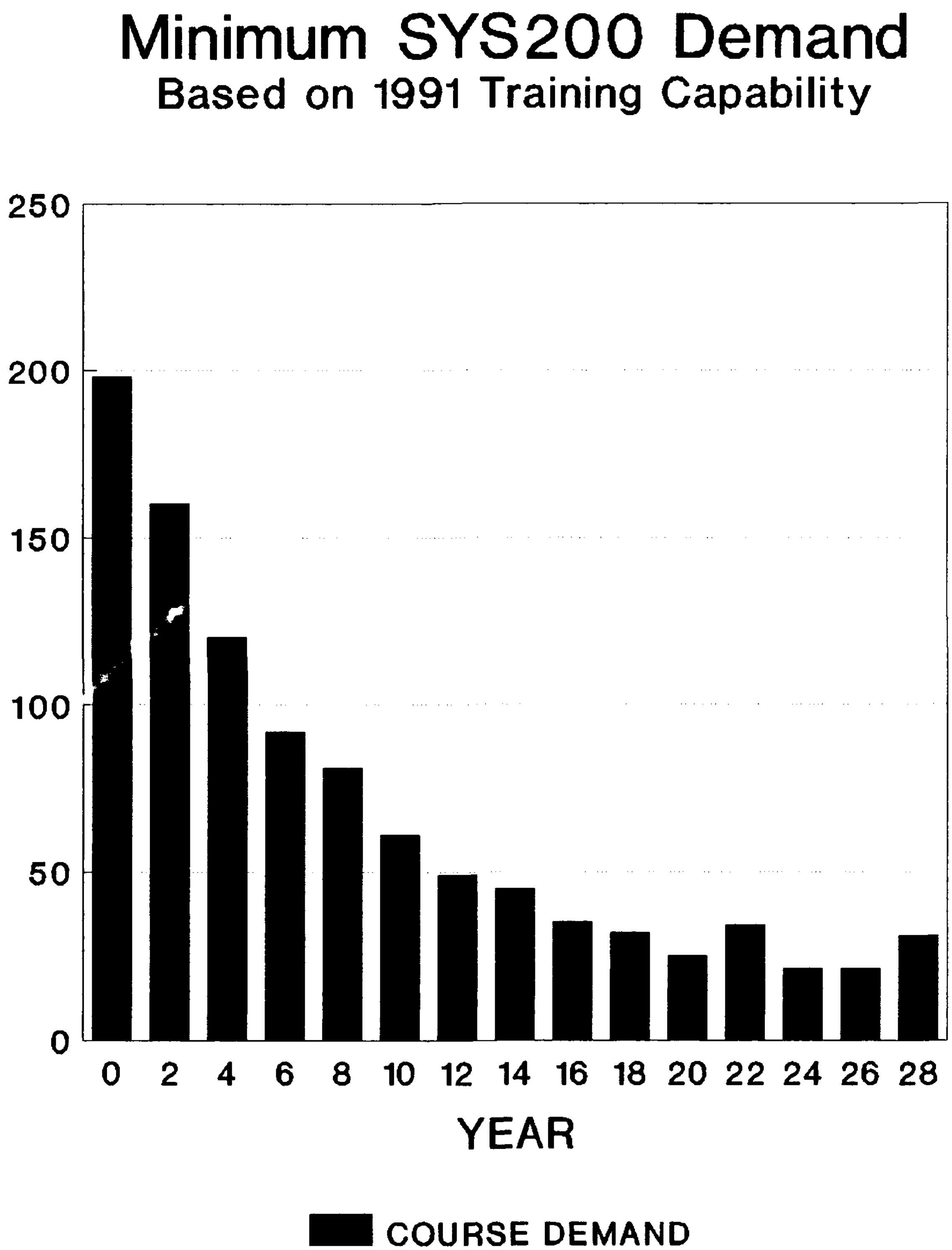
150% Training Increase

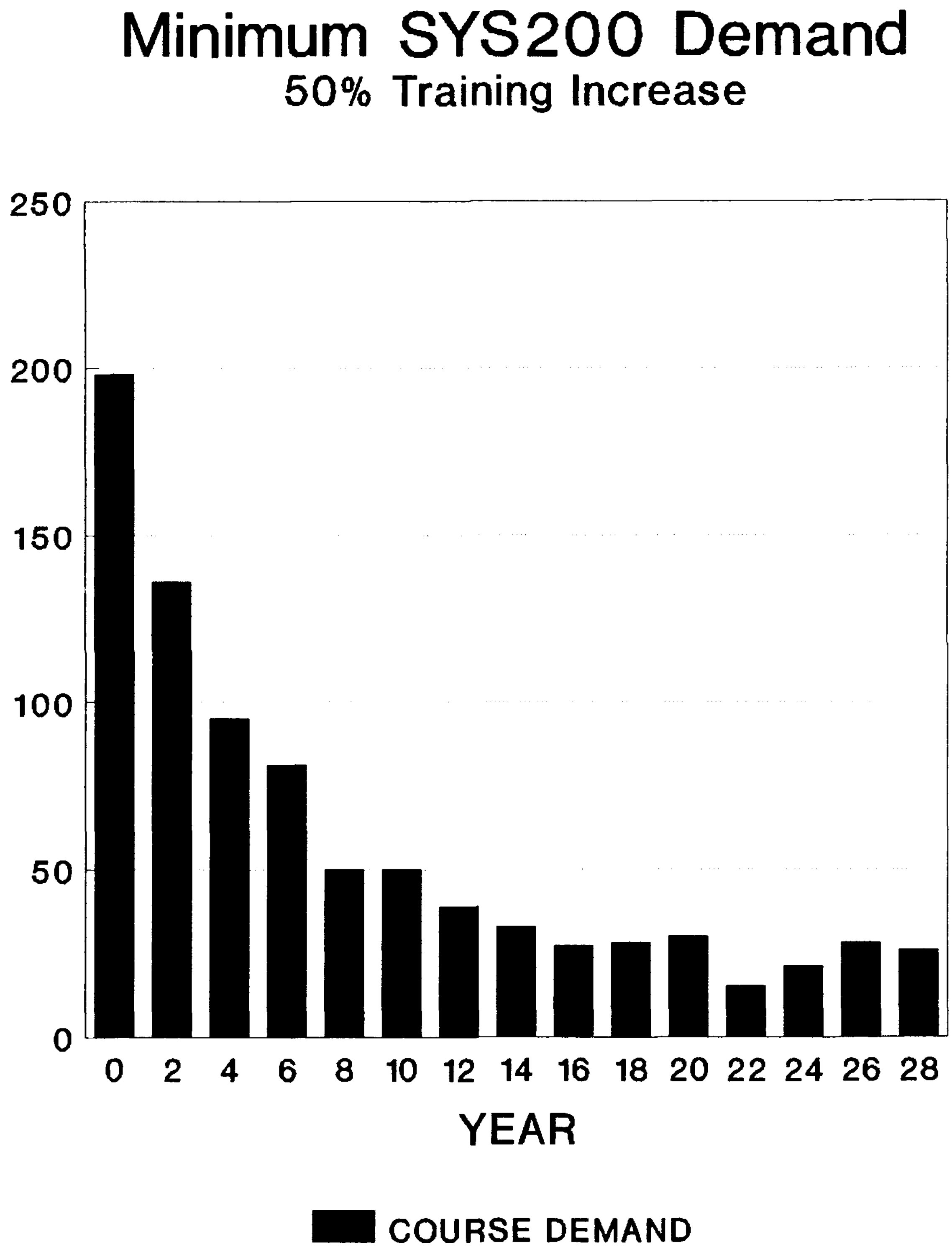




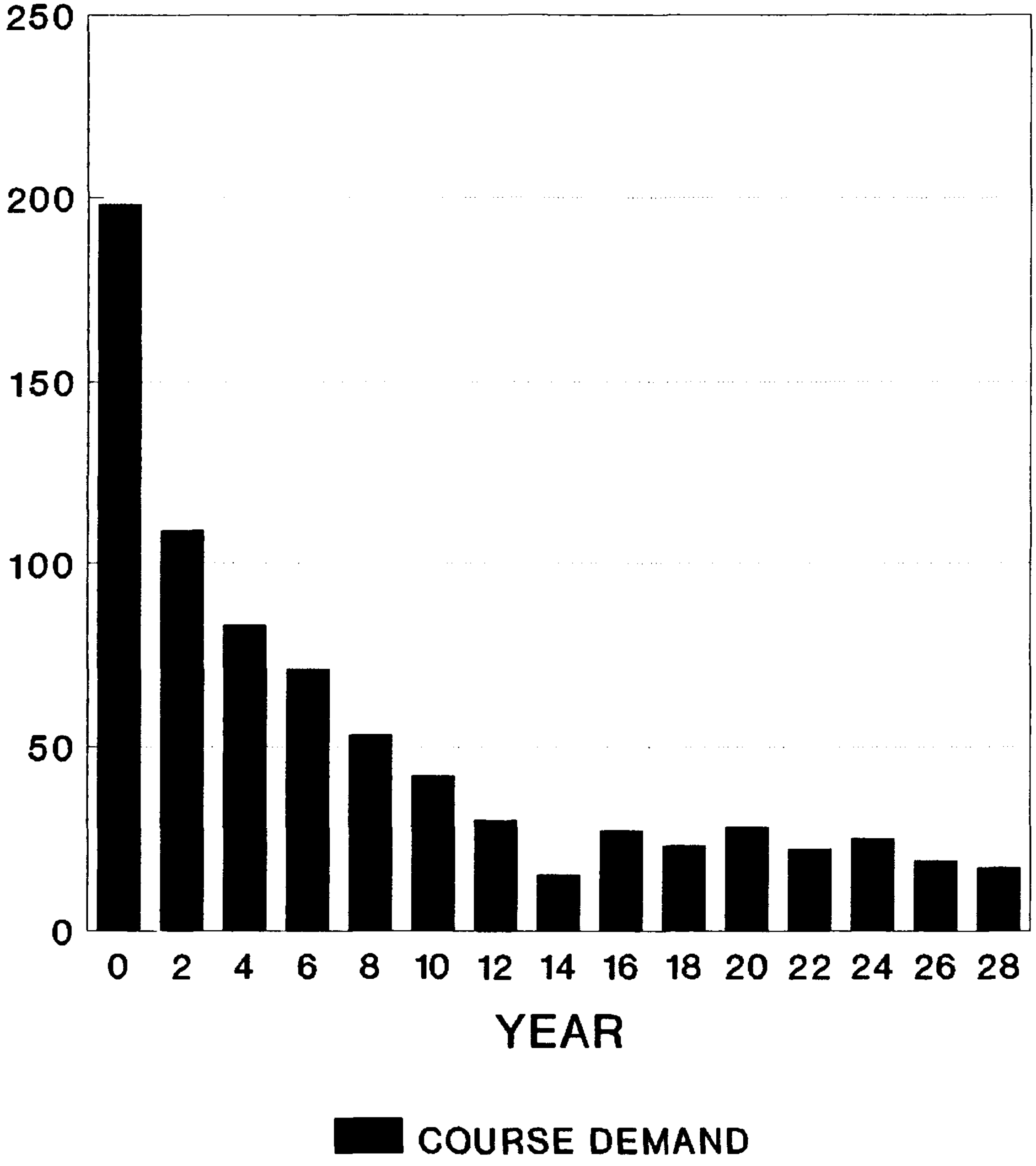
Minimum SYS100 Demand
200% Training Increase
140
120
100
80
60
40
20
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND


Minimum SYS100 Demand
250% Training Increase
140
120
100
80
60
40
20
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND


Minimum SYS200 Demand
Based on 1991 Training Capability
250
200
150
100
50
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

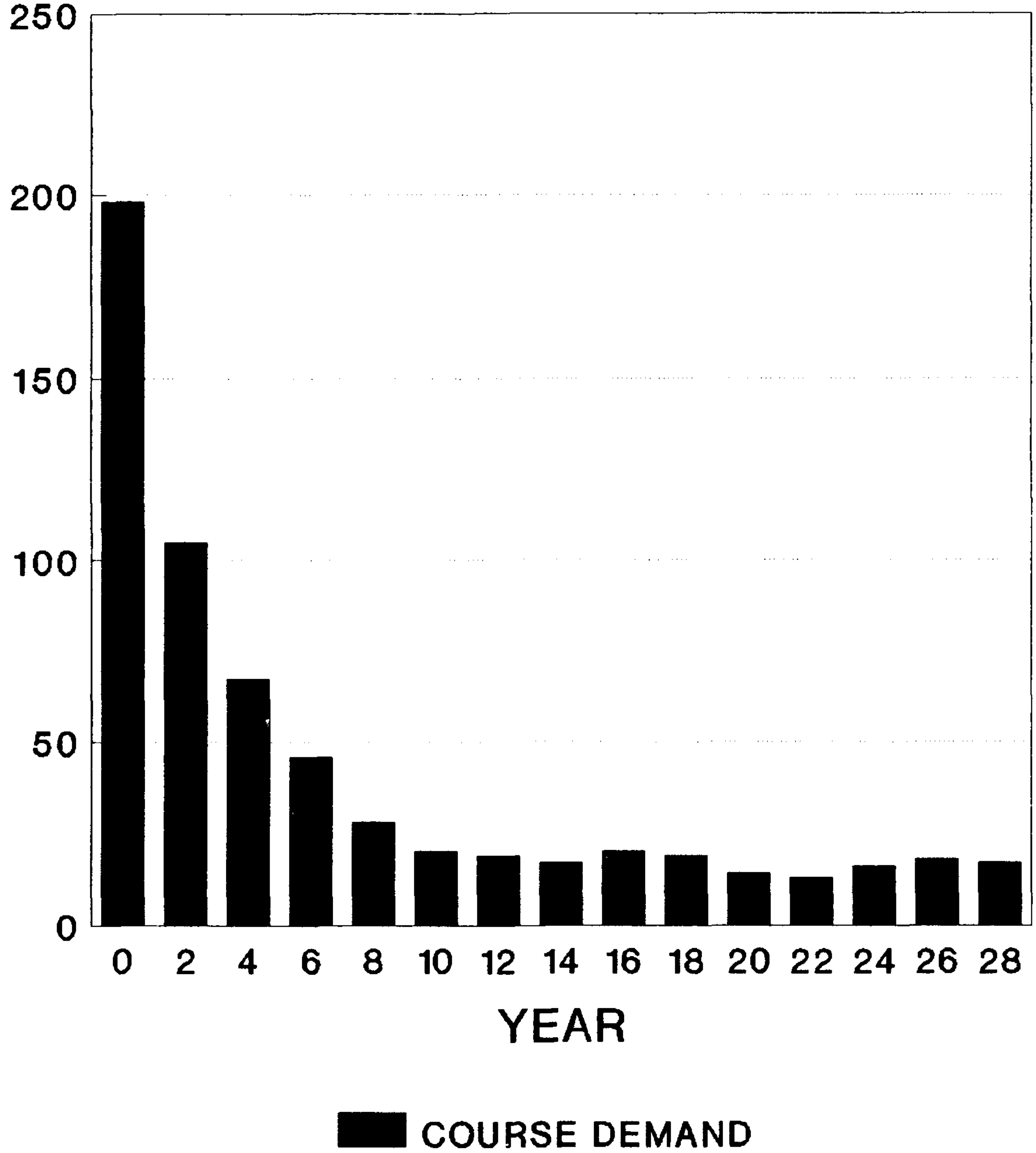

Minimum SYS200 Demand
50% Training Increase
250
200
150
100
50
0
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
YEAR
COURSE DEMAND

Minimum SYS200 Demand
100% Training Increase

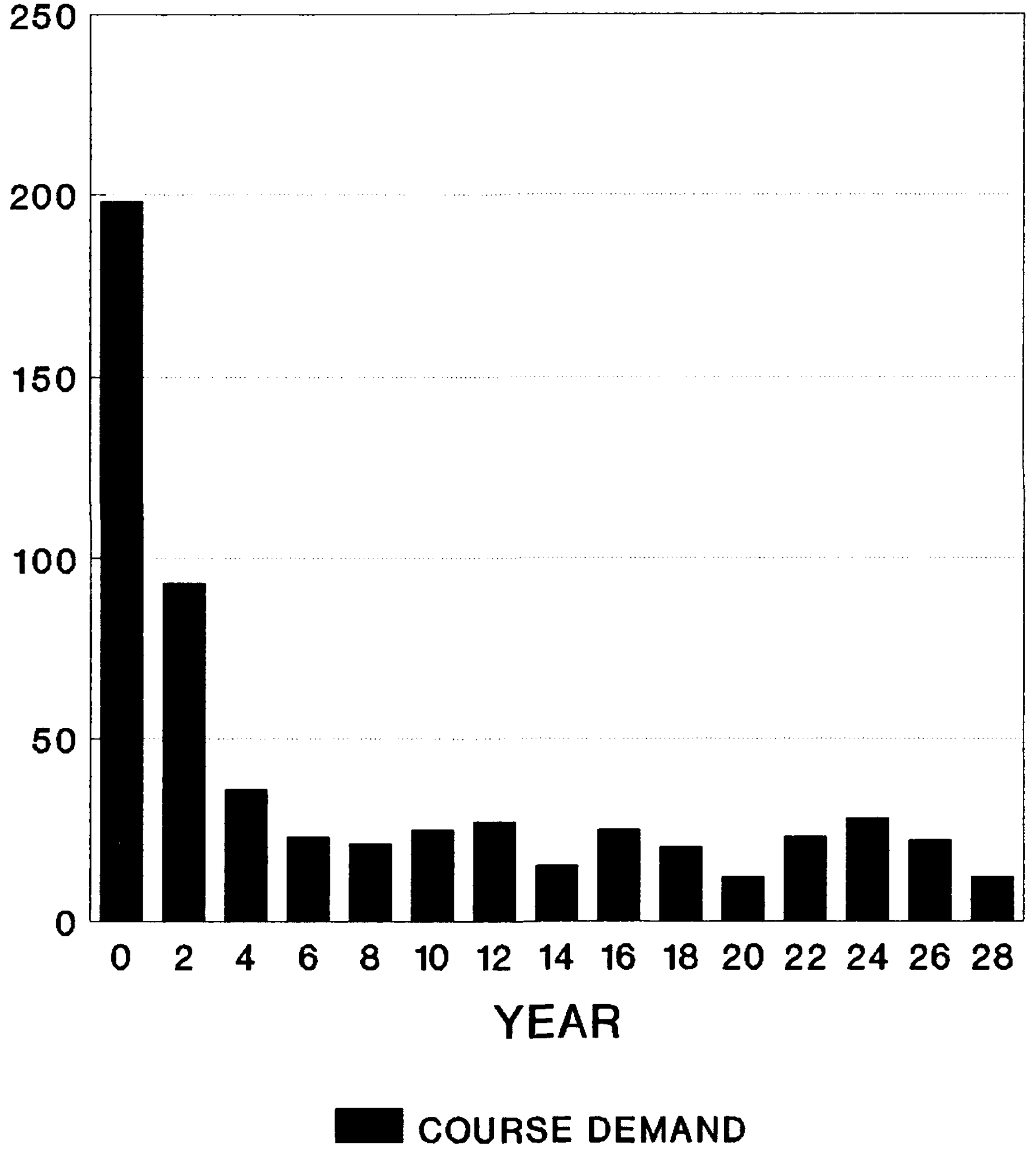


Minimum SYS200 Demand

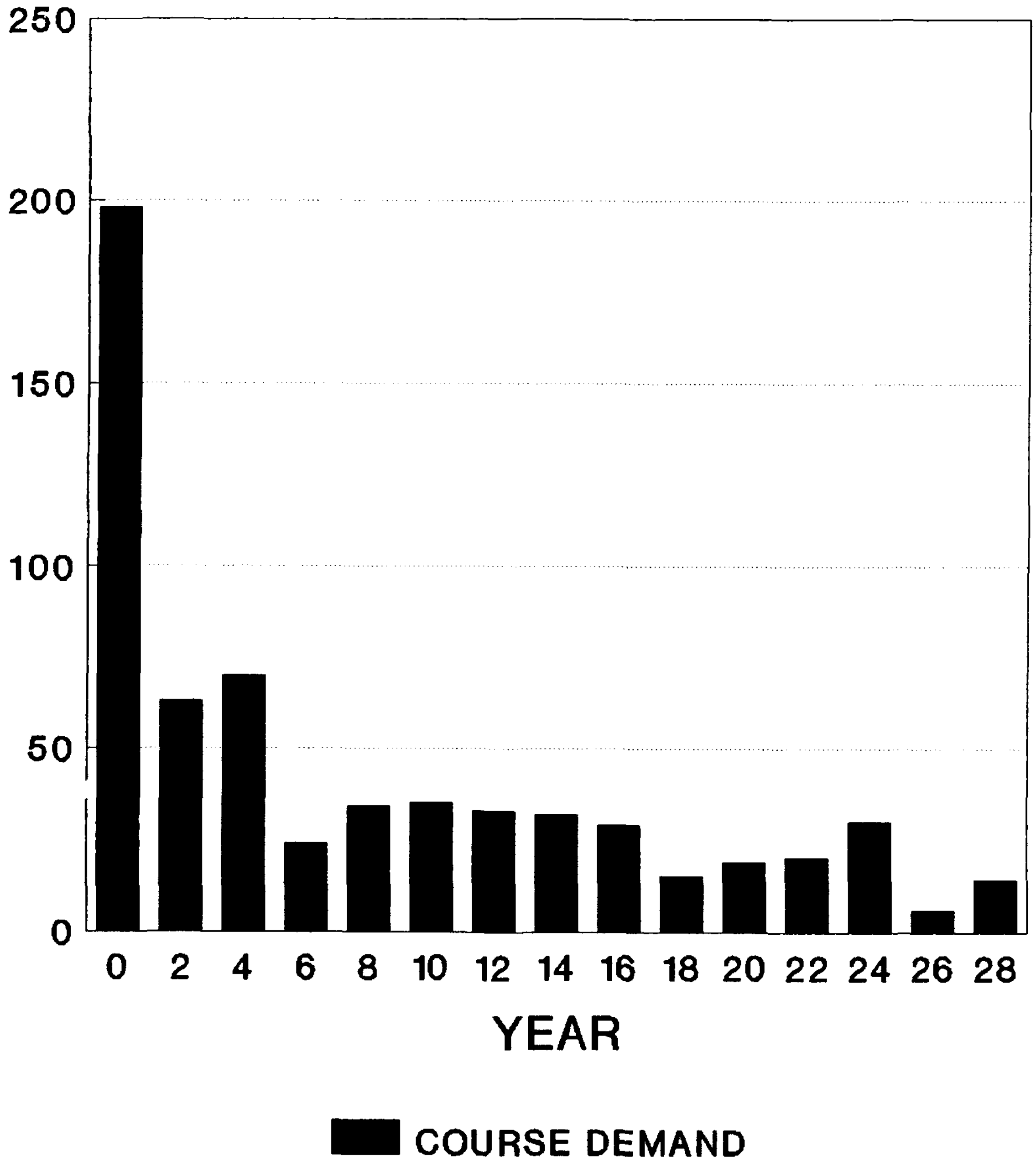
150% Training Increase



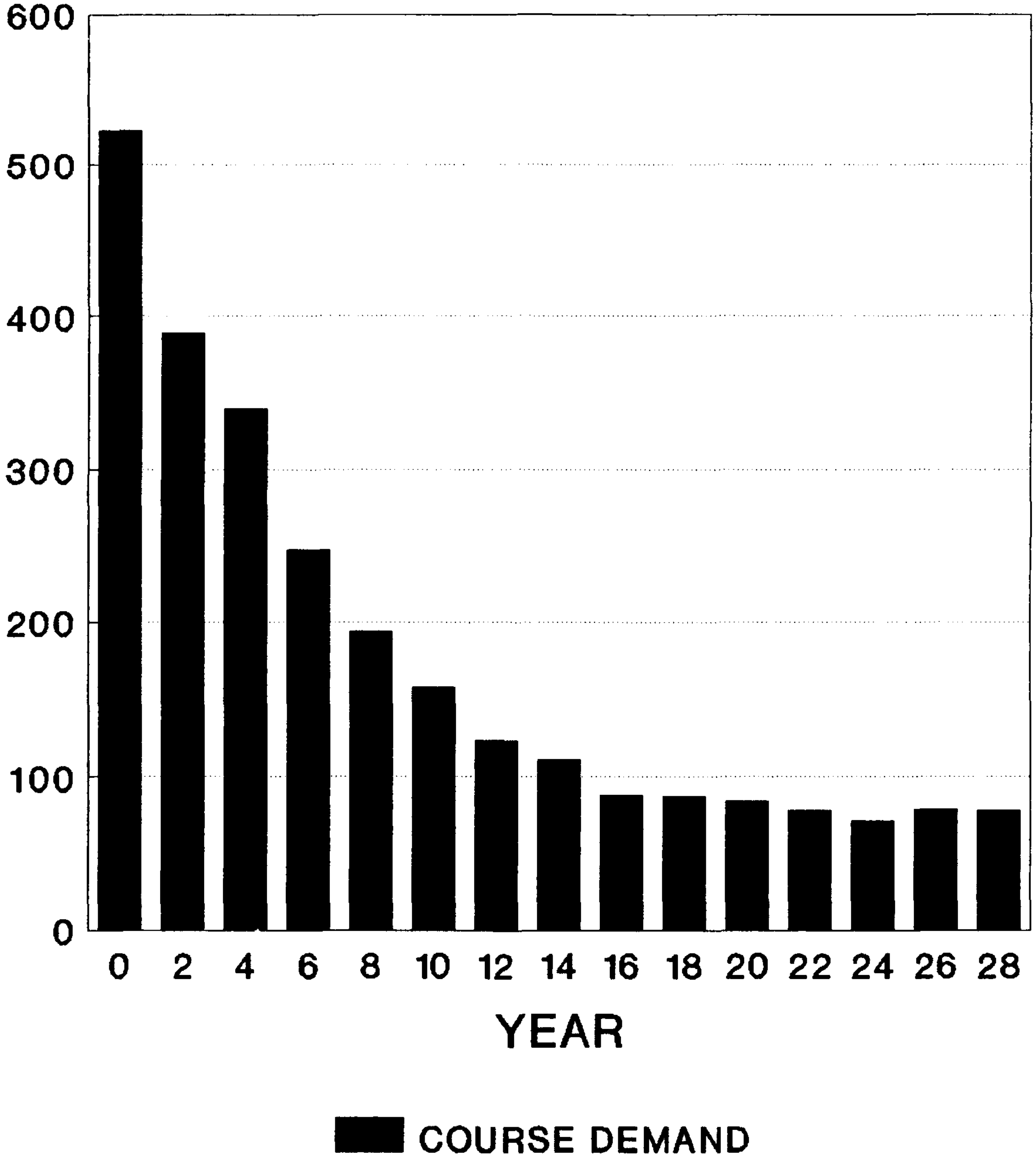
Minimum SYS200 Demand
200% Training Increase



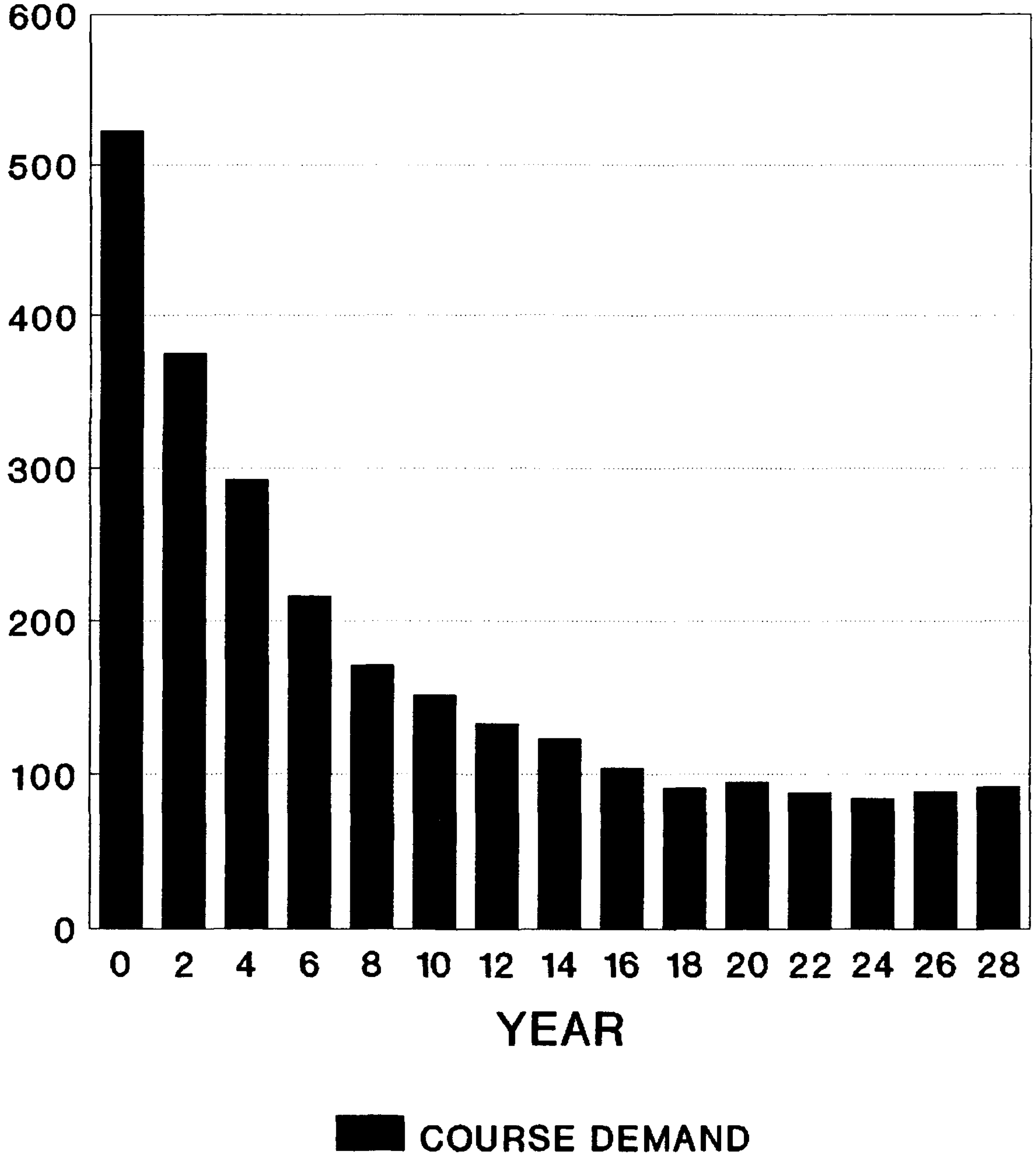
Minimum SYS200 Demand
250% Training Increase



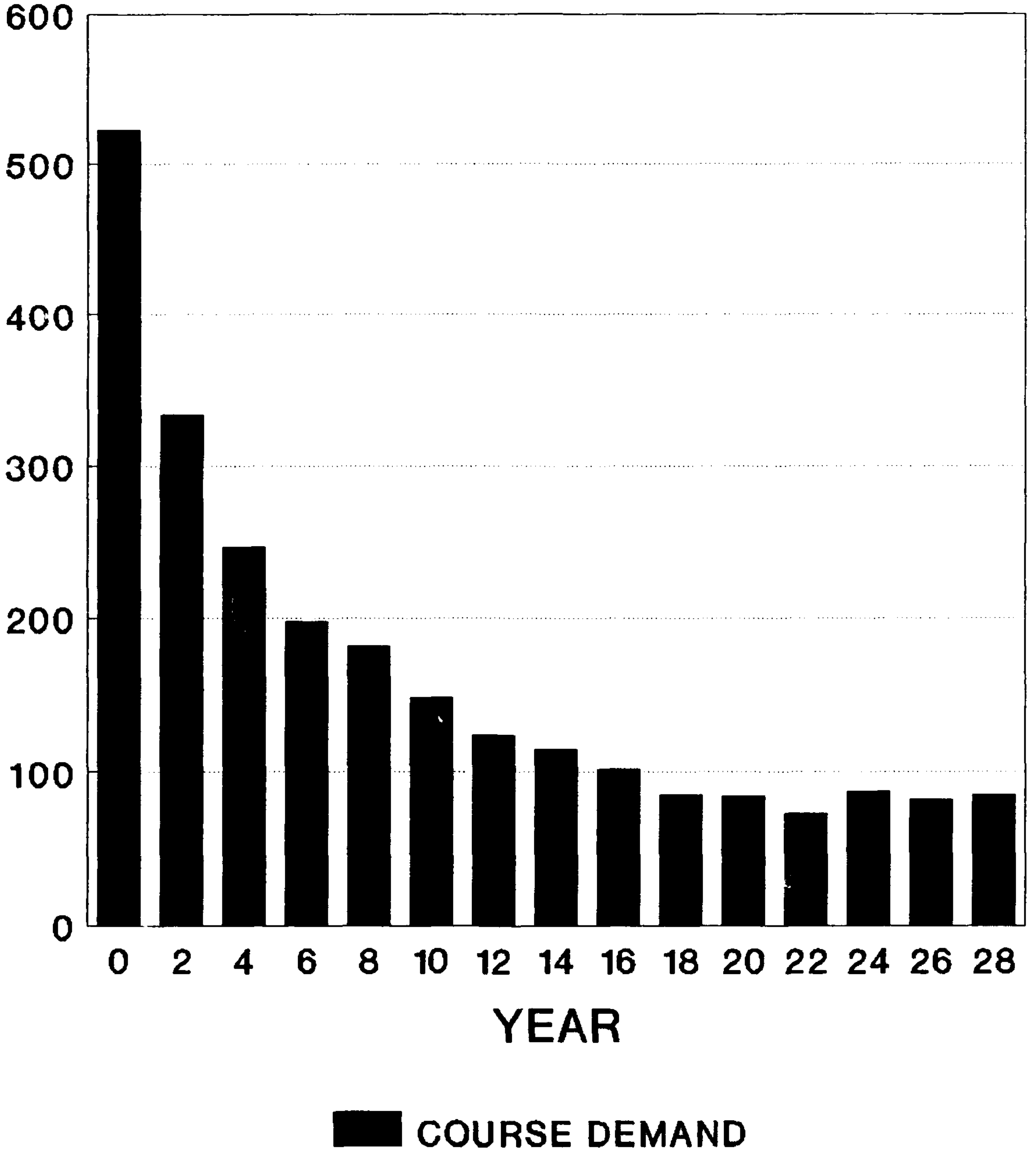
Minimum SYS225 Demand
Based on 1991 Training Capability

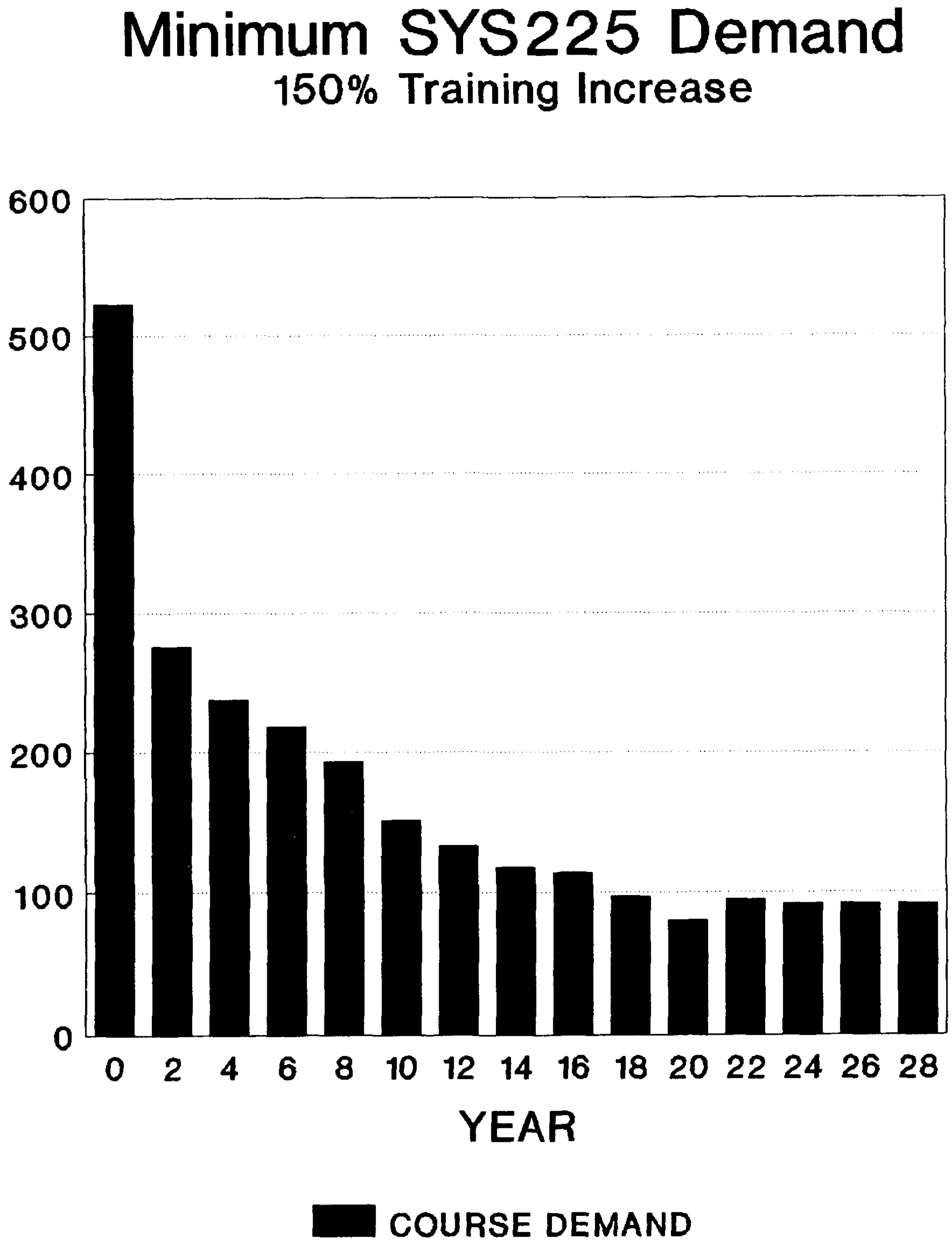


Minimum SYS225 Demand
50% Training Increase

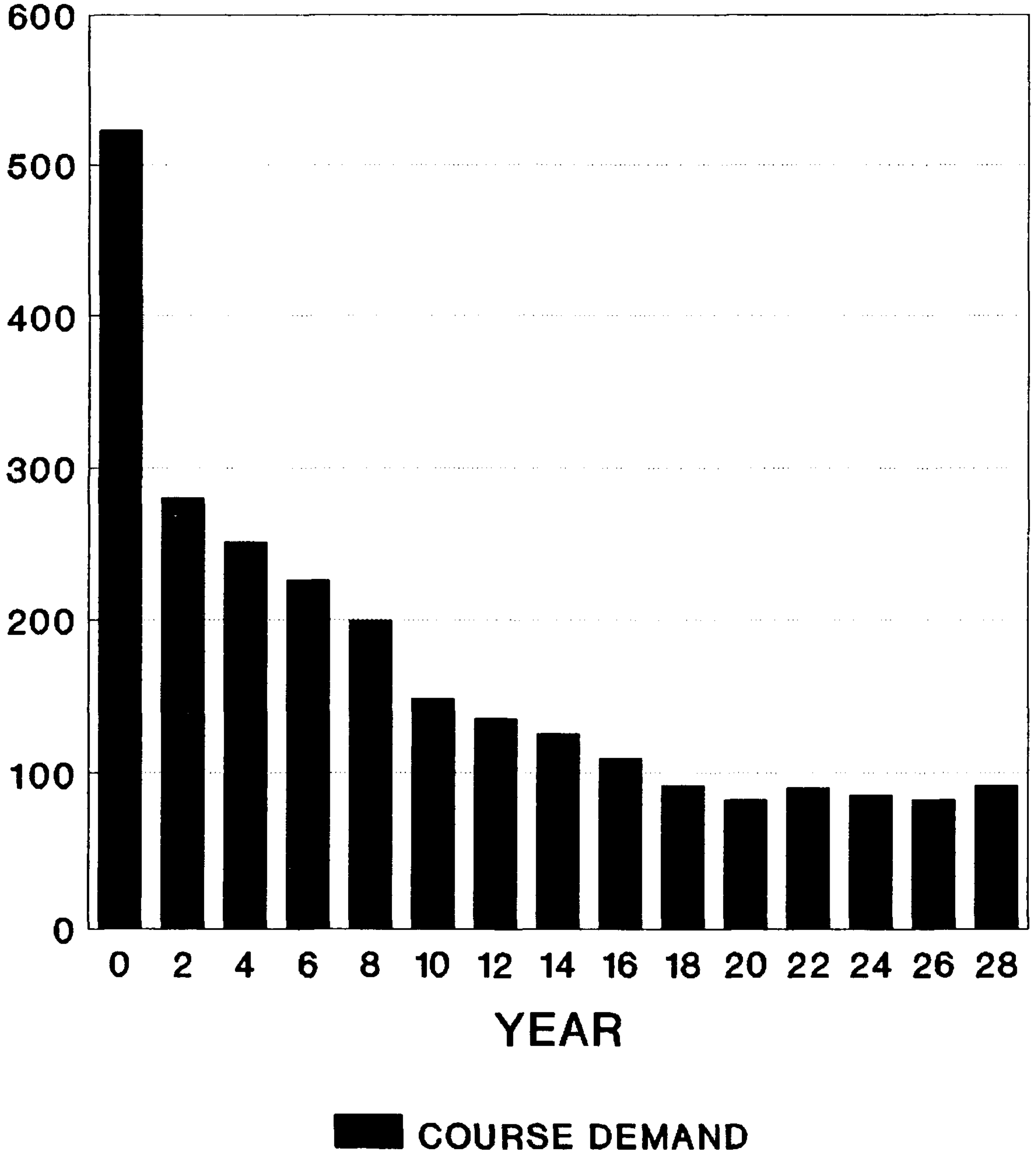


Minimum SYS225 Demand
100% Training Increase

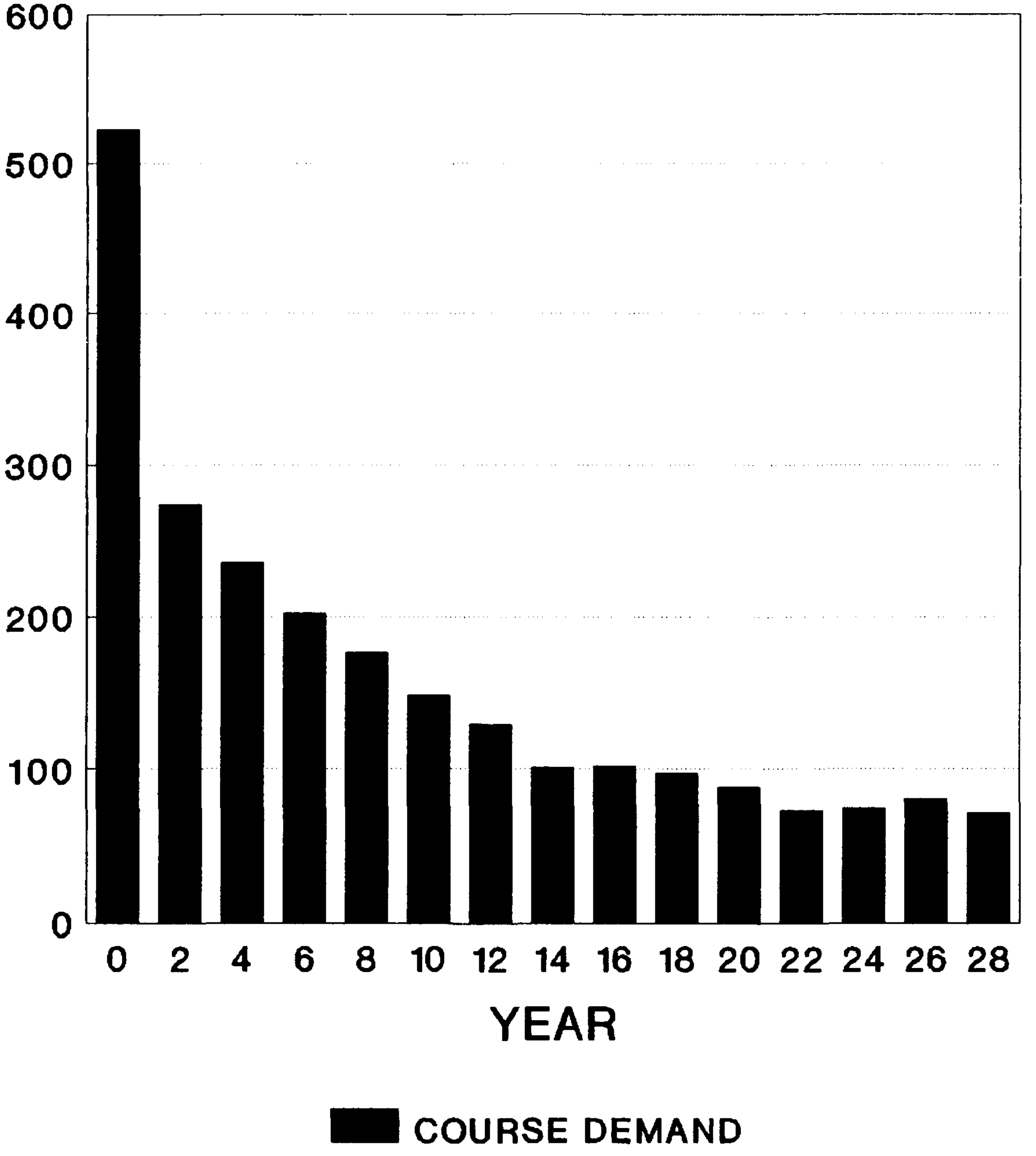



Minimum SYS225 Demand
150% Training Increase
600
500
400
300
200
100
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
YEAR
COURSE DEMAND

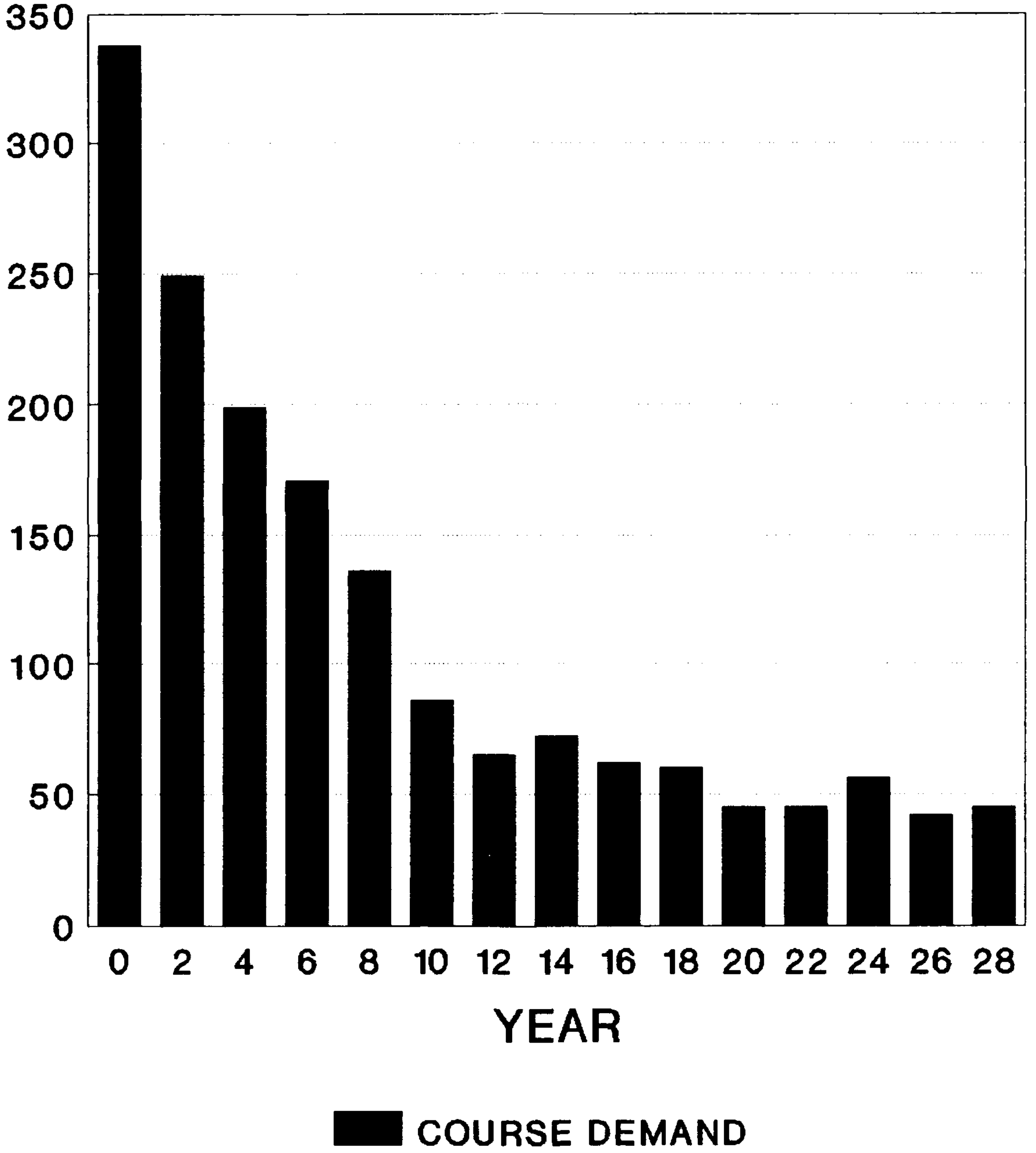
Minimum SYS225 Demand
200% Training Increase



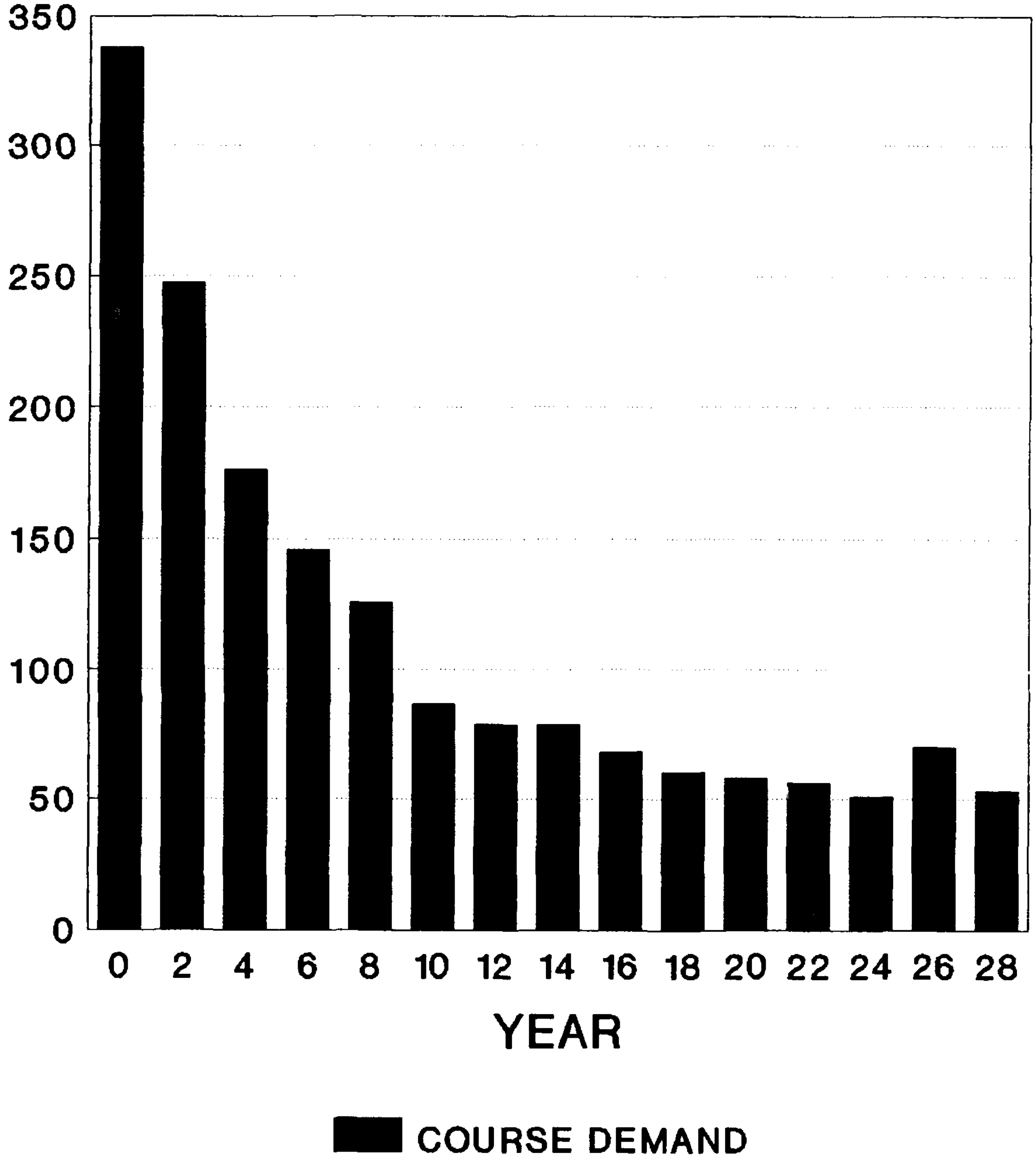
Minimum SYS225 Demand
250% Training Increase



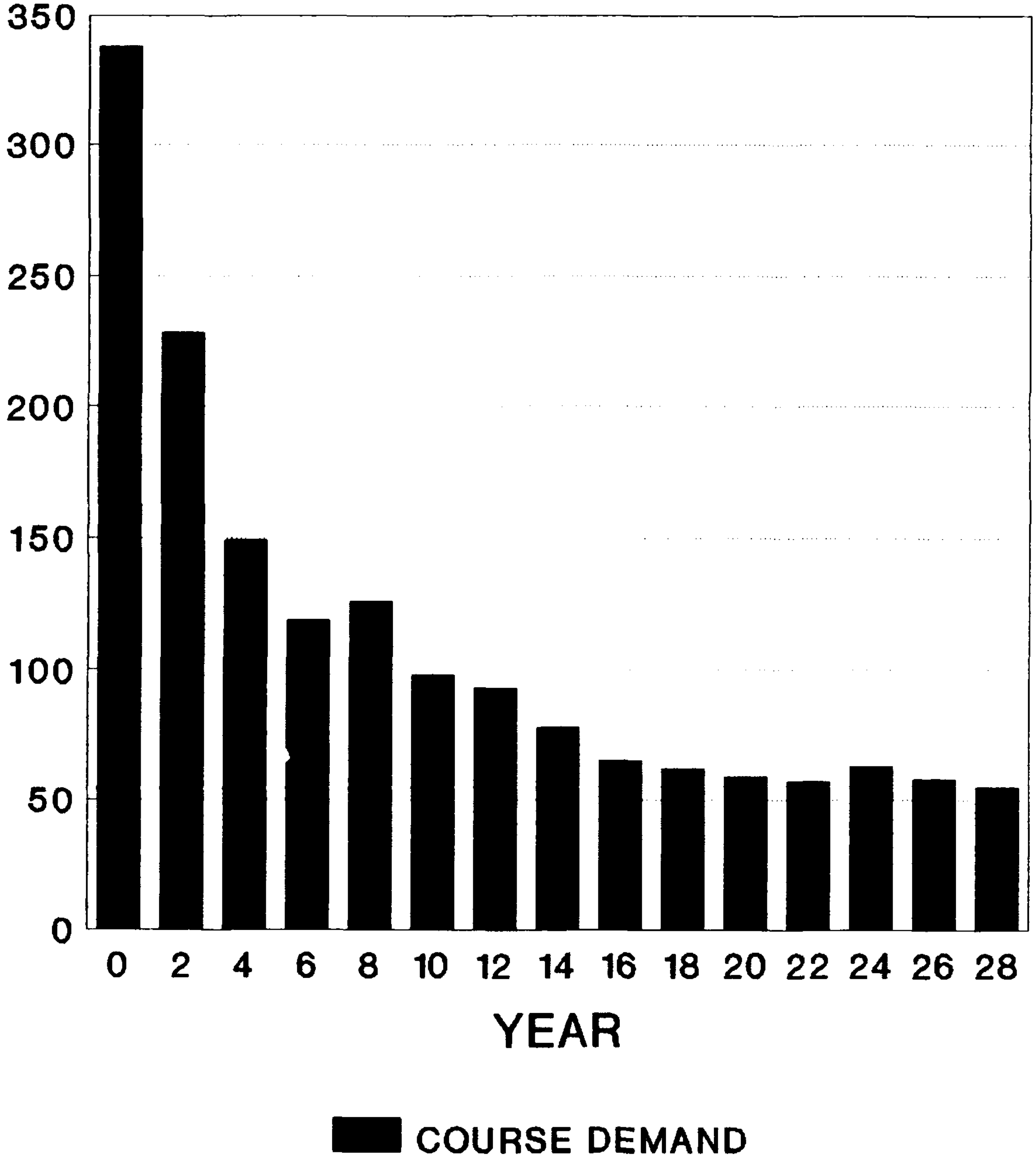
Minimum SYS400 Demand
Based on 1991 Training Capability



Minimum SYS400 Demand
50% Training Increase

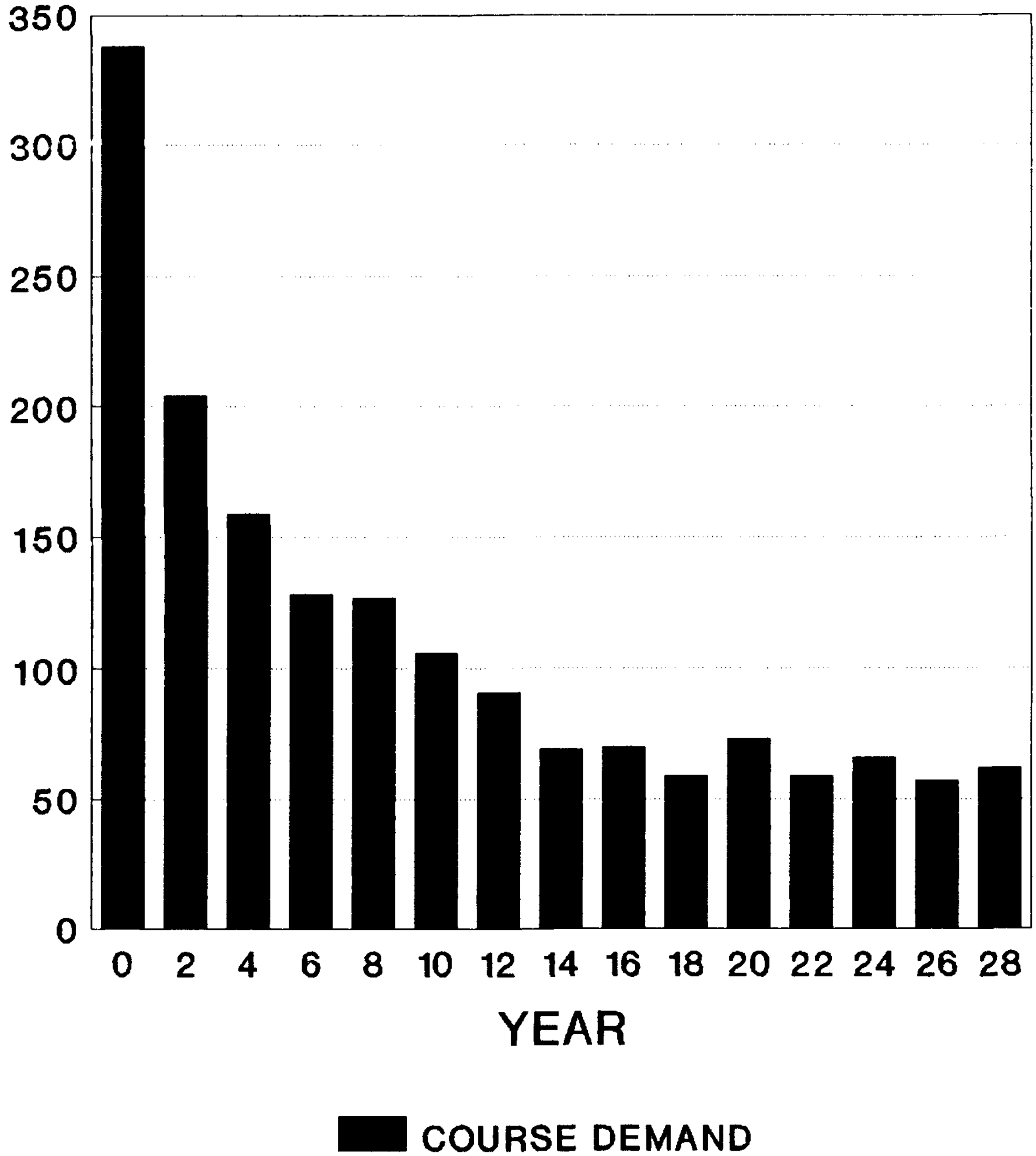


Minimum SYS400 Demand
100% Training Increase



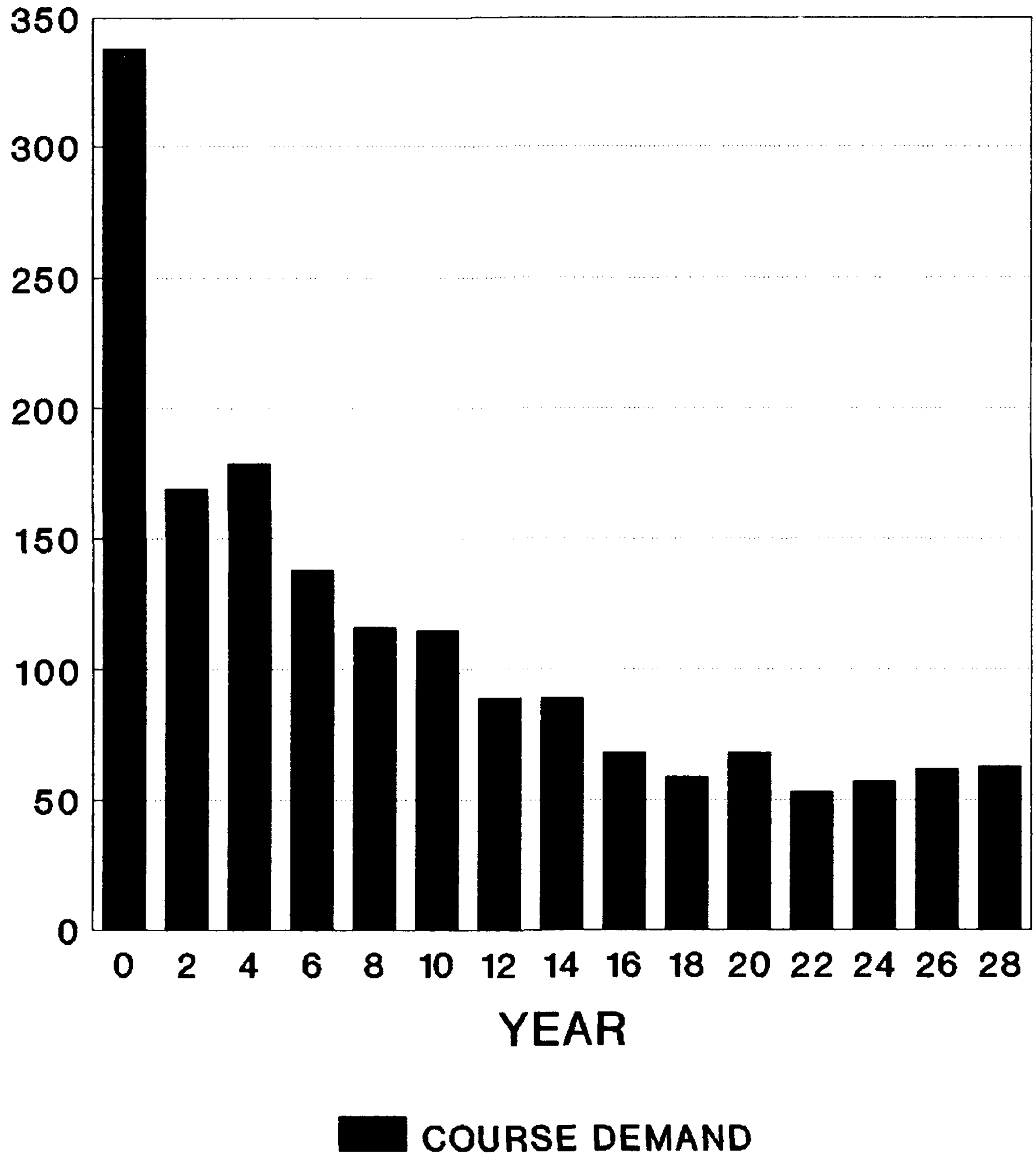
Minimum SYS400 Demand

150% Training Increase

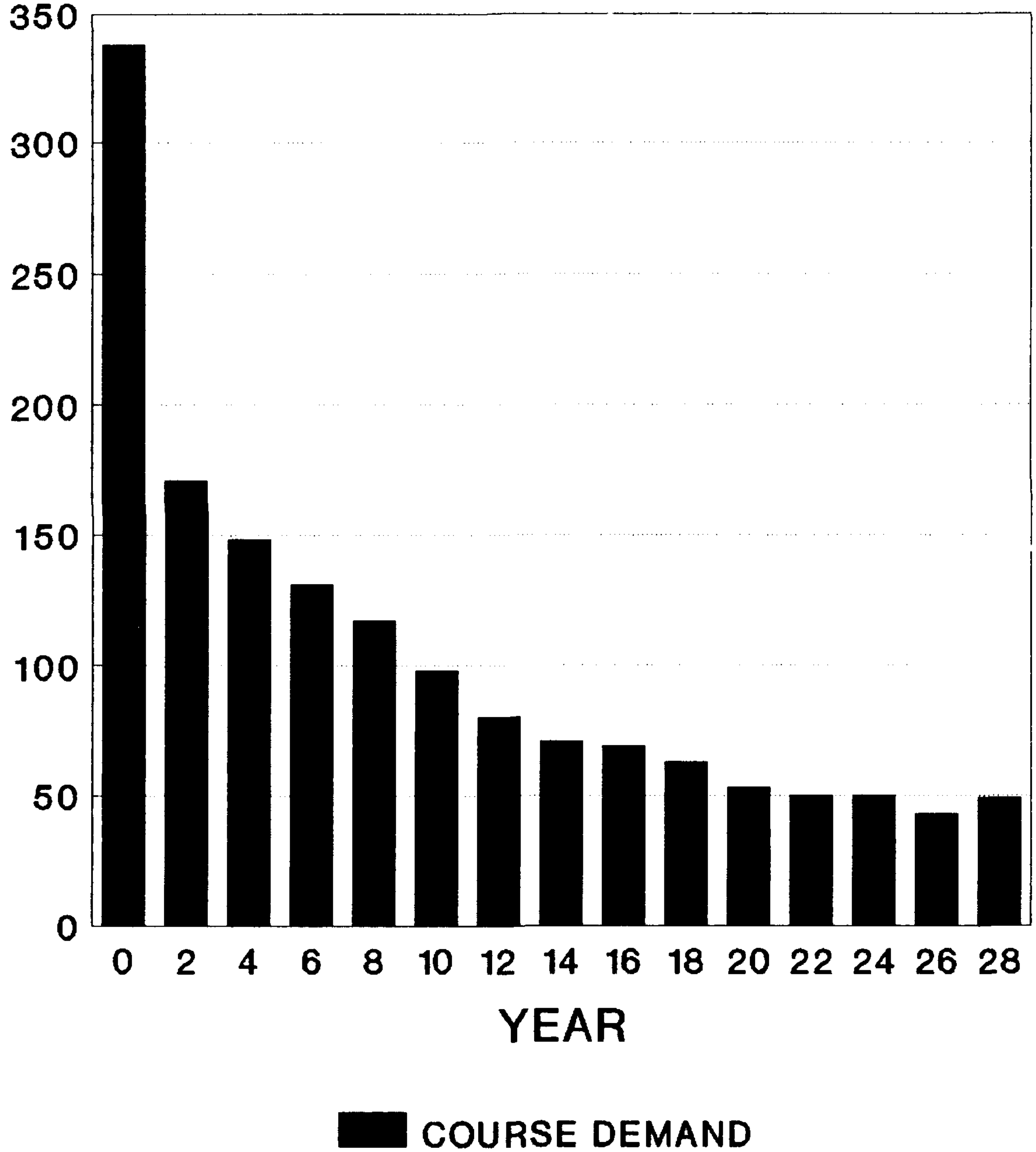


Minimum SYS400 Demand

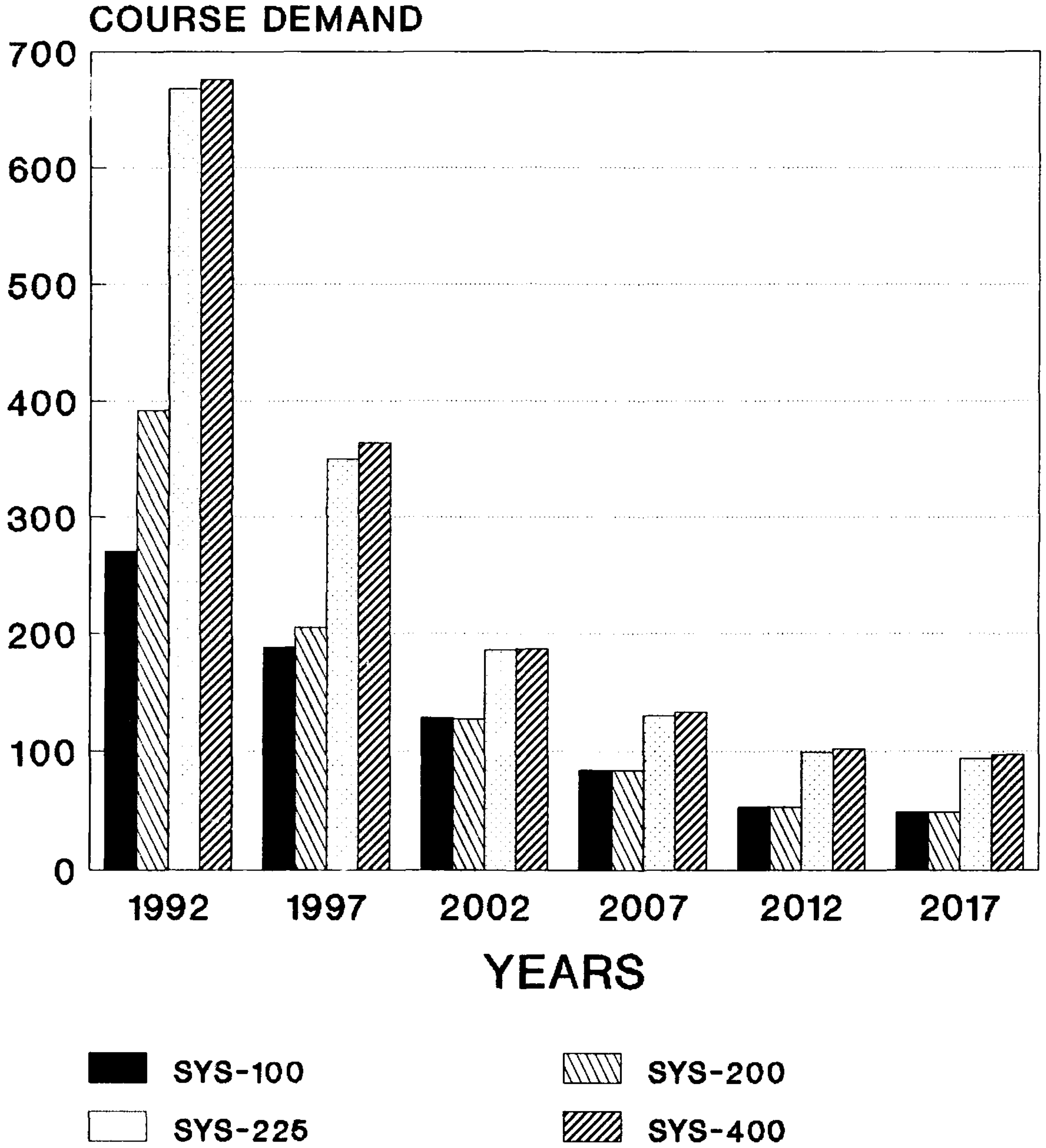
200% Training Increase



Minimum SYS400 Demand
250% Training Increase



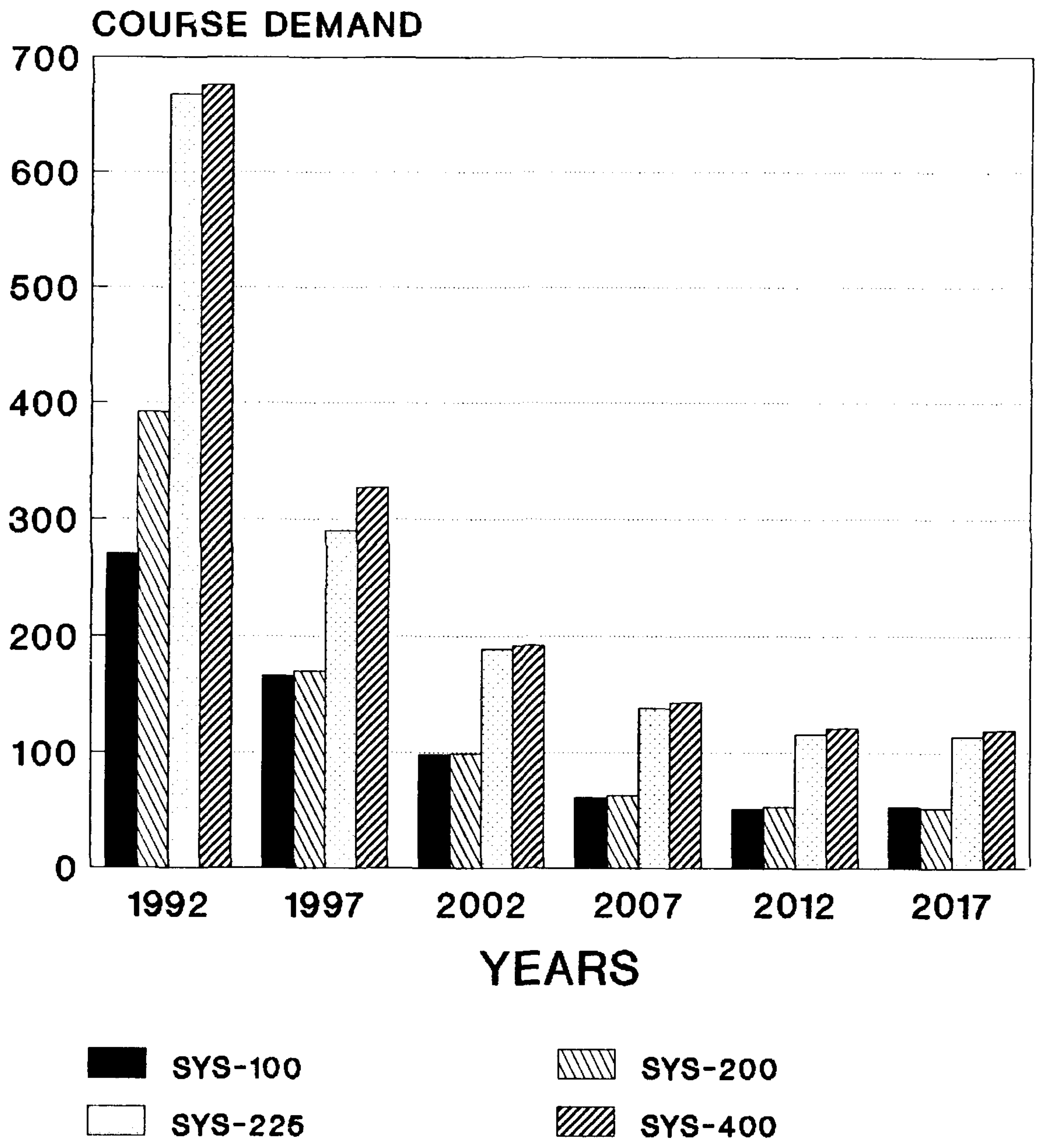
Maximum Course Demands
Based on 1991 Training Capability



OW

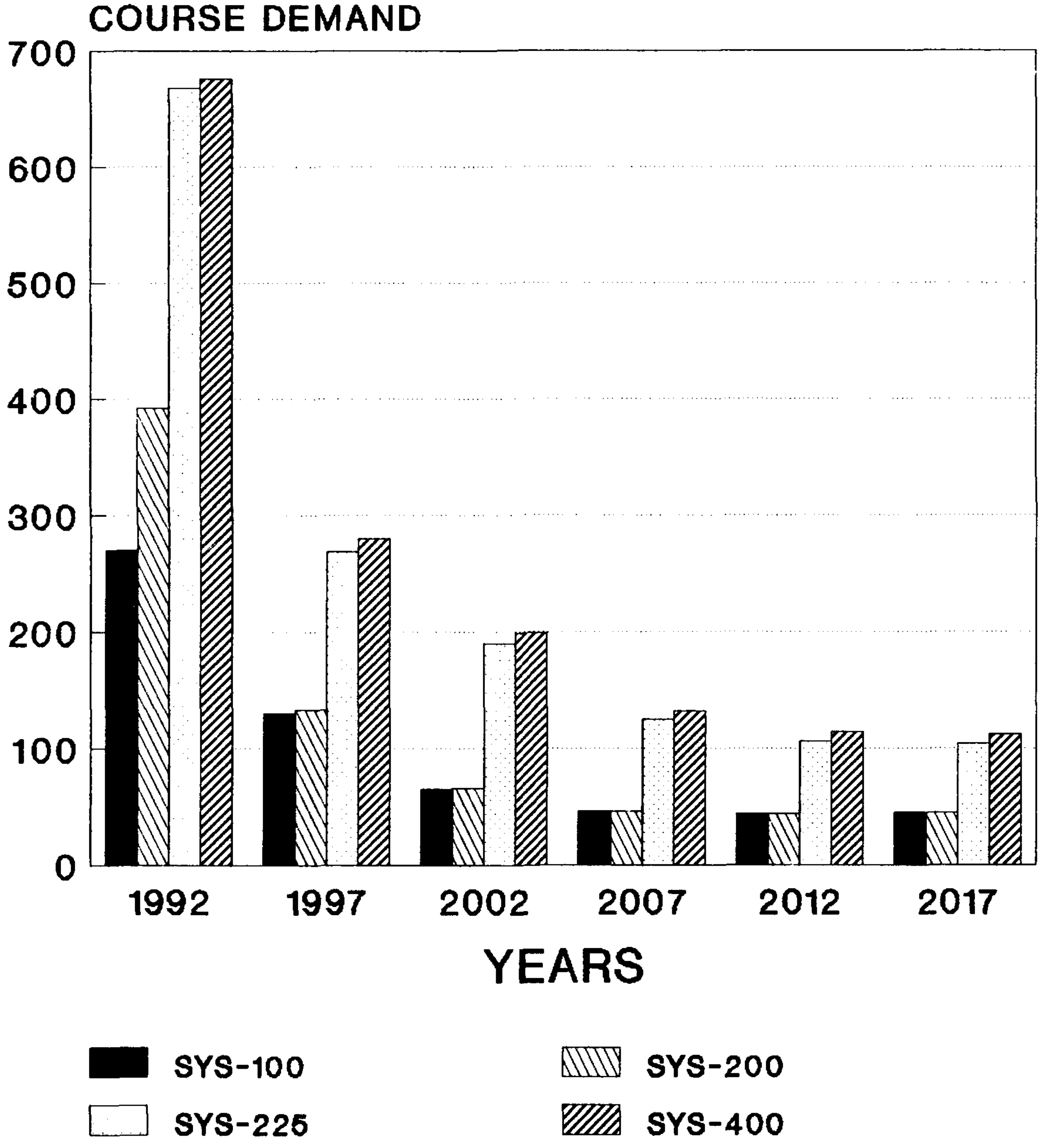
Maximum Course Demands

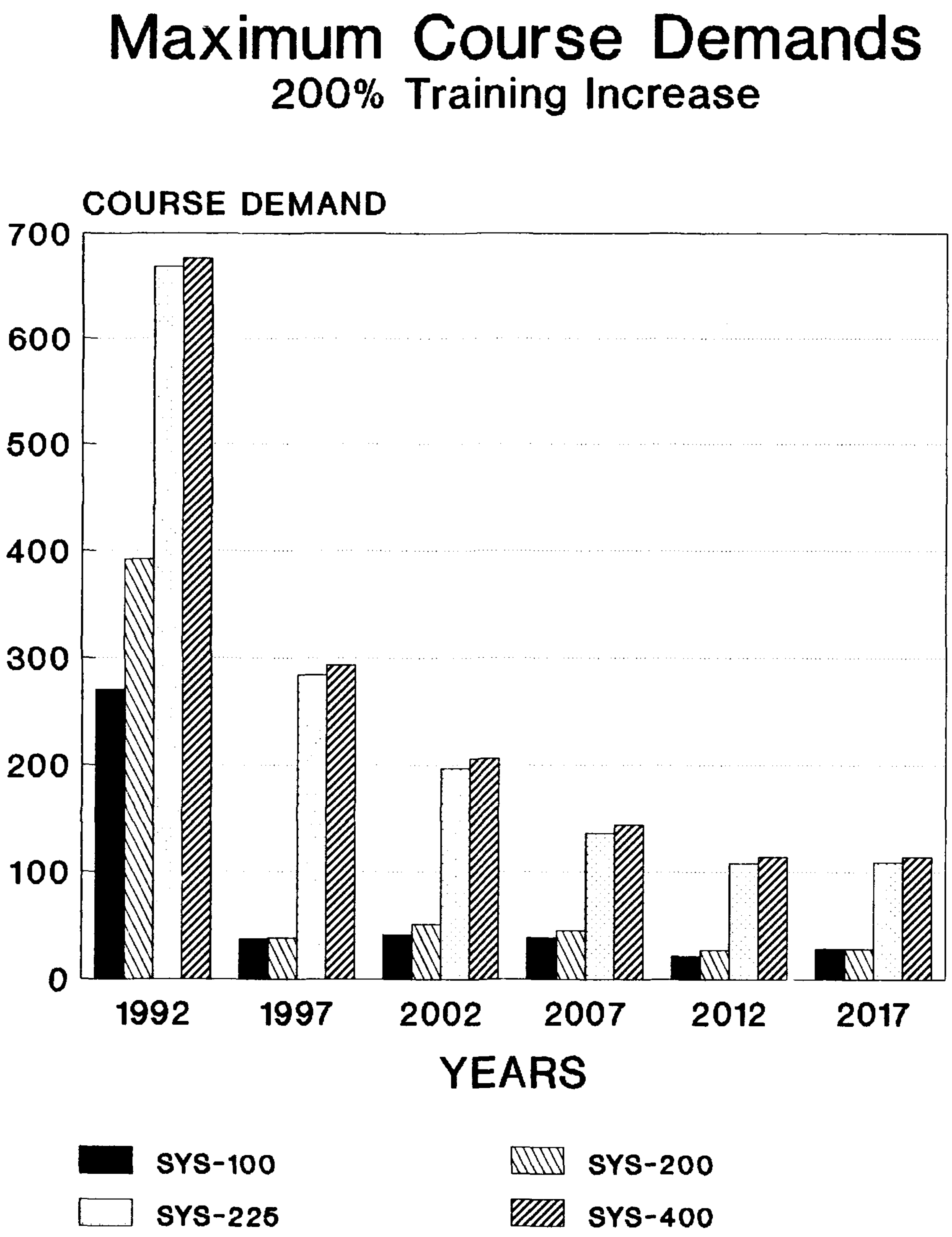
50% Training Increase



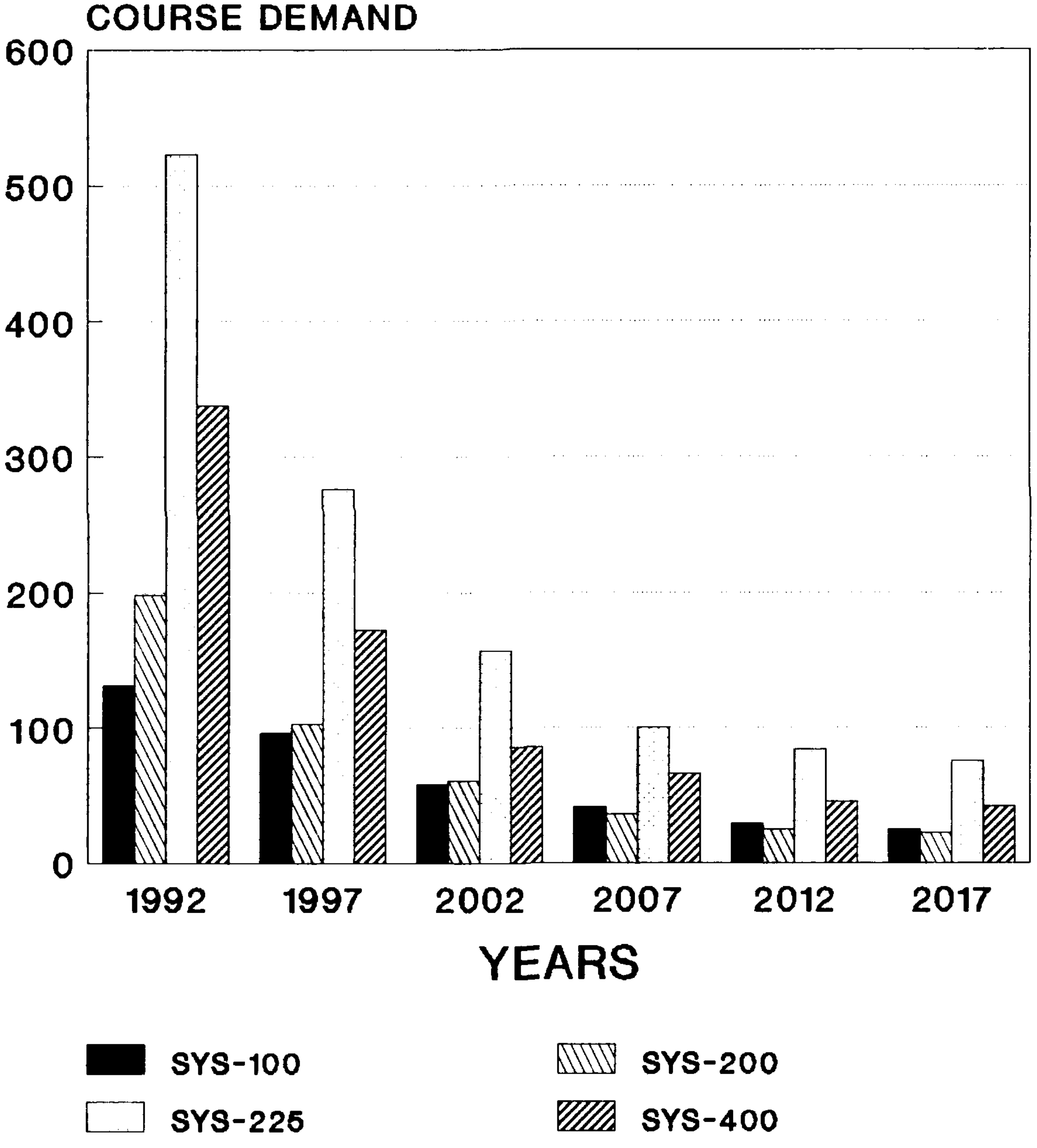
Maximum Course Demands

100% Training Increase

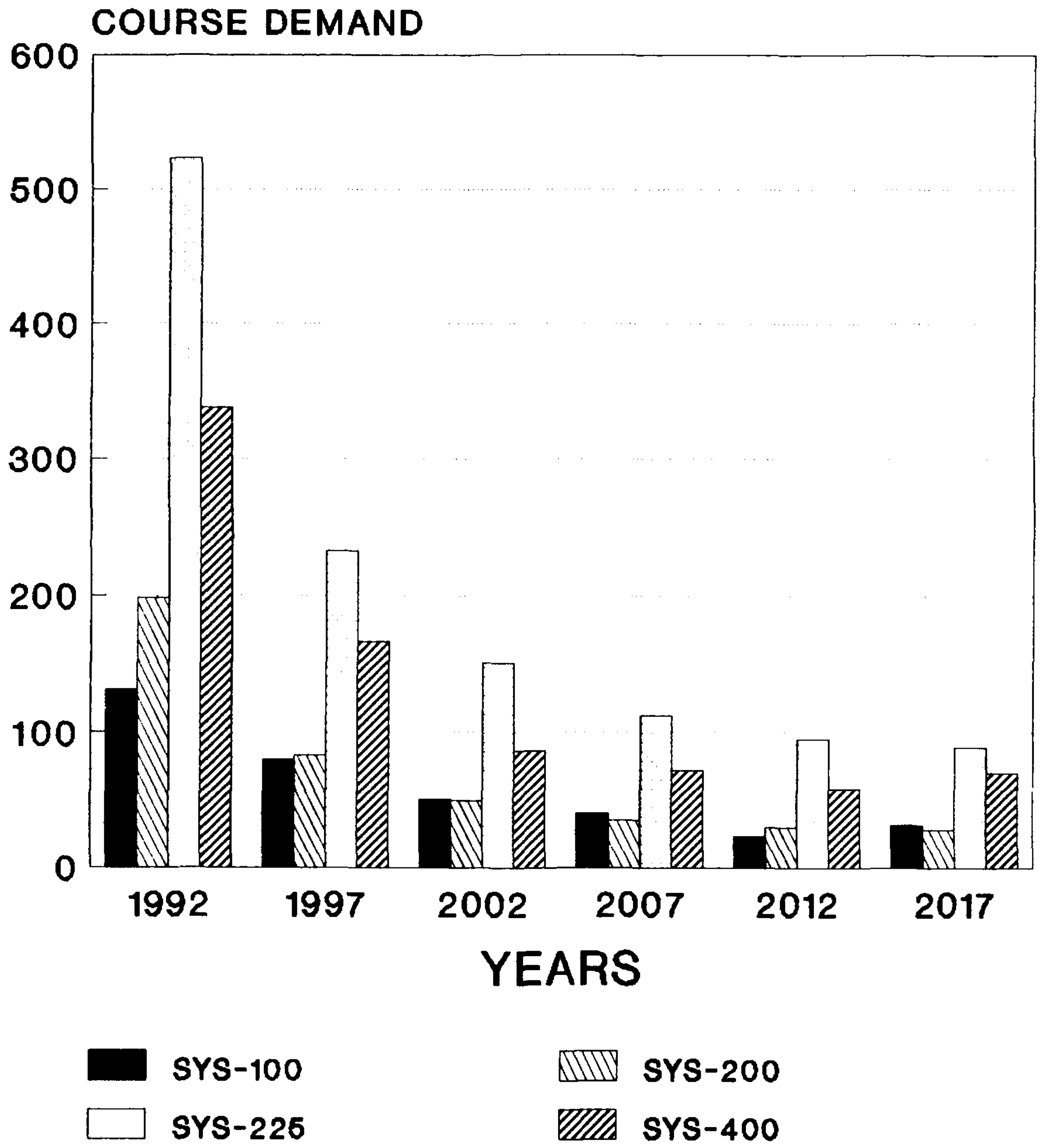



Maximum Course Demands
200% Training Increase
COURSE DEMAND
700
600
500
400
300
200
100
0
1992
1997
2002
2007
2012
2017
YEARS
SYS-100
SYS-200
SYS-225
SYS-400

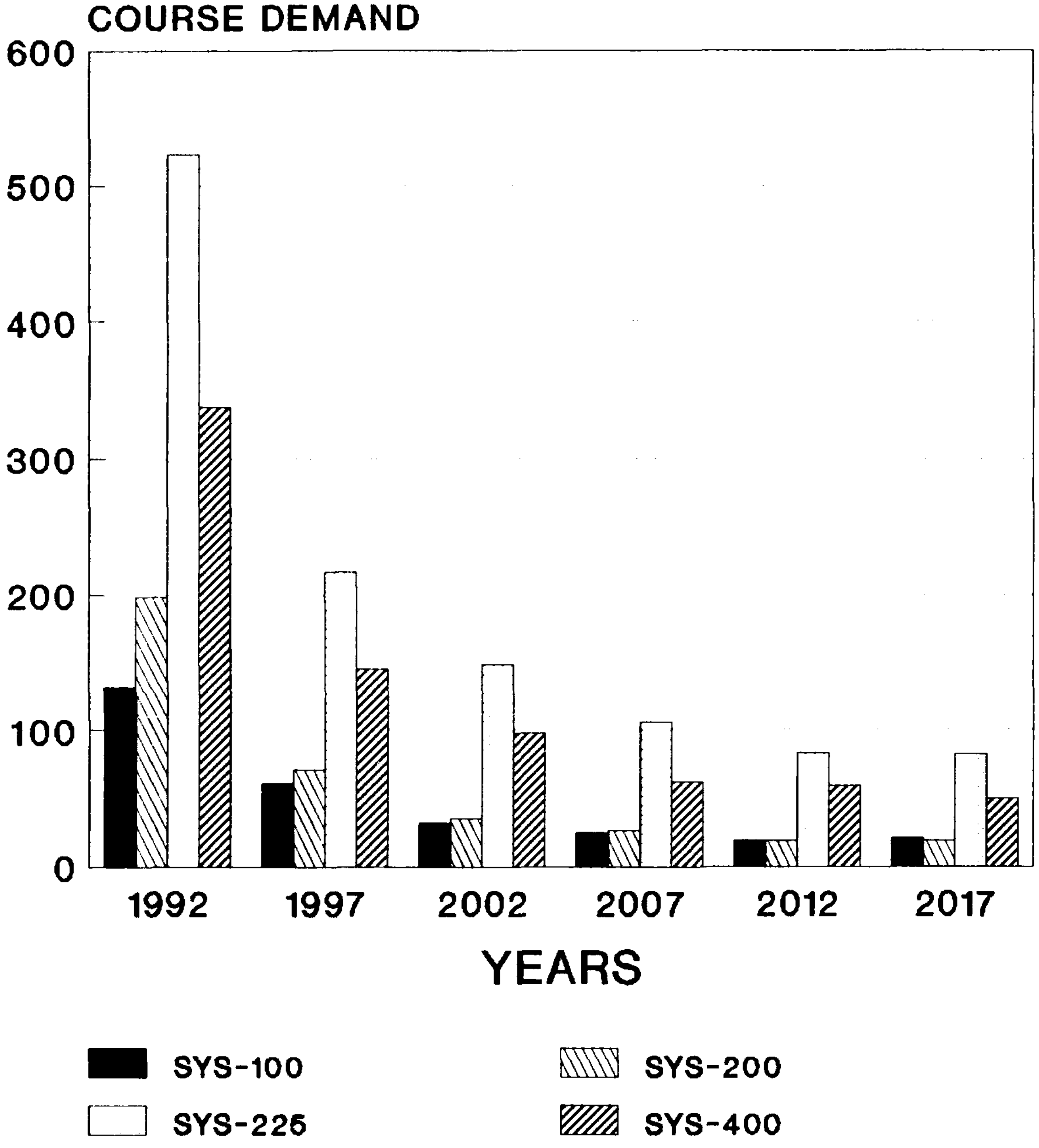
Minimum Course Demands
Based on 1991 Training Capability



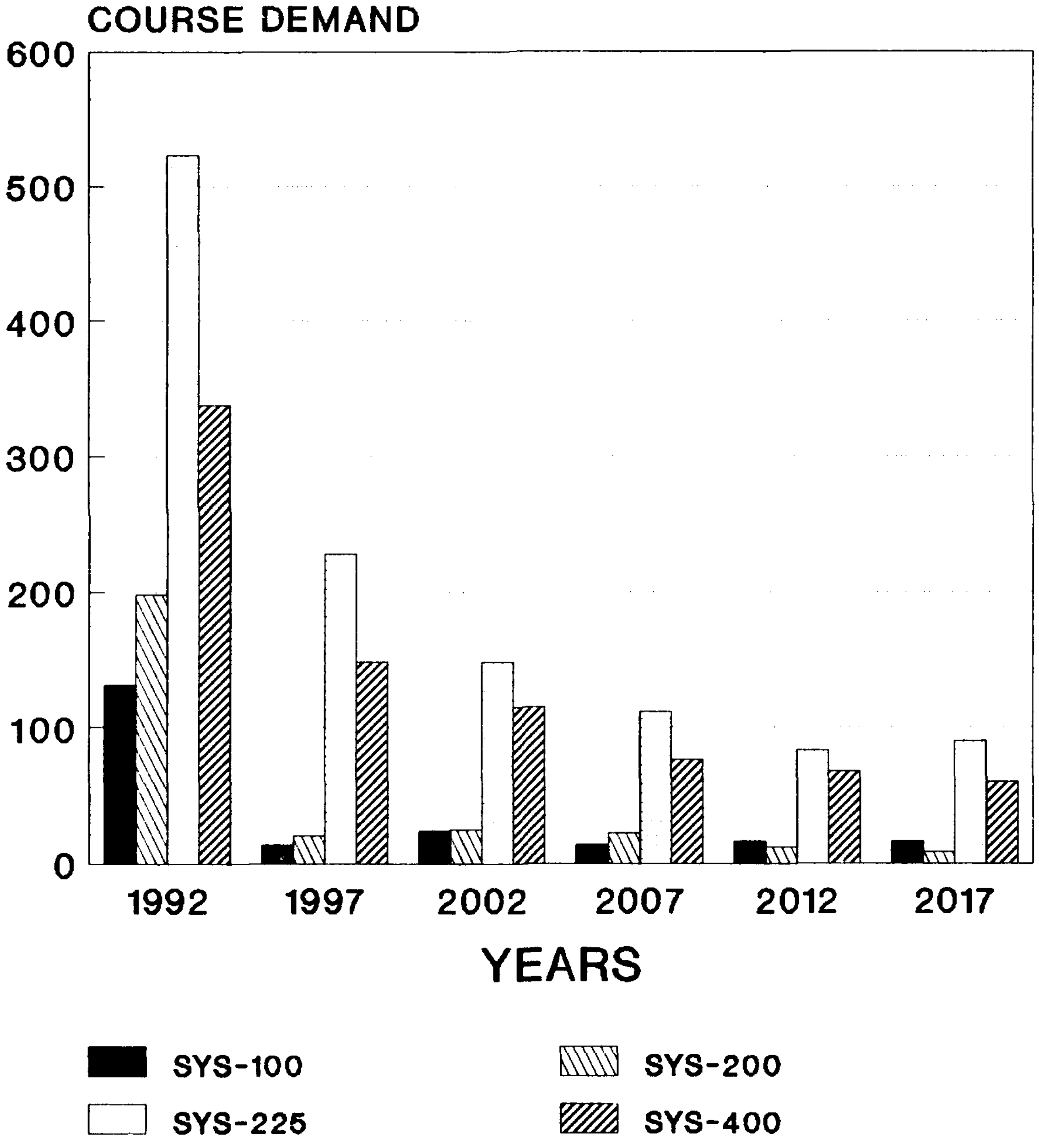
Minimum Course Demands
50% Training Increase



Minimum Course Demands
100% Training Increase



Minimum Course Demands
200% Training Increase



Appendix K

Randomize Sample Selection Routine

```
      EXTERNAL RNUNF
      INTRINSIC REAL, NINT

      INTEGER ROWS, COLS, ROWS1, I

      PARAMETER(ROWS=9116, ROWS1=1000, COLS=40)

      INTEGER POP(ROWS1, COLS), AUX(ROWS,COLS), CHOICE(ROWS)

      OPEN(UNIT=10, FILE='NEW.DAT',STATUS='OLD')
      OPEN(UNIT=12, FILE='POP.DAT',STATUS='NEW')


      DO 10 I = 1,ROWS
          DO 20 J = 1, COLS
             AUX(I,J) = 0
20        CONTINUE
10     CONTINUE


       DO 30 I = 1, ROWS
          READ(10,100,END=41) (AUX(I,J), J=1,40)
30     CONTINUE
41     CLOSE(10)

100   FORMAT(I8, 5(1X,I3), 1X, I8, 9(1X,I3), 24(1X,I4))

      DO 200 J=1,ROWS
           CHOICE(J) = 0
200   CONTINUE

      DO 300 II = 1,ROWS1
14       CONTINUE
         TEMP=RNUNF()
         JJ = NINT(TEMP*ROWS)
          IF ((JJ .LE. ROWS) .AND. (JJ .GT. 0)) THEN
              DO 400 K = 1,ROWS
                 IF (JJ .EQ. CHOICE(K)) GO TO 14
400           CONTINUE

              DO 500 KK = 1,COLS
                 POP(II,KK) = AUX(JJ,KK)
```

```
500               CONTINUE
                  CHOICE(II) = JJ
               ELSE
                  GO TO 14
               END IF
300         CONTINUE

         DO 600 LL = 1, ROWS1
            WRITE(12,100) (POP(LL,N), N=1,COLS)
600      CONTINUE
         END
```

Bibliography

1. Balman, Sid Jr. "Northrop Faces Flurry of Charges, Investigations," *Air Force Times, 50*: 24 (September 1989).

2. Barnes, Todd A. and Leslie L. Gardner. "A Simulation Model For Production Scheduling Of A Facility In A Process Industry," *Proceedings of the 1990 Winter Simulation Conference*: 609-613 (1990).

3. Center For Defense Information. "Waste in Military Procurement: The Prospect for Reform," *THE DEFENSE MONITOR, 15*: entire issue (1986).

4. Cheney, Richard, Secretary of Defense. *"Defense Management Review."* Report to the President. The White House, Washington DC, June 1989.

5. Department of the Air Force. *Acquisition Professional Development*. AFR 800-1. Washington: HQ USAF, 26 December 1990.

6. Department of the Air Force. *Air Force Acquisition System*. AFR 800-1. Washington: HQ USAF, 16 February 1990.

7. Department of the Air Force. *Civilian Career Program Management*. AFR 40-110. Washington: HQ USAF, 3 October 1988.

8. Department of the Air Force. *Occupational Analysis*. AFR 35-2. Washington: HQ USAF, 23 July 1982.

9. Department of the Air Force. *System Acquisition Program Manager Career Program, Vol XVI*. AFR 40-110. Washington: HQ USAF, 24 December 1987.

10. Department of the Air Force. *Training Decision Systems: Development of the Task Characteristics Subsystem*. AFHRL-TR-88-15. Texas: Brooks AFB 1988.

11. Department of Defense. *Reporting Functional and Training Related Data on DOD Military and Civilian Acquisition Personnel*. DOD Instruction 5000.55. Washington: Government Printing Office, 15 November 1991.

12. Department of Defense. *Career Development Program For Acquisition Personnel.* DOD Manual 5000.52. Washington: Government Printing Office, 15 November 1991.

13. Dornheim, Michael A. "Judge Fines Rockwell $5.5 Million For Concealing Navstar Double-Billing," *AVIATION WEEK & SPACE TECHNOLOGY, 130*: 24-25 (March 1989).

14. Hansen, Gen Alfred G. "AFLC Raises the Procurement Standard," *AIR FORCE Magazine, 72*: 102-105 (January 1989).

15. Mann, Paul J. "U.S. Fraud Probe Rekinkles Military Waste Controversy," *AVIATION WEEK & SPACE TECHNOLOGY, 129*: 14-19 (July 1988).

16. Packard, David A. "Micromanagement: The Fundamental Problem With the Acquisition System," *Defense 88, Special Issue on Acquisition Leadership*: 7-13 (1988).

17. Phillips, Don T. "Simulating Markovian Processes: The Pipeline Problem Revisited," *Operations Research Principles and Practices.* John Wiley & Sons, Inc. 1976.

18. Pritsker, Alan B. *Introduction to Simulation and SLAM II.* New York: Halsted Press, 1986.

19. Ruck, Hendrick W. "The Air Force Training Decision System: Modeling Management and Policy Issues For Efficient Training," *Proceeding of the Annual Conference of the Military Testing.* 1-6.

20. U.S. Congress. *Defense Acquisition Workforce Improvement Act.* Public Law No. 510, 101st Congress, 2nd Session. Washington: Government Printing Office, 1989.

21. U.S. Congress. *Defense Authorization Act.* Public Law No. 456, 100th Congress, 2nd Session. Washington: Government Printing Office, 1989.

22. Vaughan, D. S. "Integration/Optimization Subsystem: An Integrated Modeling Approach," *Proceedings of the 30th Annual Conference of the Military Testing Association.* 803-808. Arlington, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Vita

Captain Paul A. Nicholson was born on 4 July 1964 in Rocky Mount, North Carolina. He graduated from Southeast Halifax High School in 1982. After high school, he attend North Carolina Central University where he majored in the fields of Math and Computer Science. While at North Carolina Central University, he cross-enrolled in the Air Force Reserve Officer Training Corps at Duke University. Upon recieving his degree in 1986, he was commissioned into the Air Force and assigned to the 1912 Computer Systems Group at Langley AFB, Virginia where he served as a Test Manager for Joint Tactical Command and Control Systems. Captain Nicholson remained at Langley AFB until his selection to the Air Force Institute of Technology, School of Engineering, Department of Operational Research in 1990.

Captain Nicholson married the former Wendy Ann Nelson of Rome, New York.

Permanent address: Route 1 Box A440
Enfield, North Carolina 27823