AIR COMMAND
AND
STAFF COLLEGE

STUDENT REPORT
STUDENT MIX SOFTWARE SYSTEM (SMSS) REHOST

MAJOR ARTHUR G. DECELLES 88-0700
MAJOR RICHARD M. JENSEN

"insights into tomorrow"

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

MAY 11 1988

88 5 10 282
REPORT NUMBER 88-0700
TITLE STUDENT MIX SOFTWARE SYSTEM (SMSS) REHOST

AUTHOR(S) MAJOR ARTHUR G. DECELLES, USAF
           MAJOR RICHARD M. JENSEN, USAF

FACULTY ADVISOR MAJOR JOHN P. BUCKNER, ACSC/DO

SPONSOR MAJOR JOHN P. BUCKNER, ACSC/DO

Submitted to the faculty in partial fulfillment of requirements for graduation.

AIR COMMAND AND STAFF COLLEGE
AIR UNIVERSITY
MAXWELL AFB, AL 36112-5542
STUDENT MIX SOFTWARE SYSTEM (SMSS) REHOST (U)

Decelles, Arthur G., Major, USAF; Jensen, Richard M., Major, USAF

Student mixing is integral to the operation of the Air Force Air Command and Staff College (ACSC). The concept of mixing and balancing student characteristics and backgrounds within seminars is designed to increase the exposure of each student to different experiences and expertise, and facilitate self-improvement feedback from multiple instructors. The Student Mix Software System (SMSS) is used at the school to maintain student data and, in conjunction with prioritized mixing rules, assign students to seminars. This report documents the rehost and use of SMSS for the Zenith 158 personal computer, and includes the computer code and User Manual for SMSS. The SMSS rehost was designed to take advantage of the data handling and user interface capabilities of the SMART integrated software package, and the programming capabilities of GW-BASIC. The rehosted SMSS dramatically increases balancing capability, and the over-all utility of the system. The rehosted SMSS is so successful that it is already in operation; it was used to generate the third mix student assignments for ACSC class of 1988.
This report documents the rehost and use of the Student Mix Software System (SMSS) for the Zenith 158 personal computer. SMSS is used at the Air Force Air Command and Staff College (ACSC) to maintain characteristic and background data on students and, in conjunction with prioritized mixing rules, assigns students to seminars. The rehosted SMSS dramatically improves balancing capability, and the over-all utility of the system. The rehosted SMSS is so successful that it is already in operation; it was used to generate the third mix student assignments for ACSC class of 1988.

The authors wish to publicly acknowledge the assistance and support of many people who supported the completion of this project. Special thanks and recognition is given to Major John Buckner, USAF, who patiently answered the many questions necessary to finalize the rehost products. We also thank Captain Ronald Ford, USAF, who assisted in the acquisition of the GW-BASIC compiler used in this project.

This project includes over 3,000 lines of computer source code which is listed in the SMSS User Manual (Appendix A) and is maintained on three 5 1/4" computer diskettes (Appendix B of original report only). The computer code for this project may be obtained by contacting the ACSC Mix Master, ACSC/DO, Maxwell AFB, AL 36112-5564. Requestor must supply three MS-DOS formatted, double sided, double density, 5 1/4" diskettes.

This research project is submitted to simplify and optimize a process that is key to the education of all students at ACSC, that of the seminar mix. We sincerely hope that future classes capitalize on the benefits and opportunities that a balanced seminar provides, a balance achieved as a result of the rehosted SMSS.
Major Arthur G. Decelles is from Storrs, Connecticut. He graduated from the University of Connecticut in 1972 with a Bachelor of Arts degree in Mathematics and a minor in Computer Science. In 1983 he completed his Masters degree in Business Administration with an emphasis on computer based systems management from the University of Wyoming. Major Decelles received his Air Force commission in 1975 (Officer Training School) and has held positions at Headquarters Military Airlift Command (MAC); the 90th Strategic Missile Wing, Strategic Air Command; and Headquarters Electronic Systems Division (ESD), Air Force Systems Command. He has four years experience as a computer programmer/analyst satisfying MAC's Command and Control software and Computer Simulation modeling needs. At ESD, he managed the only Air Force program dedicated to reducing the exploding costs associated with the acquisition of Mission Critical Computer Resources. Major Decelles' technical experience includes an extensive knowledge of varied computer hardware systems, and the ability to program in seven different computer languages. He is a 1983 graduate of the Air Force Squadron Officer School. Major Decelles is married to the former Rita Desautels of Willimantic, Connecticut and they have two sons, Timothy and Andrew.

Major Richard M. Jensen is from Brea, California. He attended the United States Military Academy and graduated from California State University, Fullerton with a Bachelor of Arts degree in Mathematics and a Bachelor of Science degree in Computer Science in 1973. Later, he completed a Master of Arts degree in Computer Sciences from the University of Texas at Austin. Major Jensen was commissioned in the United States Air Force in 1975 and has held positions at Headquarters, Air Training Command; Headquarters Air Force; the National Emergency Airborne Command Post (Organization of the Joint Chiefs of Staff); and the Air Force Military Personnel Center. He is a graduate of the Air Force Squadron Officer School and Air Command and Staff College. Major Jensen and his wife, Laurel, have two children, Mark and Amanda.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>iii</td>
</tr>
<tr>
<td>About the Authors</td>
<td>iv</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>vi</td>
</tr>
<tr>
<td><strong>CHAPTER ONE - INTRODUCTION TO STUDENT MIXING</strong></td>
<td></td>
</tr>
<tr>
<td>History of Automated Mixing</td>
<td>1</td>
</tr>
<tr>
<td>Purpose of this Paper</td>
<td>2</td>
</tr>
<tr>
<td><strong>CHAPTER TWO - THE NEED TO REHOST SMSS</strong></td>
<td></td>
</tr>
<tr>
<td>SMSS Defects and Deficiencies</td>
<td>5</td>
</tr>
<tr>
<td>Benefits of Rehosting SMSS</td>
<td>6</td>
</tr>
<tr>
<td><strong>CHAPTER THREE - CONCLUSION</strong></td>
<td></td>
</tr>
<tr>
<td>Third Mix Comparisons</td>
<td>11</td>
</tr>
<tr>
<td>Recommended Changes</td>
<td>14</td>
</tr>
<tr>
<td><strong>BIBLIOGRAPHY</strong></td>
<td>17</td>
</tr>
<tr>
<td><strong>APPENDICES:</strong></td>
<td></td>
</tr>
<tr>
<td>Appendix A - SMSS User's Manual</td>
<td>19</td>
</tr>
<tr>
<td>Appendix B - SMSS Computer Software (Three 5 1/4&quot; Diskettes)</td>
<td>21</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

FIGURES

FIGURE 1 -- Rehosted SMSS Third Mix Results .......................... 12
FIGURE 2 -- Original SMSS Third Mix Results .......................... 13
Chapter One

INTRODUCTION TO STUDENT MIXING

Student mixing is integral to the operation of the Air Force Air Command and Staff College (ACSC) at Maxwell AFB, Alabama. In fact, the mixing of ACSC students is directly linked to the mission of the school. The ACSC mission "... to enhance the professional knowledge, skills, and perspectives of mid-career officers for increased leadership roles in command and staff positions" (3:1) led to the establishment of several goals. Of particular importance to this paper are the school's goals to provide a forum for each student's professional contribution, and an environment for the personal and professional growth of each student. (3:1) The concept of mixing and balancing student characteristics and backgrounds within seminars is designed to achieve these goals by increasing the exposure of each student to different experiences and expertise, and facilitating self-improvement feedback from multiple Faculty Instructors (FIs). (3:3) The student mixing concept is such an integral part of the school's mission that the Director of Operations appoints a mix master for the entire school and each squadron commander assigns an FI as the focal point for student mixing requirements.

In addition to the extensive people resources assigned to the student mixing effort, the school is actually structured to facilitate the mixing concept. The school is currently made up of four squadrons, each consisting of up to eleven seminars of approximately thirteen students each. The school and squadron mixers balance the seminars by evenly assigning officers of like characteristics and backgrounds across seminars. For example, each seminar has at least one International Officer (10) assigned, as well as one Army officer. Other types of officers, like females and minorities, are spread across seminars as evenly as possible. This balancing process occurs three times during the ACSC school year. With the exception of 1Os and S0S faculty, ACSC students must be assigned to new seminars (different FIs) each mix. In addition, the number of seminar classmates with whom a student was previously assigned is minimized. To accommodate for the graduation of all the 1Os at the end of the second mix, one seminar in each squadron is closed down for the third mix.

Assigning approximately 570 ACSC students into 44 seminars may seem like a trivial task, but when you add some thirty different rules and constraints to a process that occurs three times each school year the task quickly becomes complex and time-consuming. The basic idea behind the
mixing rules is to provide the widest range of experience possible to expose each student to new and different ideas, career fields, and service and country backgrounds. It is also important that each seminar contain representative experience and expertise as each major area of the ACSC curriculum is studied. Topics such as tactical, strategic and space operations and acquisition, logistics and budgeting functions are more quickly grasped and understood when first-hand knowledge is available within the seminar. Expertise within each seminar is especially desirable since the students present and lead the majority of the seminar lessons themselves. A well balanced student mix requires the constant and repetitive consideration of numerous factors, a task particularly well suited for a computer.

HISTORY OF AUTOMATED MIXING

Student mixing was first automated as the "Seminar Automated Mixer" (SAM) which ran on the mainframe Honeywell computer at Gunter AFS. The system was extremely slow requiring seven to ten days to receive the output reports. In addition, the SAM system violated most of the mixing rules and required extensive manual manipulation by the school and squadron mixers. (8:--) It took approximately two months prior to the start of the ACSC school year to input and process the student data, and produce just the initial reports. (4:2) Although a breakthrough from the totally manual method of mixing, the SAM system was terribly unresponsive and quite inefficient by today's standards. It was subsequently replaced with the current automated system, the "Student Mix Software System" (SMSS).

SMSS was designed to run on the Zenith 120 computer using the ZBASIC programming language and the CONDOR III Relational Data Base Management System (DBMS), systems which were readily available at ACSC. (4:3) This configuration brought the automated mixing capability directly into the mix master's office, and provided real-time, hands-on access. SMSS incorporated all the mixing rules, heuristics and factors needed by the school at the time, and it sharply reduced the amount of manpower required to produce a final mix from about 200 hours (4:31) to about 40 hours. (5:--) The system was severely constrained, however, by the limited memory capacity of the Z-120 (256K Bytes RAM) and the limited report formatting capability of CONDOR III and ZBASIC. SMSS was also unnecessarily constrained by a three tiered priority scheme for implementing the mixing rules; a constraint which severely handicapped the mix masters ability to properly balance student characteristics and skills automatically. (5:--) In the final analysis, SMSS was a giant step forward because it brought a much needed capability directly into the user's office. By providing initial reports within an hour of initiation SMSS was much more responsive than the old SAM system; so responsive that the mix master could repeatedly refine and rerun the mix several times a day. Unfortunately, as will be seen in the next chapter, this capability would become the most redeeming feature of a system which was plagued with problems.
PURPOSE OF THIS PAPER

The heart and soul of this project is the computer software and User’s Manual attached to this paper (appendix A and B). The software provides the actual data manipulation, mixing, and report producing capabilities, while the User’s Manual provides the ability to easily use and maintain the system. By far the majority of the effort went into the development of these products. The purpose of this paper is to establish the absolute need for these products, and consequently, justify the effort spent.

This chapter introduced the ACSC student mixing concept and provided some background into the development of SMSS. Chapter two assesses the need for rehosting the SMSS system by identifying the many shortcomings of SMSS and exploring the many benefits realized in rehosting the system. Lastly, chapter three provides some conclusions about the SMSS rehost effort, how well the new SMSS operates in comparison to the old system, and what improvements could be made to make it even better.
Chapter Two

THE NEED TO REHOST SMSS

SMSS was a tremendous step forward for ACSC student mixing but it definitely had its problems and limitations. By providing real-time, hands-on access and rapid turnaround, SMSS provided a responsiveness that was previously only dreamed of. But some SMSS functions just didn't work correctly and several others left a lot to be desired. These defects and deficiencies were important ingredients which fed into the decision to rehost SMSS, but the predominant reason for rehosting the SMSS system was its inherent lack of flexibility.

Like the earlier SAM system, SMSS was unable to accommodate even minor changes to the mixing process. Since the development of SMSS, two student characteristics ceased to be factors in the mixing process: that of "Communication Skills" and "Air Research Institute" students. (5:--) It wasn't difficult for SMSS to ignore these traits, but it was definitely a problem to incorporate new mixing rules into SMSS. The student characteristic of "Space Operations Experience" became a major mixing factor in 1987. The only way the mix master could make SMSS handle this new mixing requirement was to use the existing "Communication Skills" field to identify students with "Space Operations Experience." (5:--) This was a brute force method to make SMSS handle the new requirement, but the SMSS reports still reflected the "Communications" label. However, SMSS could not handle any other additional characteristics, even the brute force way. The ACSC mix master needed the capability to handle additional student mixing characteristics, especially considering the overwhelming emphasis being placed on "Joint Service Experience" in upcoming ACSC classes. This is a capability that SMSS currently lacks.

Another major change since the development of SMSS was an ACSC decision to standardize the school's computer hardware and software. In late 1986 the school made an investment of about $225,000.00 in purchasing 65 Zenith 158 (IBM-compatible) personal computers, and acquiring the SMART integrated software package for each machine. (6:--) The Z-158 has more than twice the internal memory (640K Bytes RAM) of the Z-120 and, consequently, isn't as constrained in the number and type of computer application programs that can be used. In addition, the SMART software package provided a capability which integrated four vital computer applications within one system: wordprocessing, data base management, telecommunications, and spreadsheet. In essence, the school decided to standardize, for the foreseeable future, on the capabilities and user
interfaces inherent in these systems. To maximize the return on the school's investment in standard computer resources, the ACSC Director of Operations set an objective to get "... the faculty proficient and using the SMART software system." (2:1) However, as the ACSC mix master soon discovered, SMSS would not run on the Z-158s, and it was not compatible with the SMART data base manager. (5:--) This was an SMSS limitation that needed to be rectified.

SMSS clearly lacked the flexibility necessary to accomodate even the most minor mixing changes. This chapter will further assess the need to rehost SMSS by first identifying SMSS defects and deficiencies, and then by exploring the many benefits to be gained in rehosting the SMSS system.

SMSS DEFECTS AND DEFICIENCIES

SMSS has been used since 1986 to maintain student data, perform the student mixing, and generate the required reports. During that time, several difficulties were encountered in using SMSS. (5:--) The ACSC mix master was repeatedly interviewed to identify the problems and limitations of the current SMSS system, and to determine the existing student mixing requirements and practices. This information was further supplemented by reviewing related Air University and ACSC regulations dealing with seminar organization, rank, and procedures, and by inspecting the actual SMSS computer code and documentation. The result was the identification of several SMSS defects and deficiencies. A defect refers to a capability that doesn't work correctly, while a deficiency refers to a capability that is difficult to use or incorporate. Both defects and deficiencies have hidden costs associated with them.

The first defect of SMSS is that the mixing module incorrectly assigns Seminar Leaders (SLs) and Assistant Seminar Leaders (ASLs). The school has a very strict procedure for assigning SLs and ASLs, and the assignment rules change depending on the mix being processed. (1:1) The SMSS mixing module assigned students that were either not eligible for the position, or didn't assign students who were. The result was several hours of manual checking by the mix master to insure the SLs and ASLs were correctly assigned. (5:--) This was such a cumbersome task that, eventually, the mix master started pre-assigning students to seminars and to SL/ASL positions manually within the SMSS data base module. This one SMSS defect cost several hours of mix master time each mix, and considerably reduced the randomness and flexibility of the mixing process.

Another defect of the SMSS mixing module is the inaccurate statistical report summaries that it produced. (5:--) This defect had an indirect, yet more significant, cost associated with it. The mix master could get the correct mix statistics by generating SMSS data base reports using CONDOR III capabilities. This requires minimal effort by the mix master, but, the balancing capabilities of the SMSS mixing module strongly depends upon the student statistics in maintains. (4:58-97) Consequently, the
inaccurate statistics gave the mixing module a false indication of seminar balance, and directly affected student assignments. The result is the inability of SMSS to properly balance seminars, the prime SMSS function. Ineffective seminar balancing cost several hours of manual changes, each by the school and squadron mixers to achieve an acceptable balance, a job that SMSS was created to do and which it should have done right by itself.

The last SMSS defect is that the mixing module periodically caused the computer to "hang-up". (8:--) This defect is an irritant to the mix master because the only way to get out of this condition is to re-start the computer and re-run the mix. (5:--) Fortunately, this condition occurs infrequently and cost very little in time and effort, however, it demonstrates the immaturity of the SMSS system. The SMSS mixing module contains recursive computer code (code that calls itself) and unstructured computer code. (4:58-97) Such programming practices lead to conditions that cause computers to "hang-up" and make it nearly impossible to maintain the system. All three of these SMSS defects need fixing, but the mix master could often tolerate them more than he could the other SMSS deficiencies.

The most significant deficiency of SMSS is the inadequate control it provides to the mix master over mixing rule priorities. SMSS was developed using a three tiered priority scheme which enabled the mix master to designate each mixing rule as being either "Always Important", "Sometimes Important", or "Not Important". (4:21) With the requirement to mix students on eighteen different characteristics, this priority scheme didn't provide the capability needed to define the relative importance of each rule during a given mix. As a result, the mix master had to adjust and re-run the mix several times to even come close to the desired results. (5:--) This deficiency caused several hours of mix master time in re-running the mix, and in manual adjustments to seminar assignments afterwards. The developers of SMSS recognized the balancing inefficiencies of the system and suggested an additional computer program be written to adjust the seminar assignments made by the mixing module. (4:32) It is absolutely ridiculous to write a computer program to do the job that the original computer program is suppose to do. The three tiered priority scheme incorporated within SMSS unnecessarily constrained the mix master, it caused the SMSS mixing module to work harder and longer than was really required, and in the end it caused poor mix balances. The seminar imbalances produced by SMSS are the direct result of an inadequate mixing rule priority scheme.

Another deficiency of SMSS is its poor report generation capability. As noted by the original developers, SMSS "... does not automatically produce the final report in the correct format." (4:32) This deficiency is attributable to the limitations of the CONDOR III DBMS which provides all the information necessary for the Student Statistical Report, but which cannot handle the required report format. SMSS also lacks the capability to provide real-time, ad hoc reports. (5:--) The ACSC mix
master is constantly called upon to provide special purpose, one-time reports, especially between mixes. Although this is a major requirement of the mix master, SMSS is deficient in the support it provides in this area.

Three of SMSS's capabilities don't work correctly and two others are deficient in providing all the capability the mix master's needs. These defects and deficiencies, along with the general inability of SMSS to handle changing conditions, led to the decision to rehost SMSS.

**BENEFITS OF REHOSTING SMSS**

Solving all the problems of the original SMSS system required much more than quick patches, it involved a complete rehosting and redesign of the system. SMSS needed the ability to run on the ACSC standard Z-158 computer using the SMART data handling and user interface capabilities. The SMSS mixing module needed to be rewritten to correct the current defects and to "optimize" the mix balancing capability. Last but not least, a comprehensive User's Manual needed to be developed. This was the scope of the SMSS rehost effort, an effort which would provide the mix master with enormously increased capabilities and the ability to produce the mixing products in a fraction of the time. The remainder of this section explores the benefits possible in four areas of SMSS: defects, deficiencies, flexibility, and maintainability.

Just fixing the three SMSS defects identified earlier would be strong justification for the rehost effort. Correcting the Seminar Leader and Assistant Seminar Leader assignment defect would save the time required to manually recheck or pre-assign students each mix. Likewise, fixing the SMSS mixing module statistics defect would save time by reducing the number of manual reassignments necessary to achieve an acceptable mix balance. Correcting the SMSS computer "hang-up" defect would save the time needed to re-run the mix and would help restore user confidence in the SMSS system. Fixing the three SMSS defects would save an estimated 20 hours each mix in the time and effort required by the school and squadron mixers to produce the necessary mixing products.

Correcting the SMSS deficiencies would not only save user time, but would provide capabilities comparable to the "responsiveness" provided by the original SMSS system. Replacing the three tiered priority scheme of SMSS with a ten level priority scheme would provide the mix master with a control over student mixing never thought possible. It would save the time required to re-run the mix over and over, and would "optimize" the student mixes. Providing the capability to generate correctly formatted reports directly with the SMSS data base module would save the time required for the mix master to reformat reports with a word-processor and would support the mix master in the real-time generation of adhoc reports. Fixing the SMSS deficiencies would provide the mix master with tremendous new capabilities; the capability to "optimize" the mixing process, and the
capability to produce reports as needed. These new SMSS capabilities alone would justify the SMSS rehost effort, but even greater benefits are possible.

Correcting the SMSS defects and deficiencies were major determinants in rehosting the SMSS system, but the predominant reason for rehosting SMSS to the Z-158 was to make the SMSS system more flexible, usable, and maintainable. The original SMSS is unable to handle additional student characteristics and mixing rules. One of the benefits of the rehost would be the addition of five unlabeled mixing identification fields which are ready for immediate use. SMSS would also be designed to allow even more fields to be added. Further flexibility would be attained by rehosting SMSS to IBM-compatible computer hardware (Z-158), and by providing three different user definable SMSS configurations: dual floppy disks, hard disk and floppy disk, and RAM disk with either hard or floppy disk. These features would increase the flexibility and usability of SMSS by expanding the number of computers SMSS could potentially run on.

Rehosting the SMSS to the Z-158 under the SMART integrated software package would make SMSS more usable and maintainable. The SMART database manager would provide SMSS with a standard ACSC user interface capability, and would provide the SMSS user with an automated "help" feature for each SMSS menu option. The SMSS "help" feature would inform the user of each menu option available and would provide textual rational for choosing an option. A critical product for the usability of the SMSS system is adequate user documentation. The SMSS rehost effort would produce a comprehensive User's Manual (Appendix A) which would thoroughly explain the design of the rehosted SMSS, and would guide the user through each step of the SMSS mixing process. The User's Manual would also be invaluable for modifying the SMSS computer code and for maintaining the system in the future. In addition, all SMSS computer code would be structured and documented to further enhance the maintainability of the system.

The rehost of SMSS corrected defective and deficient capabilities, and provided powerful, new capabilities for using and maintaining the system. The next chapter will compare the performance of the rehosted SMSS to the original system, and suggest improvements which could make it even better.
Chapter Three

CONCLUSION

Although the SMSS rehost effort required a complete redesign of the system and rewrite of the computer code, the original SMSS name was not changed. SMSS is an automated mixing concept, a concept which really didn't change with the rehost. The rehost fixed SMSS defects and deficiencies and provided additional capabilities, but the mixing concept stayed the same. In addition, the original SMSS was developed as a prototype system (4:3), a prototype which never reached full maturity. Consequently, the original name was retained.

The success of the SMSS rehost effort can be judged in its performance against the performance of the original system. The rehosted SMSS was used to generate the third mix student assignments for ACSC class 1988. The original SMSS was also run for purposes of comparison. The rest of this chapter compares the performance of the two systems and recommends improvements which can make the rehosted SMSS even better.

THIRD MIX COMPARISONS

Figures 1 and 2 show comparative results from the rehosted SMSS and the original SMSS for the third mix of ACSC class 1988. The results for only the 3821st Student Squadron are shown, but are representative of the entire school. The figures show the number of students SMSS assigned to each seminar which possess the characteristic listed on the left. Those student characteristics which are not completely balanced are underlined for clarity purposes. Prior to running this mix, the mix master assigned the highest rule priority of "9" to the following characteristics: Army; Reserves, ANG, Civilians, USMC, or Navy; Minorities; Females; and Medical, Legal, or Chaplains. The lowest rule priority of "0" represents characteristics which the mix master is not concerned with balancing. Priority "0" was assigned to the following characteristics: Line AD AF; No Degree; HQ Experience; Academy Graduate; Captains; and Bomber or Missile Experience. Comparable priorities were assigned by the mix master using the "Always", "Sometimes", and "Not Important" scheme of the original SMSS system. Figures 1 and 2 reflect the actual student assignments of the two SMSS systems, and do not contain any manual adjustments.

In every instance, the rehosted SMSS outperformed the original SMSS in balancing the third mix student characteristics. The only non-zero
<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Line AD Air Force</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2. No advanced Degree</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3. HQ (/)=Majcom) Experience</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4. Any Pilot</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5. Any Navigator</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6. Anyhow Unaccompanied</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7. USAFA (academy) grad</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8. Army</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9. Res,ANG,Civ,USMC,or Navy</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10. Black or other</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11. Female</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12. Current Captains</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13. Medical,legal,chaplain</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14. Bomber or missile exp</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15. Fighter pilot or WSO</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>16. PPBS experience</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>17. Aco/Log experience</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>18. Space experience</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19. Strat Ops</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**FIGURE 1.** Rehosted SMSS Third Mix Results.
### MAJOR STATISTICS - FOR SQUADRON 1

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMM SKILLS</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PSBS SKILLS</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>TAC OPS SKILL</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>STRAT OPS SKILL</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>ACO/LOG SKILL</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>PILOT</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>NAVIGATOR</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SINGLE/UNAC</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>USAFA GRADS</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>ARMY</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RES/NG/USN/USMC</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MINORITIES</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>FEMALES</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RANK - CAPT</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BA/BS/BS/A/BA/BS/A/BS/A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>SR ORG EXP</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>ARI/DES</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NO MASTER ED</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 2. Original SMSS Third Mix Results.**
priority characteristics that the rehosted SMSS failed to perfectly balance were "Unaccompanied" and "Pilots" which were assigned the lowest two non-zero priorities for this mix. Even the "Unaccompanied" category was only off from a perfect balance by two students, and the "Pilot" category was off by one student. On the other hand, the original SMSS failed to balance six characteristics of concern, including one with the highest priority (Medical, Legal, Chaplain/88XX, 89XX, 9XX). The other unbalanced characteristics of interest are: Strat Ops; Acq/Log Exp; Pilots; Navigators; Space Exp; and Unaccompanied. In addition, the original SMSS was often off in its balance by a factor of four. Clearly, the rehosted SMSS outperformed the original system. In fact, it actually "optimized" the third mix.

In addition to optimizing the third mix, the rehosted SMSS proved many of its other capabilities. The portability of the system was demonstrated by running the system on the Z-120 computer with a Gemini Board. This hardware configuration allows the Z-120 to emulate an IBM-compatible computer. The rehosted SMSS ran at about half the speed as on the Z-158, but otherwise operated without difficulty. The user-friendly, menu-driven interface of the rehosted SMSS was demonstrated by the ability of a novice user, the 3624 STUS mixer, to operate the system with minimal assistance. The system's flexibility was demonstrated by successfully using an unlabeled characteristic field during the mixing process. The student information transferred from the original SMSS data base for "Strategic Operations Experience" was flawed. The flexibility of the additional characteristic fields enabled the mix master to quickly recover from this condition without having to manually edit each student record. Last, but not least, the new mixing module correctly assigned all the Seminar Leaders and Assistant Seminar Leaders. The rehosted SMSS clearly met all the performance objectives and exceeded the expectations of all concerned.

RECOMMENDED CHANGES

One limitation of the rehosted SMSS system became apparent during processing of the third mix. The SMSS mixing module is designed to assign the SLs and ASLs before making any other assignments, just as the original SMSS did. This gives the assignment of SLs/ASLs an unintended, artificial priority above all other mixing rules. (7--) It also severely limits the school and squadron mixers' ability to manually reassign students assigned to these positions, and requires that the student dates of rank be rechecked if any manual reassignments are made. If the assignment of SLs and ASLs were moved from the SMSS mixing module to the SMSS data base module, all students could be manually reassigned by the mix master without regard to the final determination of the SLs and ASLs. Besides providing additional capability, this change would provide even more flexibility and randomness to the mixing process.

The rehosted SMSS has dramatically reduced the workload of the school and squadron mixers. It provides much better products, with greater
flexibility, in about the same amount of time as the original system. But, regardless how good the rehosted SMSS, no automated mixing system will ever totally remove all manual adjustments. There are always person unique mixing factors, such as personality, which cannot be automated, but which will always play an important role. The rehosted SMSS "optimizes" the mixing process and, as a tool, provides a tremendous capability to the mix master; a capability which is highly flexible, useable, and maintainable, and which will continue to satisfy the school's future student mixing needs.
BIBLIOGRAPHY

A. REFERENCES CITED

Official Documents


Unpublished Materials


4. Ritchhart, Kenneth M., Major USAF and Simmons, Robert L., Major USAF. ACSC Project 86-2120, The Student Mix Software System (SMSS). Research Study prepared at the Air Command and Staff College, Air University, Maxwell AFB, AL. April 1986.

Other Sources


7. Hansen, Steven L., Major, USAF. 3824 STUS Mixer, Air Command and Staff College, Maxwell AFB, AL. Multiple Interviews: Feb 88.

8. Romer, Irving, F., Major, USAF. Former ACSC Mix Master, Air Command and Staff College, Maxwell AFB, AL. Interview: Jan 88.

B. RELATED SOURCES

Official Documents

Other Sources

10. Innovative Software’s SMARTWARE. The SMART Data Base Manager. PO Box 15998, Lenexa, KS. 66215-9990, 1986.


Appendix A

STUDENT MIX SOFTWARE SYSTEM (SMSS)

USER’S MANUAL
AIR COMMAND AND STAFF COLLEGE

STUDENT REPORT

STUDENT MIX SOFTWARE SYSTEM

USER'S MANUAL

"insights into tomorrow"
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Illustrations</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td><strong>CHAPTER ONE -- SMSS OVERVIEW.</strong></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>1.1</strong></td>
<td>What is SMSS?</td>
<td>1</td>
</tr>
<tr>
<td><strong>1.2</strong></td>
<td>SMSS Structure</td>
<td>1</td>
</tr>
<tr>
<td><strong>1.3</strong></td>
<td>Using This Manual</td>
<td>2</td>
</tr>
<tr>
<td><strong>CHAPTER TWO -- INSTALLING AND STARTING SMSS</strong></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td><strong>2.1</strong></td>
<td>The SMSS Release Disks</td>
<td>3</td>
</tr>
<tr>
<td><strong>2.2</strong></td>
<td>Runtime Disk Configurations</td>
<td>3</td>
</tr>
<tr>
<td><strong>2.3</strong></td>
<td>Setup Steps</td>
<td>4</td>
</tr>
<tr>
<td><strong>2.4</strong></td>
<td>Running SMSS</td>
<td>6</td>
</tr>
<tr>
<td><strong>2.4.1</strong></td>
<td>Disk Option Screen</td>
<td>6</td>
</tr>
<tr>
<td><strong>2.4.2</strong></td>
<td>SMSS Main Menu</td>
<td>7</td>
</tr>
<tr>
<td><strong>2.4.3</strong></td>
<td>Menu Selection Rules</td>
<td>7</td>
</tr>
<tr>
<td><strong>2.4.4</strong></td>
<td>Help Screens</td>
<td>8</td>
</tr>
<tr>
<td><strong>CHAPTER THREE -- FILE MAINTENANCE.</strong></td>
<td></td>
<td>11</td>
</tr>
<tr>
<td><strong>3.1</strong></td>
<td>Build Student Records From Another Disk</td>
<td>11</td>
</tr>
<tr>
<td><strong>3.2</strong></td>
<td>Add Student Records Manually</td>
<td>13</td>
</tr>
<tr>
<td><strong>3.3</strong></td>
<td>Edit Existing Student Record</td>
<td>14</td>
</tr>
<tr>
<td><strong>3.4</strong></td>
<td>Delete Student Record</td>
<td>15</td>
</tr>
<tr>
<td><strong>3.5</strong></td>
<td>Designate Student Numbers</td>
<td>15</td>
</tr>
<tr>
<td><strong>3.6</strong></td>
<td>Edit Environment Record</td>
<td>15</td>
</tr>
<tr>
<td><strong>3.7</strong></td>
<td>Delete International Officers</td>
<td>20</td>
</tr>
<tr>
<td><strong>3.8</strong></td>
<td>Post Mixer Tentative Assignments</td>
<td>20</td>
</tr>
<tr>
<td><strong>CHAPTER FOUR -- PERFORM MIXING.</strong></td>
<td></td>
<td>23</td>
</tr>
<tr>
<td><strong>4.1</strong></td>
<td>The SMART/Mixer Interface</td>
<td>23</td>
</tr>
<tr>
<td><strong>4.2</strong></td>
<td>SMSS Mixing Menu</td>
<td>24</td>
</tr>
<tr>
<td><strong>4.3</strong></td>
<td>Mixing Sequence of Operations</td>
<td>24</td>
</tr>
<tr>
<td><strong>CHAPTER FIVE -- PRODUCE REPORTS.</strong></td>
<td></td>
<td>27</td>
</tr>
<tr>
<td><strong>5.1</strong></td>
<td>Report Selection Menus</td>
<td>27</td>
</tr>
<tr>
<td><strong>5.2</strong></td>
<td>School Mixing Summary</td>
<td>29</td>
</tr>
<tr>
<td><strong>5.3</strong></td>
<td>Squadron Mixing Summary</td>
<td>29</td>
</tr>
<tr>
<td><strong>5.4</strong></td>
<td>Seminar Mixing Report</td>
<td>29</td>
</tr>
<tr>
<td><strong>5.5</strong></td>
<td>Complete Mixing Report</td>
<td>31</td>
</tr>
<tr>
<td><strong>5.6</strong></td>
<td>Alpha Rosters</td>
<td>31</td>
</tr>
<tr>
<td><strong>CHAPTER SIX -- MIXMASTER'S CHECKLIST.</strong></td>
<td></td>
<td>35</td>
</tr>
<tr>
<td><strong>APPENDICES:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Appendix One -- Student File</td>
<td></td>
<td>39</td>
</tr>
</tbody>
</table>
Appendix Two -- SMART Project Files - Program Listings...... 43
Appendix Three -- SMSS Mixer Program - Program Listing...... 77
Appendix Four -- SMSS.BAT - Program Listing....................101
# LIST OF ILLUSTRATIONS

## FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>SMART Configure Screen</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Disk Option Screen Display</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>SMSS Main Menu</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Example Help Display</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>File Maintenance Menu</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Build Student File Menu</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Student Record Data Entry Form</td>
<td>13</td>
</tr>
<tr>
<td>3.4</td>
<td>Record Selection Menu</td>
<td>14</td>
</tr>
<tr>
<td>3.5</td>
<td>Environment Record Page 1</td>
<td>17</td>
</tr>
<tr>
<td>3.6</td>
<td>Environment Record Page 2</td>
<td>18</td>
</tr>
<tr>
<td>3.7</td>
<td>Environment Record Page 3</td>
<td>19</td>
</tr>
<tr>
<td>3.8</td>
<td>Environment Record Page 4</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Mixing Option Menu</td>
<td>23</td>
</tr>
<tr>
<td>5.1</td>
<td>Report Mix Selection Menu</td>
<td>27</td>
</tr>
<tr>
<td>5.2</td>
<td>Reports Selection Menu</td>
<td>28</td>
</tr>
<tr>
<td>5.3</td>
<td>Report Output Selection Menu</td>
<td>29</td>
</tr>
<tr>
<td>5.4</td>
<td>School Mixing Summary</td>
<td>30</td>
</tr>
<tr>
<td>5.5</td>
<td>Squadron Mixing Summary</td>
<td>31</td>
</tr>
<tr>
<td>5.6</td>
<td>Seminar Mixing Report</td>
<td>32</td>
</tr>
<tr>
<td>5.7</td>
<td>Student Alpha Roster</td>
<td>33</td>
</tr>
</tbody>
</table>
Chapter One

SMSS Overview

1.1 What is SMSS?

The Student Mix Software System (SMSS) is a set of computer programs used to distribute (or mix) Air Command and Staff College (ACSC) students into ten to thirteen-person seminars to which they will be assigned for parts of the school year. The school philosophy is to distribute students evenly among seminars based on characteristics and skills of each student so that each seminar has a uniformly homogeneous "mix" of students. Students are generally reassigned three times each academic year. The following features of SMSS aid the ACSC staff in assigning students:

- Student data base.
- Menu driven program that "walks" the user through the mixing process.
- Predefined reports

SMSS is designed to run on the Zenith 158 (Z158) computer system, which is commonly used at ACSC. Additionally, it will run on any IBM-compatible personal computer with a configuration similar to the ACSC standard Z158.

1.2 SMSS Structure.

There are two main software elements to SMSS: (1) the user interface and (2) the mixing program. The user interface runs under the SMART integrated software package which is a commercial software system owned by ACSC. SMART contains a word processor, spreadsheet, data base manager, and a telecommunications package. SMSS uses the data manager to maintain its student data base, detailed in Appendix 1. Also, there is programming language within SMART, called the SMART Project Processing feature. The SMSS user interface is written in this language and runs as a SMART project file. Appendix 2 is a source listing of all the SMSS project files. See the SMART user's manuals located in the ACSC Computer Room (room 236) for more information on the SMART data manager and project processor. However, because SMSS guides the user through a menu driven process, it should not be necessary for him/her to be an expert on the SMART data manager.

SMART does not provide the detailed programming capability needed to do student mixing, so the SMSS mixing program is written in the
GW-BASIC programming language. A complete source listing of the mixer is at appendix 3. Again, a knowledge of BASIC is not needed to run SMSS.

1.3 Using This Manual.

This user's manual is organized in the same sequence as the steps involved in the student mixing process. Chapter 2 describes how to load SMSS, how to configure it for your computer, and how to get it started. Next, Chapter 3 shows you how to build and maintain the SMSS data base. Chapter 4 deals with student mixing, and Chapter 5 discusses the reports that show the results of mixing. Finally, Chapter 6 lists the sequence of steps that the ACSC "mixmaster" (the staff officer responsible for student mixing) should go through, using SMSS. Therefore, the first-time user should start at the beginning and march through sequentially as the manual will describe the mixing process as well as how to use SMSS in its support.

There are several examples throughout the manual that show sequences of commands to type into the computer. Actual keyboard inputs will be shown boldfaced and underlined.
Chapter Two

Installing and Starting SMSS

This chapter gives step by step instructions for installing SMSS on your computer and shows how to get the system started. If SMSS is already installed, skip to paragraph 2.4 for running instructions.

2.1 The SMSS Release Disks.

SMSS comes to you on three, 5 1/4 inch floppy diskettes. Disk 1 (SMSSPROG) contains the SMSS source and object programs, and all the auxiliary files needed to run and maintain SMSS. Because the whole system will not fit on one diskette, the large student database is on disk 2 (SMSSSTUS). Together, disk 1 and disk 2 contain all of SMSS. A third disk (SMSSRUN) is similar to disk 1 but contains only the object, or run-time versions of the mixer and user interface programs. Disk 3 is useful if space is limited and you don't want to load up the program source code or if you want to release the system to someone else and don't wish to make the source code available. Make a backup copy of all three diskettes using the MS-DOS, "DISKCOPY" command.

2.2 Runtime Disk Configurations.

Some actions you will take to prepare your computer for SMSS depend on how you plan to run the system. SMSS can be run from either of three disk configurations using (1) floppy disk, (2) hard disk, or (3) a combination of floppy disk and virtual (in-memory) disk, sometimes called ramdisk. Each configuration has certain benefits and constraints in terms of speed and transportability as follows:

FLOPPY DISK ONLY -- If you don't have room on your hard disk to permanently install SMSS, or you want to be able to transport the system between computers, this is the way to go. You must have a computer with two floppy disk drives -- drive A will be used for the SMSSPROG or SMSSRUN disk and drive B will be used for the SMSSSTUS disk. This is the slowest-running option.

HARD DISK -- If you've got a hard disk with at least 750k bytes available, this is the fastest and preferable way to run SMSS. Transportation between computers can be tedious.
FLOPPY AND VIRTUAL DISK -- Virtual disk (VDISK) is an MS-DOS feature that allows you to set aside an area of memory that is treated as if it were a disk device. In this option, SMSS programs run from floppy drive A, but the student data base is copied into VDISK at the beginning of your session and copied back to floppy disk at the end. This method is a compromise between the other two options both in terms of speed and transportability.

If you’re not constrained to a particular mode of operation, or haven’t decided how you will configure SMSS, go ahead with all the following setup steps to allow all modes of operation.

2.3 Setup Steps

Load the SMSSPROG disk into drive A (the upper floppy disk drive) and SMSSSTUS into drive B (the lower one). Type cd a: to switch to drive A. Perform any of the following steps that are applicable to your configuration.

SETUP HARD DISK -- If you will run SMSS from hard disk, you must create a directory and copy all of SMSS to it. There is a routine on the SMSSPROG disk to do this. Type hardset to run this routine. HARDSET creates a subdirectory called SMSS and copies the SMSSPROG and SMSSSTUS disks to it.

SETUP SYSTEM PATH COMMAND -- SMSS is designed so that it can run from any subdirectory or disk. Therefore, SMART must be accessible from wherever SMSS is working. Type in the system command, path. The response will be a list of subdirectories which MS-DOS searches to find program names entered as commands. The directory, \smart, must be among this list. If it is not, change the system autoexec.bat file so that it includes \smart in its path command. See your MS-DOS manual for details on the path command and the autoexec.bat file.

SETUP VIRTUAL DISK -- If you are going to run SMSS with the virtual disk option described above, you must define an area of main memory to set aside for virtual disk. This is done in the system config.sys file. See the MS-DOS manual for a detailed discussion of virtual disk and the config.sys file. There is a routine on the SMSSPROG disk to create a proper config.sys. Type vdiskset to run this routine. To activate the newly defined virtual disk after vdiskset runs, you must re-boot the system by striking the "Ctrl", "Alt", & "Del" keys simultaneously (remove the floppy disks from their drives first). As long as this config.sys file is active, 180Kb of main memory will always be set aside for virtual disk. SMSS will run on the standard ACSC Z158 computer with this memory reserved, but some other very large applications may not.
SETUP CONFIG.SYS -- Whether or not you use virtual disk, your config.sys file must contain a "FILES=20" command and a "BUFFERS=20" command. Vdiskset inserts these commands for you. But, if you do not run vdiskset because you don't want to configure virtual disk, you must manually edit config.sys to insert both these commands.

Configuration
Parity: Even Odd None
Stop bits: 1 2
Plotter pen speed (1-10): 10

------------------ Graphics Display Screens ------------------
1 IBM Color Display Adapter 6 Apricot-xl
2 AT&T 6300 102 IBM EGA - RGB
3 IBM 3270 PC 103 IBM EGA - enhanced/monochrome
4 Hercules 110 Tandy 2000
5 STB Graphics Plus II(color) 201 IBM Monochrome

Graphics display screen number: 1
Paging file path:  c:\smartdat
Application data paths
  Spreadsheet/Graphics:  c:\smartdat
  Word Processor:  c:\smartword

= Data Manager: 

F1 Help  F2 Edit text  F3 Blank text  F10 Finished
configure

CONFIGURE - change configuration settings

Figure 2.1 SMART Configure Screen

SETUP SMART DATA PATH -- SMSS is designed to operate from whichever directory it is called from. For example, if you are running from floppy disk, you would first enter cd a: to make drive A the active directory before calling SMSS. Similarly, if you wanted to run from hard disk, you would first enter cd \sms to activate the SMSS subdirectory. But, on most ACSC Z158 computers, SMART is set up to look to the floppy drive A only for database data. Therefore, you must first use the SMART "configure" command to set up SMART
correctly. Call SMART and blank out the Data Manager application data path by keying the following commands:

```
smart (Enter) Calls SMART main menu
3 Selects command line 3
C Selects the "Configure" command
↓ (Space) (Enter) Hit the space bar to blank out the "Data Manager" field
F10 F10 function key to exit "Configure"
F10 F10 function key to exit main menu
Q Select "Quit" command to exit SMART
```

Blanking the Data Manager application data path tells SMART to get data base data from whichever subdirectory is current when SMART is called. See the SMART system manual for details of the "configure" command.

2.4 Running SMSS.

To run SMSS, you must be "in" the directory where the SMSS programs reside. In other words, if you previously loaded the system to a hard disk subdirectory called SMSS, you must first use the MS-DOS "chdir", or "cd" (for change directory) command to make SMSS the current directory. Once there, simply type the command, `smss`. Here's an example:

```
 cd c:\smss (Enter) Changes to the hard disk subdirectory, SMSS
smss (Enter) Calls SMSS
```

If you're running SMSS from the floppy disk, first load the SMSSPROG diskette in drive A, then type:

```
a: (Enter) Changes to the floppy drive
smss (Enter) Calls SMSS
```

2.4.1 Disk Option Screen

In either case, control will pass to the SMSS project file in SMART and the screen shown in figure 2.2 will appear. This screen is typical of the menu screens that appear throughout SMSS. In this one, you are asked to tell SMSS what disk option you are using: floppy, hard, or floppy with virtual disk. To select an option, move the arrow to your choice with the "↑" (up arrow) or "↓" (down arrow) keys, then press the "Enter" key. You can also press the "Esc" key to choose the default (hard disk) option and continue to the next screen. To get helpful information about this menu displayed on the screen, you can press the (upper or lower case) "H" key for help. Help screens are described in detail in paragraph 2.4.3 below.

6
2.4.2 SMSS Main Menu.

After indicating your disk configuration, you'll see some student data records flash on the screen as SMSS loads the student and other files. Note that in the lower right hand corner of the display you see messages such as, "LOADING STUDENT FILE". These are simply information messages to let you know that SMSS is performing some action and is not waiting for keyboard input. The next menu to appear is the main menu as shown in figure 2.3. Its three options: "File Maintenance", "Perform Mixing", and "Produce Reports" are major subsections of SMSS. Each is described in subsequent chapters. Every menu in SMSS looks essentially like this one and is subject to the following general menu option selection rules.

2.4.3 Menu Selection Rules.

When selecting options from an SMSS menu, the only valid keyboard inputs are the up-arrow (↑), down-arrow (↓), "Enter", "H" (upper or lower case), and "Esc" keys. All other keys will cause an audible warning beep. Here are the meanings of the valid keys:

**UP OR DOWN ARROW** -- Use these keys to move the display pointer up or down to the option you want to select. The pointer will wrap around if moved below the last option or above the first option.
ENTER -- Striking the "Enter" key selects whichever option
the display pointer is pointing to.

ESC -- This key causes SMSS to "pop back" to the last menu
displayed. "Escaping" from the main menu (figure 2.3) is the
way to exit from SMSS. Exit is to the SMART data manager.
From there, an "F10" and "Q" (for "quit") will exit back to
MS-DOS. The one exception to the "pop back" rule is the disk
option menu (figure 2.2). Escaping from that menu pops
forward to the main menu.

H -- The "H" (or "h") key calls for a help screen. Each menu
in SMSS has an associated help screen that explains the
option selections. The next paragraph describes how to
manipulate the help screens.

2.4.4 Help Screens.

SMSS is somewhat self-documenting since you can call up
information explaining each menu option by pressing the "H" key.
Figure 2.4 is an example of a help display for the main menu. If
there is more than one page to the help display, move back and
forth between pages with the "Pg Dn" and "Pg Up" keys. You can
print the screen by holding down the "Shift" key and pressing "Prt
Sc". Return to the menu with the "Esc" key. These instructions
are printed at the bottom of the boxed area on each help display.
Help displays are implemented in SMSS as screens of a dummy data
base called HELP. When you press the "H" key, SMSS performs a SMART "update" command on this data base. Since there is no data in HELP, the update, and hence, the SMART function key operations (F2,F3,...,F10) displayed at the bottom (below the double line ) of screen are meaningless. Use only the commands defined above.

Figure 2.4 Example Help Display
Chapter Three

File Maintenance

This chapter describes the functions that are performed in the file maintenance section of SMSS. File maintenance functions are those used to build and maintain the student and environment files. When you select FILE MAINTENANCE from the main menu, figure 3.1 will appear on the screen. The eight functions shown are described in the subsequent paragraphs. Exit the file maintenance section by pressing the "Esc" key at the file maintenance menu. Control will pass back to the main menu.

**FILE MAINTENANCE**

**OPTIONS**

1. BUILD STUDENT RECORDS FROM ANOTHER DISK
2. ADD STUDENT RECORDS MANUALLY
3. EDIT EXISTING STUDENT RECORD
4. DELETE STUDENT RECORD
5. DESIGNATE STUDENT NUMBERS
6. EDIT ENVIRONMENT RECORD (MIXING RULES)
7. DELETE IO'S (FOR MIX 3)
8. POST MIXER TENTATIVE ASSIGNMENTS

Enter h for HELP, Esc to exit to last menu

Figure 3.1 File Maintenance Menu

3.1 Build Student Records from Another Disk

This operation builds much of the student file and is generally the first SMSS function to be performed prior to the start of each academic year. The personnel office at Headquarters, Air
University (AU) can provide data on each active duty Air Force student projected into ACSC. Contact AU/DPXM at extension 6272 to request this data. They have direct access into the central Air Force personnel computer at the Air Force Military Personnel Center, but you should give them a week or two lead time. SMSS expects this data on a single file on a floppy diskette in the ASCII data format discussed in the SMART Data Base Manager Reference Guide, page Read-1. Note that only fields 1 through 27 (see Appendix One) of the student file are read in from floppy disk. All other fields are computed by the system or entered manually.

---

**SMSS - BUILD STUDENT FILE** menu level 111

This option allows you to build the student file by reading student records from an ASCII floppy disk file. See the SMSS users manual for the correct format of this file. You can build the student file from scratch or you can add new records to the end of the existing file.

****WARNING****

If you build from scratch, any records in the current student file will be lost.

OPTIONS

=> 1. BUILD STUDENT FILE FROM SCRATCH
2. ADD RECORDS TO STUDENT FILE

Enter h for HELP, Esc to exit to last menu

---

Figure 3.2 Build Student File Menu

After you select the BUILD STUDENT RECORDS option from the SMSS file maintenance menu, you will see figure 3.2. This menu gives you the option of either totally starting from scratch or adding new student records from the floppy disk to the student file. If you choose the first (start from scratch) option, all existing records will be erased. Note -- you can use the "copystu" command described in paragraph 6.1 to save the previous data base if necessary. Next, SMSS will ask you for the name of the file on the floppy diskette. If you don't know, back out of SMSS and use the MS-DOS "dir a:" command to display the diskette directory. After one more warning message to make sure you understand what you're doing, SMSS will ask you to load your diskette into drive
A, read the diskette and build new records into the student file. If you are running SMSS from floppy disk, there will be some disk switching ordered -- just follow the directions on the screen. This operation takes several minutes and is completed when the file maintenance menu reappears.

### 3.2 Add Student Records Manually

Unfortunately, the whole student file cannot be built with the option of paragraph 3.1. AU/DPXM can only give us automated data on active duty Air Force students. Information on international officers, other service officers, civilians, guardsmen, and reservists must be entered manually from the keyboard. Choose this option to type in one or more student records. A data entry form like the one in figure 3.3 will appear and the cursor will move to the first field on the form. You simply type the correct information into each field. See Appendix One for the meaning and data ranges of each field. You cannot enter anything into the last column, "COMPUTED FIELDS", since these values are computed by the system based on other information in the record.

---

### Figure 3.3 Student Record Data Entry Form

![Student Record Data Entry Form](image)
SMSS uses the SMART "enter" command to implement this option. See the SMART Base Manager Reference Guide, page Enter-3 for the meaning of the many helpful movement and editing keys, some of which are listed at the bottom of the screen. When you are done entering records, press the "F10" function key to return to the file maintenance menu.

3.3 Edit Existing Student Record

Select this option to change any of the fields in an existing student record. You will see the menu shown in figure 3.4, which will ask you to identify a particular student record to edit. You can select a record by name, social security number (SSAN), or student number (we'll assign student numbers later). Choose one and a box will appear in which to enter the search value. You need not enter a complete name, but the SSAN or student number must be complete. SMSS will find the record and display it in a form like that of figure 3.3. If the record is not found, you'll see an error message and control will return to the file maintenance menu. If found, the SMART "update" command is used to edit the record. Use the editing keys described in the SMART Data Base Manager Reference Guide, page Enter-3. When finished, press the "F10" function key to save the changes and return to the file maintenance menu. Note that you cannot edit the NAME, SS'N, or STU.NO fields since these are data base key fields. If you must change these fields, use SMART to load the STUDENTS file with screen, STUDENT2; then "update" to edit fields and update keys.

Choose which record to process by one of the following fields.
NOTE: SSAN & STUDENT NUMBER options are much faster.

OPTIONS

1. NAME (or partial name)
2. SSAN
3. STUDENT NUMBER

Enter h for HELP, Esc to exit to last menu

Figure 3.4 Record Selection Menu
3.4 Delete Student Record

Use this option to delete a record if a student disenrolls or you enter an erroneous record. You'll see the menu shown in figure 3.4 which will ask you to identify a particular student to delete. Recommend you use SSAN or student number to be precise. If that record is not found you'll see an error message and control will pass to the file maintenance menu. If found, the record is deleted using the SMART "delete" command. Note that deleted records are not actually physically dropped from the student file. You can recover a deleted record by manually using the SMART "delete" command.

3.5 Designate Student Numbers

ACSC uses student numbers internally to identify students. Once you've created the entire student file, use this option to assign student numbers. The file is ordered alphabetically and sequential multiples of five (0005, 0010, 0015,...) are generated into the STU.NO field. If you add students after performing this operation, run it again to regenerate numbers. Once numbers are issued to students, however, you should not run this again, since everyone's assigned number will change. Note that you cannot do any mixing before student numbers are assigned because the mixing program identifies students by student number. You'll get a warning message to make sure you really want to perform this operation; then, when completed, the file maintenance menu will appear.

3.6 Edit Environment Record

SMSS maintains an environment record (in the ENVIRON file) used to pass control information to the mixing program. With this record you tell the mixer:

- Which mix (1, 2 or 3) to process
- Relative priorities of mixing rules
- Organizational structure of the school

You must have edited the environment record before mixing. There are four pages to the environment record data entry form shown in figures 3.5, 3.6, 3.7, and 3.8. When you choose this option, SMSS uses the SMART "update" command and displays page one on the screen. Enter or change the following fields depending on the mix environment you desire:

MIX NUMBER -- The mixer uses different algorithms depending on which of the three annual mixes it is working on. Enter a value of 1, 2, or 3 here.

MAX NUMBER OF PREVIOUS SEMINAR MATES -- The mixer avoids assigning together students who have previously been assigned together in
past seminars. Enter here the number of previously-assigned-together students you want to allow in each seminar. The number must be from 2 through 9. If the mixer is forced to violate this rule, it will issue a warning message.

**SYSTEM MIXING RULES AND PRIORITIES** -- Students are mixed according to a set of adjustable mixing rules. The rules shown on page 1 of the environment record (figure 3.5) are system-wide rules as follows:

- **ARI** -- Airpower Research Institute students should not change seminars. Enter 9 to honor this rule. Any other value (from 0 to 8) will be treated as a student mixing rule (see below), and ARI students will be mixed like any other student attribute.

- **SOS** -- Squadron Officer School instructors should not change seminars. (9 to honor, any other to mix SOS instructors)

- **10** -- International Officers should not change seminars. (9 to honor, any other to mix 10's)

- **BUDDIES** -- How important is it (which priority) that the number of students in the MAX-NUMBER-OF-PREVIOUS-MATES field not be assigned together? This feature is currently not implemented, and is treated as a 9 (highest) priority regardless of what you enter here.

- **SEMCHANGE** -- All other students (not ARI, SOS, 10) must change seminars. (9 to honor, any other to ignore). This feature is not currently implemented. All other students will always change seminars.

**STUDENT MIXING RULES AND PRIORITIES** -- The mixing program’s primary task is to distribute students equitably among seminars according to attributes and skills each student possesses. There are 23 such attributes shown in figures 3.6 and 3.7. The priorities you assign to each attribute define how the mixer will operate. It first “spreads-out” all the students with priority nine attributes, then the eights, and so on down through priority ones. Thus it is more likely that students with high priority attributes will be distributed evenly among seminars. Students with low priority attributes are more likely to “bunch up” together. For example, if the school determines that it is very important that pilots are spread out among seminars, give the PILOT attribute rule a relatively high priority number. Similarly, if it is less important that students with PPBS experience are uniformly distributed, give the PPBS attribute a relatively low priority number. The actual values from 9 down to 1 are unimportant -- only the relative values among the priorities are important. A priority of 0 (zero) means, “do not consider this
attribute at all in mixing". By experimenting with different priority relationships, you can adjust the mixer output.

The five "TBD" (To Be Determined) fields in figure 3.7 allow the system to respond to changing school mixing requirements. For example, assume that in the future, the school determines that it is important to spread out students with joint organization experience. After you identify these students, you can use the EDIT STUDENT RECORD option to mark these records with a "Y" in the TBD1 field. Then, edit the environment record to assign that attribute a mixing priority and enter "Joint Experience" in the TBD1 description field to indicate what TBD1 is being used to represent. As long as the TBD fields are not being used, their priorities should be zero.

Figure 3.5 Environment Record Page 1

SEMINAR STRUCTURE -- The fourth page (figure 3.8) of the environment record is used to define the organization of the
school. Enter here the first (lowest numbered) and last (highest numbered) seminar numbers in each squadron. If any seminars are missing from that range, (as they might in mix 3 after the international officers depart), you can designate up to three missing seminar numbers.

Finally, after the environment record is set like you want it, press the "F10" function key to save the changes and return to the file maintenance menu.

Figure 3.6 Environment Record Page 2
<table>
<thead>
<tr>
<th>RULE</th>
<th>PRIORITY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRATOPS</td>
<td>9</td>
<td>Bomber or missile exp</td>
</tr>
<tr>
<td>TACOPS</td>
<td>9</td>
<td>Fighter pilot or WSO</td>
</tr>
<tr>
<td>PPBS</td>
<td>9</td>
<td>PPBS experience</td>
</tr>
<tr>
<td>ACQLOG</td>
<td>9</td>
<td>Acq/Log experience</td>
</tr>
<tr>
<td>SPACE</td>
<td>6</td>
<td>Space experience</td>
</tr>
</tbody>
</table>

Following Can Be Defined Later if needed

TPD1 0 unassigned
TPD2 0 unassigned
TPD3 0 unassigned
TPD4 0 unassigned
TPD5 0 unassigned

Next Page is for designating school structure... hit Pg Dn

Figure 3.7 Environment Record Page 3
SMSS can account for a variable number of seminars in each of 4 Squadrons. Enter the first (lowest numbered) and last (highest numbered) seminar numbers below for each squadron. Also, you can designate up to 3 missing seminars per squadron. Missing seminars must be within the first/last range.

<table>
<thead>
<tr>
<th>SQUADRON</th>
<th>FIRST SEMINAR</th>
<th>LAST SEMINAR</th>
<th>MISSING SEMINARS (if any)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3821</td>
<td>1_</td>
<td>11</td>
<td>1_ 0_ 0_</td>
</tr>
<tr>
<td>3822</td>
<td>12</td>
<td>22</td>
<td>22 0_ 0_</td>
</tr>
<tr>
<td>3823</td>
<td>23</td>
<td>33</td>
<td>25 0_ 0_</td>
</tr>
<tr>
<td>3824</td>
<td>34</td>
<td>44</td>
<td>40 0_ 0_</td>
</tr>
</tbody>
</table>

*****END*****

3.7 Delete International Officers

International officers (10's) graduate before the end of the academic year. Choose this option to delete 10 records after they graduate. After a warning message to double check that you really want to do this, the operation is completed when you see the file maintenance menu reappear.

3.8 Post Mixer Tentative Assignments

The mixing program only makes tentative seminar assignments. Thus, after a mix, you can examine the mixing results and rerun it or make manual adjustments, if necessary. MIXX is the tentative seminar assignment field in the student file. When the mix is acceptable, use this option to move MIXX values into MIX1, MIX2, or MIX3, thereby making the mix assignment permanent. Also, the mixer sets fields SLX and ASLX to a value of "Y" if the student would be a seminar leader or assistant seminar leader, respectively, in the new mix. This option updates fields SL and
ASL based on SLX and ASLX, as well. After selecting the option, you're asked to enter a value of 1, 2, or 3 to indicate which mix to post.
Chapter Four

Perform Mixing

When the student file has been built and the environment record reflects the proper mixing arguments, you are ready to mix the students into their seminar assignments. Mixing is accomplished by selecting option 2, the PERFORM MIXING option, from the main menu of figure 2.3.

4.1 The SMART/Mixer Interface

As mentioned in Chapter One, SMART is not capable of performing the complex operations of student mixing. Therefore, SMSS contains a mixing program (the mixer) written in GW-BASIC which operates directly under MS-DOS. The mixer cannot read the student file or environment record directly since these are in a special SMART data base manager format. Therefore, we use the SMART "write" command to create two files, STU.ASC and ENV.ASC, which contain selected student and environment information in a form that is readable by the mixer. When you select the PERFORM MIXING option,

```
SMSS MIXING OPTION

If there have been no changes to the student file since the last mixing run, you can skip the ASCII file update. If there have been changes, or you don't know, be safe by running the ASCII file update.

OPTIONS

⇒ 1. SKIP ASCII FILE UPDATE
   2. RUN ASCII FILE UPDATE

Enter h for HELP, Esc to exit to last menu
```

Figure 4.1 Mixing Option Menu
SMSS creates these files and passes control to the mixer, outside of SMART. The mixer then produces a results file called STUOUT.ASC and passes control again to SMART, which updates the student file with the mix results in STUOUT.ASC.

4.2 SMSS Mixing Menu

Select the PERFORM MIXING option from the main menu and you will see the two-option mixing menu shown in figure 4.1. SMSS asks you here if you want to skip the creation of the STU.ASC file. If you have mixed before and have since made no changes to the student file, you can save a little time by not writing STU.ASC since it will not have changed since the last time you mixed. In this case, select option 1, SKIP ASCII FILE UPDATE. Otherwise, select option 2, RUN ASCII FILE UPDATE.

4.3 Mixing Sequence of Operations

From this point on, mixing is essentially a "hands-off" operation that requires no user input. You can track the following operations by watching the explanatory messages at the bottom of the display screen.

1. SMART sorts the student file by rank and date of rank.

2. SMART writes ENV.ASC, the environment record interface file for the mixer.

3. If you requested it, SMART writes the student interface file, STU.ASC.

4. SMART relinquishes control and the mixer is called.

5. The mixer asks whether or not to turn off trace output. Trace output "on" will cause a trace of mixing events to be displayed on the screen. Mixing performed.

6. If there is an abort exit from the mixer, the whole process stops in MS-DOS.

7. If successful exit, control passes to SMART. SMART reads the output interface file (STUOUT.ASC) into a data manager file called NEWMIX.

8. Student file fields, MIXX, SLX, and ASLX are updated from NEWMIX using the SMART "transaction" command.

9. Control passes to the main menu.

This whole process will take several minutes, so be patient. When complete, the MIXX field of each student record will contain the tentative seminar number of assignment. SLX will contain "Y" if
the student is designated a Seminar Leader in the new mix. ASLX will be "Y" if he/she is designated as Assistant Seminar Leader. These tentative assignments are not made permanent until you perform the POST MIXER TENTATIVE ASSIGNMENTS option of paragraph 3.8.
Chapter 5

Produce Reports

SMSS produces two types of reports:

1. Mixing summary reports that show the results of a mix.
2. Alphabetically sorted student rosters.

The mixing summary reports are usually produced after you have mixed the students. These summaries show whether you have produced a "balanced" distribution. If not, you can make manual changes to the tentative seminar assignments, or you can re-mix after adjusting mix rule priorities as in paragraph 3.6. The alpha rosters can be run at any time to support school operations.

Figure 5.1  Report Mix Selection Menu

5.1 Report Selection Menus

When you select the PRODUCE REPORTS option from the main menu, figure 5.1 will appear. Here, you simply tell SMSS which mix you will be reporting on. Remember that MIXX is the tentative seminar
assignment field. Generally, you will choose MIXX when you want the results of the last mixing run. After choosing a mix, you will see the reports selection menu of figure 5.2. Choose which report you want from this menu. The rest of the paragraphs in this chapter deal with each of these report options.

SMSS REPORTS OPTION

OPTIONS

1. SCHOOL MIXING SUMMARY
2. SQUADRON MIXING SUMMARY
3. SEMINAR MIXING REPORT
4. COMPLETE MIXING REPORT (all the above)
5. SCHOOL ALPHA ROSTER
6. SQUADRON ALPHA ROSTER
7. SEMINAR ALPHA ROSTER

Enter h for HELP, Esc to exit to last menu

Figure 5.2 Reports Selection Menu

The mixing summaries (options 1 and 2) each produce one page of output and are always directed to the printer. For all other reports, you will see the Report Output Selection Menu shown in figure 5.3. Choose whether you want the output sent to the printer, a disk file, or the display screen. If you choose "DISK", you will be prompted to enter a file name. Enter a single file name or complete path name, but do not include a file extension since SMART appends the extension, "PRT", to the name you enter. This option is implemented using the SMART "report print" command. See the SMART Data Base Manager Reference Guide, page Report-25, for details. With the "DISK" option, you can later print the file you produce with the MS-DOS "print" command. Before you print, however, enter COMPRES to put the printer in compressed text mode. COMPRES.EXE is a program supplied on the SMSSPROG disk.
REPORT OUTPUT
SELECTION

Choose Where you want report output sent

OPTIONS

1. PRINTER
2. DISK
3. SCREEN

Default (Esc) is PRINTER

Enter h for HELP, Esc to exit to next menu

5.2 School Mixing Summary

This is a one-page report (figure 5.4) that totals the 23 student attributes by squadron. Each line represents a student attribute defined in the environment record. Use this report when developing mix one to determine if each squadron has a reasonable distribution of students. This report only goes to the printer.

5.3 Squadron Mixing Summary

This report (figure 5.5) is similar to the School Mixing Summary, but it displays a column for each seminar in a particular squadron. After selection, SMSS will ask for a squadron number -- Enter 1, 2, 3, or 4. Use this report to evaluate the distribution of students into seminars across a squadron. This report only goes to the printer.

5.4 Seminar Mixing Report

The Seminar Mixing Report for a single seminar is shown in figure 5.6. There is one line for each student in the seminar and each column represents one of the student mixing attributes, printed at the top of each page. A value of "1" in any column indicates that the student possesses that attribute. Students are listed in seminar leader selection order. That is, the first student who has not previously been a seminar leader (indicated by a "Y" in the SL field) would be the seminar leader in this mix. When you choose this option, SMSS asks you for the seminar(s) to report on. Your answer can be one of three options. Enter,
ALL -- to list all seminars in the school.

SQn (where n = 1, 2, 3, or 4) -- to list all the seminars in a particular squadron.

Seminar number (e.g., 33) -- to list a single seminar.

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>SQ1</th>
<th>SQ2</th>
<th>SQ3</th>
<th>SQ4</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Line AD Air Force</td>
<td>102</td>
<td>101</td>
<td>104</td>
<td>103</td>
<td>410</td>
</tr>
<tr>
<td>2. No advanced Degree</td>
<td>49</td>
<td>46</td>
<td>37</td>
<td>41</td>
<td>173</td>
</tr>
<tr>
<td>3. HQ (&gt; = Majcom) Experience</td>
<td>80</td>
<td>78</td>
<td>83</td>
<td>76</td>
<td>317</td>
</tr>
<tr>
<td>4. Any Pilot</td>
<td>48</td>
<td>52</td>
<td>51</td>
<td>50</td>
<td>201</td>
</tr>
<tr>
<td>5. Any Navigator</td>
<td>17</td>
<td>17</td>
<td>18</td>
<td>14</td>
<td>66</td>
</tr>
<tr>
<td>6. Anyhow Unaccompanied</td>
<td>29</td>
<td>23</td>
<td>34</td>
<td>18</td>
<td>104</td>
</tr>
<tr>
<td>7. USAFA (academy) grad</td>
<td>25</td>
<td>18</td>
<td>22</td>
<td>16</td>
<td>81</td>
</tr>
<tr>
<td>8. Army</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>44</td>
</tr>
<tr>
<td>9. Res, ANG, Civ, USMC, or Navy</td>
<td>11</td>
<td>11</td>
<td>9</td>
<td>10</td>
<td>41</td>
</tr>
<tr>
<td>10. Black or other</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>11. Female</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>12. Current captains</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>13. Medical, legal, chaplain</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>14. Bomber or missile exp</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>22</td>
</tr>
<tr>
<td>15. Fighter pilot or WSO</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>16. PPBS experience</td>
<td>26</td>
<td>34</td>
<td>28</td>
<td>22</td>
<td>110</td>
</tr>
<tr>
<td>17. Acc Log experience</td>
<td>31</td>
<td>32</td>
<td>15</td>
<td>34</td>
<td>112</td>
</tr>
<tr>
<td>18. Space experience</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 5.4 School Mixing Summary
### SQUADRON 3 MIX STATISTICS

**MIX 1**

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Line AD Air Force</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>104</td>
</tr>
<tr>
<td>2. No advanced Degree</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>37</td>
</tr>
<tr>
<td>3. HQ (&gt;=Majcom) Experience</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>83</td>
</tr>
<tr>
<td>4. Any Pilot</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>51</td>
</tr>
<tr>
<td>5. Any Navigator</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>6. Anyhow Unaccompanied</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>7. USAFA (academy) grad</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>8. Army</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>9. Res,ANG,Civ,USMC,or Navy</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>10. Black or other</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>11. Female</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>12. Current captains</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>13. Medical/legal/chaplain</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>14. Bomber or missile exp</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>15. Fighter pilot or WSO</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>16. PPBS experience</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>17. Acq/Log experience</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>18. Space experience</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23. unassigned</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 5.5  Squadron Mixing Summary

### 5.5 Complete Mixing Report

This option is an "all-the-above" report. It runs the School Summary, Squadron Summary for all squadrons, and Seminar Mixing Report for all seminars. It is convenient (although time-consuming) to run this after mixing to get a complete mixing picture. Note that if you choose "DISK" output, the two summary reports will be printed while the Seminar Mixing Report will go to disk.

### 5.6 Alpha Rosters

The final three report options all produce alphabetically sorted rosters similar to that shown in figure 5.7. You can create a school, squadron, seminar, or set of seminars roster. Note that these rosters contain students' name and social security number.
so they should be protected according to the privacy act. Social security numbers have been deleted from the example.

**Figure 5.6 Seminar Mixing Report**

![Table Image]
<table>
<thead>
<tr>
<th>STUDENT NUMBER</th>
<th>NAME</th>
<th>SSN</th>
<th>RANK</th>
<th>DOR</th>
<th>COMMAND</th>
<th>PAFSC</th>
<th>COMPONENT</th>
<th>AERODROME RATING</th>
<th>MIXT</th>
<th>MIXE</th>
<th>MIXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3001</td>
<td>ADAMS TIMOTHY M</td>
<td>351200</td>
<td>04</td>
<td>ELM</td>
<td>2025</td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>21</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>1076</td>
<td>AHMED</td>
<td>061981</td>
<td>14</td>
<td>ATC</td>
<td>0950</td>
<td>USAF</td>
<td>SR NAV</td>
<td></td>
<td>21</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>1776</td>
<td>BERRIATT SIMON E</td>
<td>351800</td>
<td>04</td>
<td>HAF</td>
<td>7016</td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>21</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>1595</td>
<td>BOLTZ JON D</td>
<td>351700</td>
<td>14</td>
<td>ATC</td>
<td>22950</td>
<td>USAF</td>
<td>SR NAV</td>
<td></td>
<td>31</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>1913</td>
<td>CIUSKEL MARTIN E</td>
<td>350161</td>
<td>04</td>
<td>MAF</td>
<td>7016</td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>21</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>3610</td>
<td>DUILLY MARTIN E</td>
<td>350161</td>
<td>04</td>
<td>HAF</td>
<td>7016</td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>21</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>1295</td>
<td>HOLTZ ROBERT E</td>
<td>350161</td>
<td>04</td>
<td>USAF</td>
<td></td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>31</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>1036</td>
<td>JENSEN GORDON R</td>
<td>350161</td>
<td>04</td>
<td>USAF</td>
<td></td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>31</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>1168</td>
<td>JENSEN RICHARD M</td>
<td>350161</td>
<td>04</td>
<td>USAF</td>
<td></td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>31</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>1248</td>
<td>ROBERTS ELIZABETH M</td>
<td>350161</td>
<td>04</td>
<td>TAC</td>
<td>61716</td>
<td>USAF</td>
<td>NO RATING</td>
<td></td>
<td>21</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>1295</td>
<td>ROSS MURIEL D</td>
<td>350161</td>
<td>04</td>
<td>SAC</td>
<td>1435Y</td>
<td>USAF</td>
<td>SR PILOT</td>
<td></td>
<td>21</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>2740</td>
<td>RUSSELL MICHAEL D</td>
<td>350161</td>
<td>04</td>
<td>TAC</td>
<td>61716</td>
<td>USAF</td>
<td>PILOT</td>
<td></td>
<td>21</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.7 Student Alpha Roster
Chapter 6

Mixmaster's Checklist

The preceding chapters described how to use SMSS. This chapter lists the mixing tasks required of the ACSC staff officer responsible for student mixing. Numbers in parentheses are paragraph numbers of the SMSS operation to perform each task.

ACTIONS

6.1 After graduation in June -- Save the old student file for historical purposes. In the SMSS directory, enter `copystu`. This will copy all student information to floppy disk.

6.2 Prior to start of new class -- Get the Air Force student disk from AU/DPXM and load it. (3.1)

6.3 Obtain hardcopy information on non Air Force students from the ACSC Director of Operations. Manually load these students. Don't forget SOS instructors and ARI fellows. (3.2 and appendix 1)

6.4 When all students have been loaded -- Assign student numbers. Note that you must assign student numbers before doing any mixing. (3.5)

6.5 The following student attributes/skills must be entered by hand. (3.3)

- MAR.STAT . . . . . "U" for married, unaccompanied
- SOS . . . . . "Y" for SOS phase 11 instructors
- ARI . . . . . "Y" for Airpower Research Inst. fellows
- PPBS, ACQLOG, SPACE . "Y" for possesses that skill

6.6 Enter "Y" in the CC field of the class commander. (3.3)

6.7 Enter "Y" in SRO field of the four squadron senior ranking officers. (3.3)

6.8 If there are any new mixing criteria, choose TBD fields to represent them. Identify students with the attribute/skill and update appropriate TBD field with "Y". (3.3)

6.9 Edit all the fields in the environment record for mix 1. (3.6)
6.10 If you want to pre-assign any student to a particular seminar, update his/her MIX1 field with that seminar number. (3.3)

6.11 Perform mixing for mix 1. (4.2)

6.12 Run the complete mixing report on MIXX and analyze it for acceptability. (5.5)

6.13 If the mix is somehow "uneven", adjust the rule attributes/skill priorities and mix again until it comes out right. (3.6, 4.2)

6.14 To make any manual mix adjustments, edit the MIXX field. Note that if you make any manual changes, you must manually compare dates-of-rank and manually make any necessary changes to the SLX and ASLX fields. (3.3)

6.15 When the mix is satisfactory -- post tentative mix assignments to MIX1. (3.8)

6.16 Run the Alpha Rosters. (5.6)

6.17 Make a backup copy of the student file to protect against loss of the working copy. On a hard disk system, run "copystu" (6.1). On a floppy system, copy the SMSSSTUS disk using the MS-DOS "diskcopy" command.

6.18 As necessary throughout the school year -- delete any disenrolled students (3.4), and make any personnel data changes for promotions, change in marital status, etc. (3.3)

6.19 Prior to mix 2 -- do steps 6.8 through 6.16 for mix 2.

6.20 Prior to mix 3 -- Delete International Officers (3.7), then do steps 6.8 through 6.16 for mix 3.
Appendix One

Student File

<table>
<thead>
<tr>
<th>NO.</th>
<th>NAME</th>
<th>LENGTH</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAME</td>
<td>27</td>
<td>A</td>
<td>Student's Name</td>
<td>any (e.g.) SMITH JOHN R</td>
</tr>
<tr>
<td>2</td>
<td>RANK</td>
<td>2</td>
<td>A</td>
<td>Rank/Pay Grade</td>
<td>01-15 (that's, zero-one)</td>
</tr>
<tr>
<td>3</td>
<td>SEX</td>
<td>1</td>
<td>A</td>
<td>Gender</td>
<td>M or F</td>
</tr>
<tr>
<td>4</td>
<td>RACE</td>
<td>3</td>
<td>A</td>
<td>Ethnicity</td>
<td>BLK, CAU or OTH</td>
</tr>
<tr>
<td>5</td>
<td>MAR.ST</td>
<td>1</td>
<td>A</td>
<td>Marital Status</td>
<td>D, M, S, U</td>
</tr>
<tr>
<td>6</td>
<td>SSAN</td>
<td>9</td>
<td>A</td>
<td>Social Security No.</td>
<td>0000000000 - 9999999999</td>
</tr>
<tr>
<td>7</td>
<td>DOB</td>
<td>6</td>
<td>N</td>
<td>Date of Birth</td>
<td>yymmdd (e.g.) 510922</td>
</tr>
<tr>
<td>8</td>
<td>COMP</td>
<td>5</td>
<td>A</td>
<td>Service Component</td>
<td>AFRES, ANG, CIV, USA, USAF, USMC, USG, USN or country indicator (e.g.) PAKIS</td>
</tr>
<tr>
<td>9</td>
<td>DOR</td>
<td>6</td>
<td>N</td>
<td>Date of Rank</td>
<td>yymmdd (e.g.) 860801</td>
</tr>
<tr>
<td>10</td>
<td>PLSD</td>
<td>6</td>
<td>N</td>
<td>Pay List Service Date (Total Service)</td>
<td>yymmdd (e.g.) 750815</td>
</tr>
<tr>
<td>11</td>
<td>COMM</td>
<td>7</td>
<td>A</td>
<td>Commissioning Source</td>
<td>ACAD MI, AF ACAD, DIRECT, DP CIV, DP MIL, OCS, OTHER, OTS, OTS DMG, ROTC, ROTC DG, ROTCDMG, UNKNOWN, USMA, USNA</td>
</tr>
<tr>
<td>12</td>
<td>AERO</td>
<td>9</td>
<td>A</td>
<td>Aeronautical Rating</td>
<td>MAST NAV, NAV, NAVIGATOR, NFO, NO RATING, NONE, PILOT, SR NAV, SR PILOT, (blank)</td>
</tr>
<tr>
<td>13</td>
<td>RTFD</td>
<td>4</td>
<td>N</td>
<td>Return to Fly Date</td>
<td>blank or yymm (e.g.) 9008</td>
</tr>
<tr>
<td>NO.</td>
<td>NAME</td>
<td>LENGTH</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>RANGE</td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>--------</td>
<td>------</td>
<td>--------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>14</td>
<td>MOF</td>
<td>3</td>
<td>A</td>
<td>Months of Flying</td>
<td>blank or nnn (e.g.) 085</td>
</tr>
<tr>
<td>15</td>
<td>AC1.HRS.DT</td>
<td>15</td>
<td>A</td>
<td>Most recent acft, hours, last flying date</td>
<td>aaaaaahhhhyymm (e.g.) F-5E/F 02168701</td>
</tr>
<tr>
<td>16</td>
<td>AC2.HRS.DT</td>
<td>15</td>
<td>A</td>
<td>Next most recent aircraft, hours, date</td>
<td>Same as field 15</td>
</tr>
<tr>
<td>17</td>
<td>AC3.HRS.DT</td>
<td>15</td>
<td>A</td>
<td>Third most recent aircraft, hours, date</td>
<td>Same as field 15</td>
</tr>
<tr>
<td>18</td>
<td>ED.LEVEL</td>
<td>4</td>
<td>A</td>
<td>Education Level</td>
<td>BAC, BAC+, DOC, MAS, MAS+, PHD, RN, 1PDG</td>
</tr>
<tr>
<td>19</td>
<td>DAFSC</td>
<td>6</td>
<td>A</td>
<td>Duty Air Force Specialty Code</td>
<td>(e.g.) K1115H, or 4916</td>
</tr>
<tr>
<td>20</td>
<td>PAFSC</td>
<td>6</td>
<td>A</td>
<td>Primary AFSC</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>21</td>
<td>AFSC2</td>
<td>6</td>
<td>A</td>
<td>Secondary AFSC</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>22</td>
<td>AFSC3</td>
<td>6</td>
<td>A</td>
<td>Tertiary AFSC</td>
<td>&quot; &quot; &quot;</td>
</tr>
<tr>
<td>23</td>
<td>PME1</td>
<td>5</td>
<td>A</td>
<td>Junior Service School Grad</td>
<td>(blank), ARMY, SOS, SOSC, SOSR</td>
</tr>
<tr>
<td>24</td>
<td>PME2</td>
<td>5</td>
<td>A</td>
<td>Intermediate Service School Grad</td>
<td>(blank), ACSC, ACSCC, ACSCS, CSC, MCCSC</td>
</tr>
<tr>
<td>25</td>
<td>PME3</td>
<td>5</td>
<td>A</td>
<td>Senior Service School Grad</td>
<td>(blank), MCCSC, OTHER</td>
</tr>
<tr>
<td>26</td>
<td>MAJCOM</td>
<td>3</td>
<td>A</td>
<td>Last Assigned Major Command, Special Operating Agency, or Joint Command/Agency</td>
<td>(blank), AAC, AAG, ACD, AFE, AFW, ATC, AUN, CBT, CMC, DMA, ELC, ELM, ESC, EUR, HAF, HRS, INT, ISC, LCT, LOG, MAC, MPC, MGS, OSI, PAF, RPC, SAC, SPC, SYS, TAC, TAP, TEC</td>
</tr>
<tr>
<td>27</td>
<td>ORG.LVL</td>
<td>3</td>
<td>A</td>
<td>Highest Organizational Level</td>
<td>(blank), ADV, DOD, HAF, MAJ, NAF, SOA, SQD, SQN, WNG</td>
</tr>
</tbody>
</table>

(Fields 28 thru 49 are entered by the mixing program or manually)

<table>
<thead>
<tr>
<th>NO.</th>
<th>NAME</th>
<th>LENGTH</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>SQUADRCN</td>
<td>1</td>
<td>A</td>
<td>ACSC Squadron</td>
<td>1, 2, 3, or 4</td>
</tr>
<tr>
<td>NO.</td>
<td>NAME</td>
<td>LENGTH</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>RANGE</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>--------</td>
<td>------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>29</td>
<td>MIX1</td>
<td>2</td>
<td>A</td>
<td>Seminar Assignment for 1st Mix</td>
<td>01 - 44</td>
</tr>
<tr>
<td>30</td>
<td>MIX2</td>
<td>2</td>
<td>A</td>
<td>2nd Mix Seminar</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>31</td>
<td>MIX3</td>
<td>2</td>
<td>A</td>
<td>3rd Mix Seminar</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>32</td>
<td>MIXX</td>
<td>2</td>
<td>A</td>
<td>Tentative Mix Assigned by mixer</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>33</td>
<td>STU.NO</td>
<td>4</td>
<td>A</td>
<td>Student Number</td>
<td>0005-9995</td>
</tr>
<tr>
<td>34</td>
<td>SOS</td>
<td>1</td>
<td>A</td>
<td>SOS Instructor</td>
<td>Y or blank</td>
</tr>
<tr>
<td>35</td>
<td>ARI</td>
<td>1</td>
<td>A</td>
<td>Air Power Research Institute Fellow</td>
<td>Y or blank</td>
</tr>
<tr>
<td>36</td>
<td>CC</td>
<td>1</td>
<td>A</td>
<td>Indicates Class Commander</td>
<td>Y or blank</td>
</tr>
<tr>
<td>37</td>
<td>SRO</td>
<td>1</td>
<td>A</td>
<td>Indicates Senior Ranking Officer in Squadron</td>
<td>Y or blank</td>
</tr>
<tr>
<td>38</td>
<td>SL</td>
<td>1</td>
<td>A</td>
<td>Has held Seminar Leader position</td>
<td>Y or blank</td>
</tr>
<tr>
<td>39</td>
<td>ASL</td>
<td>1</td>
<td>A</td>
<td>Has held Asst. Seminar leader posn.</td>
<td>Y or blank</td>
</tr>
<tr>
<td>40</td>
<td>SLX</td>
<td>1</td>
<td>A</td>
<td>Seminar Leader in tentative mix</td>
<td>Y or blank</td>
</tr>
<tr>
<td>41</td>
<td>SLX</td>
<td>1</td>
<td>A</td>
<td>Asst. Seminar Leader in tentative mix</td>
<td>Y or blank</td>
</tr>
<tr>
<td>42</td>
<td>PPBSK</td>
<td>1</td>
<td>A</td>
<td>Has PPBS skill</td>
<td>Y or blank</td>
</tr>
<tr>
<td>43</td>
<td>ACQLOGSK</td>
<td>1</td>
<td>A</td>
<td>Has Acquisition/Log skill</td>
<td>Y or blank</td>
</tr>
<tr>
<td>44</td>
<td>SPACESK</td>
<td>1</td>
<td>A</td>
<td>Has Space Ops skill</td>
<td>Y or blank</td>
</tr>
<tr>
<td>45</td>
<td>TBD1</td>
<td>1</td>
<td>A</td>
<td>Unassigned attribute</td>
<td>Y or blank</td>
</tr>
<tr>
<td>46</td>
<td>TBD2</td>
<td>1</td>
<td>A</td>
<td>Unassigned attribute</td>
<td>Y or blank</td>
</tr>
<tr>
<td>47</td>
<td>TBD3</td>
<td>1</td>
<td>A</td>
<td>Unassigned attribute</td>
<td>Y or blank</td>
</tr>
<tr>
<td>NO.</td>
<td>NAME</td>
<td>LENGTH</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
<td>RANGE</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>--------</td>
<td>------</td>
<td>--------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>48</td>
<td>TBD4</td>
<td>1</td>
<td>A</td>
<td>Unassigned attribute Y or blank</td>
<td>TBD4</td>
</tr>
<tr>
<td>49</td>
<td>TBD5</td>
<td>1</td>
<td>A</td>
<td>Unassigned attribute Y or blank</td>
<td>TBD5</td>
</tr>
</tbody>
</table>

Fields 50 thru 66 are calculated fields. That is, their values (Y or blank) are calculated by the system according to formulas defined in the data base. See SMART Data Base Manager Reference Guide, page Create-9 for details. Each field has a value of blank or "Y". The "Y" condition is shown.

<table>
<thead>
<tr>
<th>NO.</th>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>&quot;Y&quot; CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>10</td>
<td>International Officer</td>
<td>COMP = anything but US components. (i.e.) not AFRES, ANG, CIV, USA, USAF, USMC, USN, USG</td>
</tr>
<tr>
<td>51</td>
<td>USAF</td>
<td>Active Air Force</td>
<td>COMP = &quot;USAF&quot; (not ANG or AFRES)</td>
</tr>
<tr>
<td>52</td>
<td>NOMAST</td>
<td>No Master's Degree</td>
<td>ED.LEVEL shows no Master or higher</td>
</tr>
<tr>
<td>53</td>
<td>SORG</td>
<td>Sr. Orgn Experience</td>
<td>ORG.LVL shows MAJCOM or higher</td>
</tr>
<tr>
<td>54</td>
<td>PILOT</td>
<td>Pilot</td>
<td>AERO shows pilot</td>
</tr>
<tr>
<td>55</td>
<td>NAVIGATOR</td>
<td>Navigator</td>
<td>AERO shows nav</td>
</tr>
<tr>
<td>56</td>
<td>SINGLE</td>
<td>No spouse</td>
<td>Single, divorced or unaccompanied</td>
</tr>
<tr>
<td>57</td>
<td>USAFA</td>
<td>AF Academy grad</td>
<td>COMM = &quot;AF ACAD&quot;</td>
</tr>
<tr>
<td>58</td>
<td>ARMY</td>
<td>Army Officer</td>
<td>COMP = &quot;USA&quot;</td>
</tr>
<tr>
<td>59</td>
<td>OTHERCOMP</td>
<td>Not Army or USAF</td>
<td>COMP = AFRES, ANG, USMC, Navy, Coast Guard or Civilian</td>
</tr>
<tr>
<td>60</td>
<td>MINORITY</td>
<td>Ethnic Minority</td>
<td>RACE is non-caucasian</td>
</tr>
<tr>
<td>61</td>
<td>FEMALE</td>
<td>Female</td>
<td>SEX = &quot;F&quot;</td>
</tr>
<tr>
<td>62</td>
<td>CAPTAIN</td>
<td>Rank is Captain</td>
<td>RANK = &quot;03&quot;</td>
</tr>
<tr>
<td>63</td>
<td>NONLINE</td>
<td>Not a line officer</td>
<td>Civilian or PAFSC is Medical, Legal or Chaplain</td>
</tr>
<tr>
<td>64</td>
<td>STRATOPSK</td>
<td>Strategic Ops skills</td>
<td>Missiles or Bomber background</td>
</tr>
<tr>
<td>65</td>
<td>TACOPSK</td>
<td>Tactical Ops skills</td>
<td>PAFSC shows Fighter background</td>
</tr>
<tr>
<td>66</td>
<td>COULDBSL</td>
<td>Qualified to be Seminar Leader</td>
<td>Any US officer</td>
</tr>
</tbody>
</table>
Appendix Two

SMART Project Files
Program Listings

The following programs are written in the SMART project file programming language. See the SMART System Manual, Project Processing Guide for details.

<table>
<thead>
<tr>
<th>Project File Name</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS0</td>
<td>44</td>
<td>Entry and Main Routine. Displays most menus and calls other project file routines.</td>
</tr>
<tr>
<td>ENVWRITE</td>
<td>65</td>
<td>Writes environment record for mixer.</td>
</tr>
<tr>
<td>GOMIX</td>
<td>66</td>
<td>Returns to MS-DOS when mixer Requested.</td>
</tr>
<tr>
<td>IODEL</td>
<td>67</td>
<td>Deletes 10 records from student file.</td>
</tr>
<tr>
<td>LOADEM</td>
<td>68</td>
<td>Loads student and help screen files.</td>
</tr>
<tr>
<td>NEWMIX</td>
<td>69</td>
<td>Called upon return from mixer. Updates student file with tentative mix results.</td>
</tr>
<tr>
<td>RAMLOAD</td>
<td>71</td>
<td>Loads student file into Virtual disk.</td>
</tr>
<tr>
<td>RAMSAVE</td>
<td>72</td>
<td>Dumps student file from virtual disk to floppy.</td>
</tr>
<tr>
<td>STUNUMB</td>
<td>73</td>
<td>Assigns student numbers.</td>
</tr>
<tr>
<td>STUREAD</td>
<td>74</td>
<td>Reads student file data from ASCII floppy disk.</td>
</tr>
<tr>
<td>STUWRITE</td>
<td>76</td>
<td>Writes student file in ASCII format for mixer.</td>
</tr>
</tbody>
</table>
**Project File SMSS**

**COMMENT** ************************************************************************************************************

**COMMENT** ENTRY POINT FOR MAIN SMSS PROGRAM
**COMMENT** WRITTEN BY MAJ RICHARD M. JENSEN

**COMMENT** VARIABLE & PARAMETER CONVENTIONS:
**COMMENT** STANDARD PROJECT VARIABLES (TEXT1, TEXT2, VALUE1, VALUE2) HAVE ONLY A
**COMMENT** TEMPORARY SCOPE. ALL OTHER VARIABLE NAMES (#name) HAVE SPECIFIC
**COMMENT** USES AND ARE NOT USED FOR MORE THAN ONE PURPOSE.
**COMMENT** PARAMETERS %1-%8 HAVE TEMPORARY SCOPE. %9 AND %0 HAVE SPECIFIC USES
**COMMENT** (%9 INDICATES CHOSEN MIX # IN REPORTS SECTION, %0 INDICATES MEDIA
**COMMENT** SELECTION THROUGHOUT)

**COMMENT** ---------------------------------------------------------------------------------------------------------------

**COMMENT** HOUSEKEEPING AND ONE TIME SETUP
**QUIET** on
**singlestep** off
**COMMENT** DELETE ENV.ASC IF IT HAPPENS TO EXIST
**COMMENT** IN CASE WE EXIT ABNORMALLY, DON'T WANT SMSS.BAT TO THINK WE'RE MIXING
if file("env.asc")=1.0 then file erase env.asc
**COMMENT** OPEN THE ENVIRONMENT FILE
load environ screen environ
**COMMENT** ENTRY TO SMSS FROM SMSS.BAT COULD BE FROM ONE OF 2 PLACES:
**COMMENT** 1. INITIAL ENTRY (PRE-MIXING) OR
**COMMENT** 2. AFTER MIXING
**COMMENT** EXISTENCE OF MIXER OUTPUT FILE, STUOUT.ASC IS INDICATOR OF WHICH.
**COMMENT** NOTE, MIXER CREATES STUOUT.ASC. TEST FOR STUOUT.ASC
**COMMENT** IF IT EXISTS GO COPY THE TENTATIVE MIXING INFO INTO THE DATA BASE
**COMMENT** IF NOT ASK USER WHICH MEDIA HE'S OPERATING FROM
if file("stuout.asc")=1.0
  execute newmix in-memory
else
  call getmedia
**COMMENT** GETMEDIA AND NEWMIX RETURN #MEDIA AND %0 TO INDICATE USER'S CHOICE
**COMMENT** OF MEDIA. DON'T USE %0 FOR ANYTHING ELSE
endif

**COMMENT** main menu (level 1) loop
**label** main
**COMMENT** DISPLAY WELCOME TITLE
call maintitl
let $lvl=1
**menu print** 8 36 15 4 OPTIONS
**menu print** 10 24 15 4 1. FILE MAINTENANCE
**menu print** 12 24 15 4 2. PERFORM MIXING
**menu print** 14 24 15 4 3. PRODUCE REPORTS
**COMMENT** GET MENU SELECTION
#$firstline=10

44
***** Project File SMSS *****

$lastline=14
$linediff=2
$colm=23
label inloop1
call selopt
COMMENT DEBUG SINGLESTEP ON
COMMENT TEST INPUT
if $option = 1
   call preprocess
   jump main
elseif $option= 2
   call mixem
   jump main
elseif $option= 3
   call reports
   jump main
COMMENT TEST FOR H
elseif $key= 104 or $key=72
   call help
   jump main
COMMENT TEST FOR ESCAPE MEANING EXIT
elseif $key= 27
COMMENT EXISTENCE OF FILE ENV.ASC, THE ASCII ENVIRONMENT FILE, IS THE KEY
COMMENT WHETHER EXIT BACK TO DOS IS FROM HERE OR FROM A MIXING REQUEST.
COMMENT ERASE ENV.ASC TO SHOW WE'RE EXITING NORMALLY
   menu clear 15 1
   if file("env.asc")=1.0 then file erase env.asc
   unload all
COMMENT IF WE'VE BEEN WORKING ON VDISK, IT NEEDS TO BE SAVED TO FLOPPY
   if $media=3 then execute ramsave in-memory
end
else beep on
   beep 3 illegal entry--try again
   jump inloop1
endif

COMMENT************************LEVEL 1 SUBROUTINES************************

COMMENT ***********************GETMEDIA*******************************
COMMENT ASK USER WHERE DATA IS COMING FROM--FLOPPY, HARD, OR VDISK
COMMENT OUTPUT: $MEDIA=1,2,OR 3. %O=PATH FOR STUDENTS DATABASE FILES
COMMENT NOTE ENTIRE SYSTEM IS TOO BIG FOR ONE FLOPPY DRIVE SO IF CHOICE
COMMENT IS FLOPPY, STUDENTS.* & NEWMIX GO TO B:, EVERYTHING ELSE TO A:
COMMENT ON OUTPUT, %0="B:", "", OR "D:" 
COMMENT ************************************************************
procedure getmedia
$lv1=0
**Project File SMSS**

```plaintext
label inloop0
COMMENT PUT OUT WELCOME HEADER
call maintitle
menu print 8 30 15 4 CHOOSE DISK OPTION
menu print 10 24 15 4 1. FLOPPY DISK ONLY--a: (programs), b: (student file)
menu print 11 24 15 4 2. HARD DISK--c: programs & student file
menu print 12 24 15 4 3. FLOPPY AND VDISK--a: (programs), d: (student file)
menu print 14 22 15 4 *** DEFAULT (Esc) IS OPTION 2 ***
$firstline=10
$lastline=12
$linediff=1
$coln=23
COMMENT GET USER INPUT
call selopt
COMMENT SET DEFAULTS (IN CASE OF ESC OR 2.) TO HARD
$media=2
%0=null
COMMENT TEST FOR ESCAPE INPUT-ASSUME HARD AND EXIT
if $key=27 then jump gmend
COMMENT TEST FOR HELP REQUEST-PRINT SCREEN AND TRY AGAIN
if $key=104 or $key=72
    call help
    jump inloop0
endif
COMMENT MUST BE AN OPTION SELECTION
$media=$option
COMMENT IF HARD DISK, ALL FILES ON SMART DATAMANAGER DEFAULT CATALOG
if $option=2
    %0=null
    jump gmend
endif
COMMENT IF FLOPPY OR VDISK, NEED TO GET STUDENTS DISK IN B:
message Load STUDENTS disk into drive B:-any key when ready
COMMENT IF VDISK, GO LOAD VDISK FROM FLOPPY
if $option=3
    %0="d:"
    execute ramload in-memory
COMMENT RAMLOAD RETURNS TEXT1=N IF USER BAILS OUT
    if text1="n" then jump inloop0
else
COMMENT MUST BE FLOPPY, RETURN POINTING TO B:
    $media=1
    %0="b:"
endif
label gmend
COMMENT NEED TO SAVE MEDIA IN ENVIRONMENT FILE, ELSE IT WILL GO AWAY WHEN
COMMENT WE GO OFF TO MIXER AND RETURN (i.e variable aren't saved)
goto file environ screen environ
```

46
goto record rec-number 1
let (media)=$media
COMMENT OPEN STUDENT FILE
execute loadem in-memory
return

COMMENT ****************MAINTITL**************
COMMENT DISPLAY A STANDARD HEADER USED IN LEVELS 1 & 0
COMMENT **********************
procedure maintitl
COMMENT PAINT SCREEN RED
menu clear 15 4
menu draw box 2 35 6 44 15 4
menu print 3 36 15 4 WELCOME
menu print 4 39 15 4 TO
menu print 5 38 15 4 SMSS
return

COMMENT*****LEVPRINT*****
COMMENT DISPLAYS THE CURRENT LEVEL NUMBER AT TOP RIGHT SCREEN
COMMENT $lvl CONTAINS LEVEL
COMMENT***********************
procedure levprint
%1=$lvl
menu print 2 64 15 4 menu level %1
return

COMMENT*****FOOTER*****
COMMENT DISPLAYS THE STANDARD HELP ESCAPE LINE AT BOTTOM SCREEN
COMMENT*******************
procedure footer
if $lvl=0 or $lvl=1311
%1="next"
else
%1="last"
endif
menu print 20 5 15 4 Enter h for HELP, Esc to exit to %1 menu
return

COMMENT *******SELOPT*****
COMMENT GIVEN A MENU, RETURNS NUMBER OF OPTION SELECTED
COMMENT *********************
COMMENT INPUT *FIRSTLINE=LINE NUMBER OF FIRST OPTION
COMMENT *LASTLINE=LINE NUMBER OF LAST OPTION
COMMENT *COLM=COLUMN NUMBER TO PRINT > OF ===> POINTER
COMMENT *LINEDIFF=LINE SPACING BETWEEN OPTIONS
COMMENT OUTPUT $OPTION= NUMBER OF OPTION CHOSEN
COMMENT IF $OPTION=0, *KEY=27 FOR ESCAPE OR "H" FOR HELP
procedure selopt
$key=0
call levprint
call footer
%i=firstline
%2=colm-4
COMMENT POINTER SYMBOL =>
%3=chr(205):chr(205):chr(16)
COMMENT DISPLAY ARROW AT 1ST OPTION
menu print %1 %2 15 4 %3
COMMENT UNTIL INPUT IS ESC, ENTER, OR h or H FOR HELP
while ($key<>27 and $key<>13 and $key<>104 and $key <>72)
COMMENT SAVE CURRENT ROW TO BLANK OUT LAST POINTER POSN
$lastrow=%
COMMENT MOVE UP OR DOWN THE MENU OR BEEP ON BAD KEY
COMMENT CHECK FOR UPARROW (18432)
if $key=18432
   %1=%1-$linediff
COMMENT CHECK FOR DOWNARROW (20480)
elseif $key = 20480
   %1=%1+$linediff
else
   beep
endif
COMMENT WRAP POINTER IF NECESSARY
if %1 < firstline
   %1=lastline
endif
if %1>$lastline
   %1=firstline
endif
COMMENT BLANK OUT LAST SPOT OF POINTER
%4=$lastrow
menu print %4 %2 4 4 " "
COMMENT PRINT POINTER AT NEW POSITION
menu print %1 %2 15 4 %3
COMMENT WAIT FOR KEY TO BE STRUCK
$key=inchar
endwhile
if $key=13
   $option=((%1-firstline)/$linediff)+1
else
   $option=0
endif
COMMENT DEBUG lprint $option; $key
return
** Project File SMSS **

```
COMMENT********HELPP*******
COMMENT DISPLAYS HELP SCREEN FOR LEVEL INDICATED IN $lv1
COMMENT********************
procedure help
COMMENT CALL UP THE APPROPRIATE HELP SCREEN
%1=$lv1
load help screen help%1
paint graphics foreground 1:
COMMENT THIS UPDATE IS JUST A DUMMY OPERATION TO GET THE HELP SCREENS UNDER
COMMENT USER CONTROL. NOT REALLY UPDATING ANYTHING
update
unload screen help%1
paint graphics foreground 15
goto file %Ostudents screen student1
return
COMMENT *****PREPROCESS******
COMMENT LEVEL 11 MENU DISPLAY
COMMENT **********************
procedure preprocess
COMMENT CLEAR KEY UPDATE INDICATOR
%key,up=0
label main11
$lv1=11
COMMENT SET UP SCREEN AS USUAL
menu clear 15 4
menu draw box 2 32 5 46 15 4
menu print 3 38 15 4 FILE
menu print 4 34 15 4 MAINTENANCE
menu print 7 36 15 4 OPTIONS
menu print 9 24 15 4 1. BUILD STUDENT RECORDS FROM ANOTHER DISK
menu print 10 24 15 4 2. ADD STUDENT RECORDS MANUALLY
menu print 11 24 15 4 3. EDIT EXISTING STUDENT RECORD
menu print 12 24 15 4 4. DELETE STUDENT RECORD
menu print 13 24 15 4 5. DESIGNATE STUDENT NUMBERS
menu print 14 24 15 4 6. EDIT ENVIRONMENT RECORD (MIXING RULES)
menu print 15 24 15 4 7. DELETE 10'S (FOR MIX 3)
menu print 16 24 15 4 8. POST MIXER TENTATIVE ASSIGNMENTS
$firstline=9
$lastline=16
$linediff=1
$colm=23
label inloop11
call selopt
if $option=1
   call sturead
```

49
**Project File SMSS**

elseif $option=2
COMMENT LOAD SCREEN THAT ALLOWS ALL FIELD ENTRIES
  goto file %Ostudents screen student2
  enter
  $keyup=1
COMMENT BACK TO NORMAL SCREEN
  goto file %Ostudents screen student1
elseif $option=3
  call recfind
COMMENT TEST FOR HELP, ESCAPE OR RECORD NOT FOUND AND UPDATE IF OK
  if $option <> 0 then update
elseif $option=4
  call recfind
COMMENT TEST FOR HELP, ESCAPE, NOT FOUND, OR ALREADY DELETED AND DELETE IF OK
  if $option <> 0 and deleted=0.0 then delete
elseif $option=5
  execute stunumb in-memory
elseif $option=6
  goto file environ screen environ1
  update
  goto file %Ostudents screen student1
elseif $option=7
  execute lodel in-memory
elseif $option=8
  menu print 15 61 15 4 Mix? (1, 2 or, 3)
  label ask8
  beep
  menu input 16 61 4 15 1 text1
  if text1>null then jump main1
  if text1<>"1" and text1<>"2" and text1<>"3" then jump ask8
  %1=text1
  menu print 20 35 4 15 Posting permanent mix %1
  query predefined postmix %1 neither
elseif $key=104 or $key=72
  call help
elseif $key=27
COMMENT ESCAPE-UPDATE KEYS IF ADD HAS BEEN DONE
  if $keyup<>0
    menu print 20 59 4 15 Updating key fields
    key update
procedure sturead
label main11
$lvl=lll
COMMENT SET UP SCREEN AS USUAL
menu clear 15 4
menu draw box 2 25 4 52 15 4
menu print 3 26 15 4 SMSS - BUILD STUDENT FILE
menu print 5 15 15 4 This option allows you to build the student file by
menu print 6 15 15 4 reading student records from an ASCII floppy disk file.
menu print 7 15 15 4 See the SMSS user manual for the correct format of
menu print 8 15 15 4 this file. You can build the student file from
menu print 9 15 15 4 scratch or you can add new records to the end of the
menu print 10 15 15 4 existing file.
menu print 11 31 4 15 *****WARNING*****
menu print 12 15 15 4 If you build from scratch, any records in the current
menu print 13 15 15 4 student file will be lost.
menu print 15 36 15 4 OPTIONS
menu print 16 24 15 4 1. BUILD STUDENT FILE FROM SCRATCH
menu print 17 24 15 4 2. ADD RECORDS TO STUDENT FILE
$firstline=16
$lastline=17
$linediff=1
$colm=23
call selopt
if $option=0 and $key=27 then return
if $option=0 and ($key=104 or $key=72)
call help
jump main11
endif
execute sturead in-memory
return

COMMENT **********RECFIND************
COMMENT LEVEL 113/4 PROCEDURE- FIND A PARTICULAR RECORD
COMMENT BASED ON NAME, SSAN OR STU.NO. GIVE USER THE OPTION WHICH
COMMENT **********RECFIND************
procedure recfind
label main113
**Project File SMSS**

$lvl=113
COMMENT SET UP SCREEN OF OPTIONS
call rfscreen
$firstline=11
$lastline=13
$linediff=1
call selopt
COMMENT CHECK FOR ESCAPE OR HELP AS USUAL
if $option=0 and $key=27 then return
if $option=0 and ($key=104 or $key=72)
  call help
  jump main[l]13
endif
COMMENT INPUT A NAME, SSAN OR STU.NO ON THE OPTION LINE
COMMENT %1=SELECTED OPTION LINE NO. %2=LENGTH OF NEXT INPUT
COMMENT %3=VALUE OF INPUT, %4=KEY TO SEARCH ON, %5=FIND MODE, %6=FIND OPTION
%1=$firstline+$option-1
COMMENT ASSUME "EQUAL" AND "GLOBAL" FIND MODE AND OPTIONS
%5="equal"
%6=""
if $option=1
  %2=27
  %4="[name]"
COMMENT ON NAME SEARCH, DO (SLOWER) PART WORD AND CASE-LESS OPTIONS
  %5="partial"
  %6="iw"
  order key [name]
elseif $option=2
  %2=9
  %4="[ssan]"
  order key [ssan]
elseif $option=3
  %2=4
  %4="[stu.no]"
  order key [stu.no]
endif
COMMENT NEED TO REFRESH SCREEN SINCE ABOVE "ORDERS" DISTURBED IT
call rfscreen
menu print 10 50 15 4 ENTER SEARCH VALUE
beep
menu input %1 50 4 15 %2 text1
COMMENT GIVE ANOTHER CHANCE TO BAIL OUT
if text1 = null
  $option=0
  return
endif
menu clear 15 4
menu print 20 58 4 15 Searching for Record
%3=text

COMMENT ALL THE ABOVE WAS SET UP TO ALLOW A BINARY (FAST) SEARCH ON SSAN OR
COMMENT STUDENT NUMBER. IT SHOULD WORK WITH "find %4 %5 %3 options g%6" BUT,
COMMENT IT DOESN'T. FOR SOME REASON I ALWAYS GET CERROR OF 3002 EVEN WHEN
COMMENT BINARY SEARCH FINDS THE RIGHT RECORD. THE FOLLOWING GETS AROUND
COMMENT THE PROBLEM BY CHECKING FOR THE ACTUAL VALUE WE'RE LOOKING FOR EVEN
COMMENT IF THE 3002 RETURN COMES BACK
find %4 %5 "%3" options g%6
COMMENT DELETED RECORDS ARE LIKE THEY'RE NOT THERE
if deleted=1.0 then jump rf2
COMMENT CERROR RETURNS 3002 IF RECORD NOT FOUND
if $option=1 and cerror<>3002 then jump rf1
text2=%4
COMMENT ON BINARY SEARCH, CHECK FOR A RETURNED VALUE THAT WE'RE LOOKING FOR
if $option<>1 and text2=text1 then jump rf1
COMMENT ELSE RECORD NOT FOUND
label rf2
  call rfscreen
COMMENT ECHO MESSAGE AND ORIGINAL VALUE
  menu print 1 6 4 15 ***NOT FOUND***
  menu print 1 5 4 15 %3
  beep 3
COMMENT SET $OPTION TO SHOW NO UPDATE ON RETURN
  $option=0
label rf1
return

COMMENT ************RFSCREEN************
COMMENT DISPLAY THE OPTION SCREEN FOR REC.find - LEVEL 113 MENU
COMMENT ************RFSCREEN************
procedure rfscreen
menu clear 15 4
menu print 3 35 15 4 STUDENT
menu print 4 36 15 4 RECORD
menu print 5 34 15 4 SELECTION
menu draw box 2 33 4 44 15 4
menu print 7 10 15 4 Choose which record to process by one of the following
fields
menu print 8 10 15 4 NOTE: SSAN & STUDENT NUMBER options are much faster
menu print 10 36 15 4 OPTIONS
menu print 11 24 15 4 1. NAME (or partial name)
menu print 12 24 15 4 2. SSAN
menu print 13 24 15 4 3. STUDENT NUMBER
return

COMMENT **************MIXEM**************
COMMENT LEVEL 12 PROCEDURE- CALL MIXING PROGRAM
COMMENT **************MIXEM**************
procedure mixem
label main12
lvl=12
COMMENT SET UP SCREEN AS USUAL
menu clear 15 4
menu draw box 2 29 4 48 15 4
menu print 3 30 15 4 SMSS MIXING OPTION
menu print 5 11 15 4 If there have been no changes to the student data base
menu print 6 11 15 4 since the last mixing run, you can skip the ASCII file
menu print 7 11 15 4 update. If there have been changes, or you don't know,
menu print 8 11 15 4 be safe by running the ASCII file update.
menu print 10 36 15 4 OPTIONS
menu print 12 24 15 4 1. SKIP ASCII FILE UPDATE
menu print 13 24 15 4 2. RUN ASCII FILE UPDATE
firstline=12
lastline=13
linediff=1
col=23
call selopt
if option = 0 and key=27 then return
if option =0 and ($key=104 or $key=72)
call help
jump main12
endif
COMMENT WRITE ASCII ENVIRONMENT RECORD
execute envwrite in-memory
COMMENT IF SELECTED, UPDATE ASCII STUDENT FILE
if $option=2 then execute stuwrite in-memory
COMMENT GO OFF TO MIXER UNDER DOS AND DON'T RETURN HERE
transfer gomix

COMMENT ****************** REPORTS ******************
COMMENT LEVEL 13 PROCEDURE - GENERATE SELECTED REPORTS
COMMENT ****************** REPORTS ******************
procedure reports
label main13
lvl=13
COMMENT ASK WHICH MIX WE'RE REPORTING ON
menu clear 15 4
menu draw box 2 28 4 48 15 4
menu print 3 30 15 4 SMSS REPORTS OPTION
menu print 6 24 15 4 Choose which mix to report on.
menu print 8 36 15 4 OPTIONS
menu print 10 24 15 4 1. MIX1
menu print 11 24 15 4 2. MIX2
menu print 12 24 15 4 3. MIX3
menu print 13 24 15 4 4. MIXX (Tentative mix results of last mixer run)
$firstline=10
$lastline=13
$linediff=1
$colm=23
call selopt
if $option=0 and $key=27 then return
if $option=0 and ($key=104 or $key=72)
call help
jump main13
endif
COMMENT SAVE SELECTED MIX NUMBER
mix=$option
if $mix=1
%9="1"
elif $mix=2
%9="2"
elif $mix=3
%9="3"
else
%9="x"
endif
COMMENT NOW LET USER SELECT REPORT
label main131
$lvl=131
menu clear 15 4
menu draw box 2 20 4 48 15 4
menu print 3 29 15 4 SMSS REPORTS OPTION
menu print 5 36 15 4 OPTIONS
menu print 7 24 15 4 1. SCHOOL MIXING SUMMARY
menu print 8 24 15 4 2. SQUADRON MIXING SUMMARY
menu print 9 24 15 4 3. SEMINAR MIXING REPORT
menu print 10 24 15 4 4. COMPLETE MIXING REPORT (all the above)
menu print 11 24 15 4 5. SCHOOL ALPHA ROSTER
menu print 12 24 15 4 6. SQUADRON ALPHA ROSTER
menu print 13 24 15 4 7. SEMINAR ALPHA ROSTER
$firstline=7
$lastline=13
$linediff=1
$colm=23
call selopt
if $option=0 and $key=27 then jump main13
if $option=0 and ($key=104 or $key=72)
call help
jump main131
endif
COMMENT PRODUCE REPORT
call doreport
jump main131

55
**** Project File SMSS ****

return

COMMENT ******************************************DOREPORT******************************************
COMMENT PRODUCE REQUESTED REPORTS
COMMENT ON INPUT, *OPTION INDICATES REQUESTED REPORT
COMMENT **MIX** INDICATES REQUESTED .MX
COMMENT ******************************************

procedure doreport
COMMENT DETERMINE WHICH REPORT IS REQUESTED

if *option=1
COMMENT SCHOOL SUMMARY
   call skoolsum
elseif *option=2
COMMENT SQUADRON SUMMARY-ASK WHICH SQUADRON
   menu print 7 52 15 4 Squadron? (1,2,3, or 4)
   menu print 8 52 15 4 382
   label ask2
   beep
   menu input 8 55 4 15 1 text1
   if text1=null then return
   if text1<>"1" and text1<>"2" and text1<>"3" and text1<>"4" then jump ask2
   call squadsum
elseif *option=3
COMMENT SEMINAR STAT REPORT
COMMENT SEMINAR ALPH-ASK WHICH SEMINAR(S)
   menu print 8 53 15 4 Seminar(s)?
   menu print 14 27 15 4 Enter: "ALL" for all seminars
   menu print 15 34 15 4 "SQ1","SQ2","SQ3", or "SQ4" for all in a sqdn
   menu print 16 34 15 4 Seminar number for single seminar
   label ask3
   beep
   menu input 9 53 4 15 3 text1
COMMENT PICK APPROPRIATE SEMINARS
   if text1=="all"
      jump sortem
   elseif left(text1,2)=="sq"
COMMENT SELECTING ONE WHOLE SQUADRON
   text1=mid(text1,3,1)
   menu print 20 56 4 15 Selecting Seminars
   query predefined squadron index temp2
   elseif text1=null
      return
   else
COMMENT SINGLE SEMINAR-
COMMENT MAKE SURE IT'S IN RANGE

56
if val(text1)<1 or val(text1) > 50 then jump ask3
COMMENT IF HE ENTERED SINGLE DIGIT, FIX IT
if len(text1)=1 then text1="0"+text1
menu print 20 58 4 15 Selecting Seminar
query predefined mix%9 index temp2
endif
order index temp2
label sortem
call semsum

elseif $option=4
COMMENT COMPLETE MIXING SUMMARY-ALL SEMINARS, SCHOOL, ALL SQUADRONS
call semsum
COMMENT CLOSE INDEX FILE TEMP
order sequential
call skoolsum
$sqcnt=1
while $sqcnt <= 4
  text1=fixed($sqcnt,0)
call squadsum
order sequential
$sqcnt=$sqcnt+1
endwhile

elseif $option=5
COMMENT SCHOOL ALPHA REPORT
order key [name]
COMMENT DETERMINE OUTPUT MEDIUM
call whichout
report print alpha %4 %5

elseif $option=6
COMMENT SQUADRON ALPHA-ASK WHICH SQUADRON
menu print 11 49 15 4 Squadron?
menu print 12 49 15 4 382
label ask5
beep
menu input 12 52 4 15 1 text1
if text1=null then return
%1=text1
if text1<>"1" and text1<>"2" and text1<>"3" and text1<>"4" then jump ask5
order key [name]
menu print 20 58 4 15 Selecting Squadron %1
COMMENT ISOLATE THAT SQUADRON
query predefined squadron index temp
order index temp
COMMENT DETERMINE WHICH OUTPUT MEDIUM
call whichout
**** Project File SMSS ****

report print alpha %4 %5

elseif $option=7
COMMENT SEMINAR ALPH-ASK WHICH SEMINAR(S)
  menu print 12 4 9 15 4 Seminar(s)?
  menu print 14 27 15 4 Enter: "ALL" for all seminars
  menu print 15 34 15 4 "SQ1", "SQ2", "SQ3", or "SQ4" for all in a sqdn
  menu print 16 34 15 4 Seminar number for single seminar
  label ask6
  beep
  menu input 13 52 4 15 3 text1
  if text1=null then return
  call getsem
  COMMENT TEST FOR ERROR RETURN
  if text1="O" then jump ask6
  order index temp
  COMMENT DETERMINE WHICH OUTPUT MEDIUM
  call whichout
  report print mix%9 %4 %5
endif

COMMENT MAKE SURE WE'RE LOOKING AT WHOLE FILE ON RETURN
order sequential
return

COMMENT ******************SKOOLSUM**********************
COMMENT PRODUCE SCHOOL MIXING STAT SUMMARY
COMMENT ******************SKOOLSUM**********************
procedure skoolsum
COMMENT SORT BY SQUADRON
  menu print 20 61 4 15 Sorting by squadron
  sort predefined squadron index temp
  order index temp
  menu print 20 47 4 15 Computing school stats
COMMENT ZERO ALL THE COUNTERS-4 SETS OF 23
  $firstunit=1
  $lastunit=4
  call czero
  goto record rec-number 1
COMMENT RUN THRU FILE-UPDATE COUNTS OF ALL ATTRIBUTES. THERE WILL BE 4
COMMENT DIFFERENT SETS OF COUNTERS FOR 4 SQUADRONS
label nextrec1
  %1=[squadron]
  call count
  if record < records
    goto record next
  jump nextrec1
endif
**** Project File SMSS ****

COMMENT PRINT OUT THE REPORT WITH ALL ATTRIBUTE COUNTERS
menu print 20 33 4 15 Printing School Stats- Make sure printer is on
lprint
lprint repeat(" ",20);"SCHOOL MIX STATISTICS"
lprint repeat(" ",38);"MIX ":"%9";repeat(" ",25);date1(today)
lprint
lprint " ATTRIBUTE";repeat(" ",18);"SQ1";" SQ2";" SQ3";" SQ4";" TOTAL"
lprint
value1=1
COMMENT THERE ARE 23 DIFFERENT ATTRIBUTES
COMMENT AND THE FIELD DESCRIPTIONS ARE IN THE ENVIRONMENT FILE
goto file environ screen environ1
while value1< 23
 %1=value1
COMMENT PRINT ATTRIBUTE COUNT, LABEL ,COUNT FOR EACH SQUADRON, & TOTAL COUNT
lprint repeat(" ",2-len("%1"));"%1. ";[d%1];repeat(" ",27-len([d%1]));%c%1_1;
lprint repeat(" ",4-len(fixed(%c%1_1,0)));%c%1_2;
lprint repeat(" ",4-len(fixed(%c%1_2,0)));%c%1_3;
lprint repeat(" ",4-len(fixed(%c%1_3,0)));%c%1_4;
lprint repeat(" ",4-len(fixed(%c%1_4,0)));%c%1_1+%c%1_2+%c%1_3+%c%1_4
value1=value1+1
endwhile
COMMENT FORM FEED
lprint chr(12)
COMMENT CLEAR AWAY THE COUNTERS
call clear
COMMENT BACK TO STUDENT FILE
goto file %Ostudents screen student1
return

COMMENT ****************SQUADSUM*******************************
COMMENT PRODUCE SQUADRON SUMMARY REPORT
COMMENT ON INPUT TEXT1= SQUADRON NUMBER
COMMENT **********************SQUADSUM*******************************
procedure squadsum
%3=text1
menu print 20 58 4 15 Selecting Squadron %3
COMMENT ISOLATE THAT SQUADRON
query predefined squadron index temp2
order index temp2
COMMENT SORT BY SEMINAR
menu print 20 61 4 15 Sorting by Seminar
sort predefined semalf%9 index temp
COMMENT ZERO ALL THE COUNTERS-NEED TO KNOW WHICH SEMINARS ARE IN THIS
COMMENT SQUADRON. ENVIRONMENT FILE HAS THAT
goto file environ screen environ1
$frstunlt=[fstsem%3]
$lastunlt=[lstsem%3]
**** Project File SMSS ****

menu print 20 61 4 15 Preparing Counters
call czero
goto file %0students screen student
order index temp
menu print 20 45 4 15 Computing squadron stats
goto record rec-number 1
COMMENT RUN THRU FILE-UPDATE COUNTS OF ALL ATTRIBUTES. THERE WILL BE 4
COMMENT DIFFERENT SETS OF COUNTERS FOR 4 SQUADRONS
label nextrec2
%1=val([mix%9])
call count
if record < records
  goto record next
  jump nextrec2
endif
COMMENT PRINT OUT THE REPORT WITH ALL ATTRIBUTE COUNTERS
menu print 20 28 4 15 Printing Squadron %3 Stats-Make sure printer is on
lprint
lprint repeat(" ",27);"SQUADRON ";"%3";" MIX STATISTICS"
lprint repeat(" ",38);"MIX ";"%9";repeat(" ",25);date1(today)
lprint
lprint " ATTRIBUTE";repeat(" ",10); COMMENT PRINT ALL THE SEMINAR NUMBERS ACROSS TOP LINE
value1=$frstunit
while value1<=$lastunit
  lprint repeat("O",2-len(fixed(value1,0))));value1=" ";
  value1=value1+1
endwhile
lprint " TOTAL"
lprint
value1=1
COMMENT THERE ARE 23 DIFFERENT ATTRIBUTES
COMMENT AND THE FIELD DESCRIPTIONS ARE IN THE ENVIRONMENT FILE
goto file environ screen environ1
menu print 20 52 4 15 Printing Squadron %3 Stats
while value1<=23
  %1=value1
  COMMENT PRINT ATTRIBUTE NUMBER, LABEL, COUNT FOR EACH SEMINAR, & TOTAL COUNT
  lprint repeat(" ",2-len("%1"));"%1. ";[d%1];repeat(" ",27-len([d%1]));
  value2=$frstunit
  $tot=0
  while value2<=$lastunit
    %2=value2
    lprint %c%1_%2;repeat(" ",3-len(fixed($c%1_%2,0)));
    value2=value2+1
    $tot=$tot+$c%1_%2
  endwhile
  lprint $tot
**** Project File SMSS ****

```plaintext
value1=value1+1
endwhile

COMMENT FORM FEED
lprint chr(12)

COMMENT CLEAR AWAY THE COUNTERS
call clear

COMMENT BACK TO STUDENT FILE
goto file %Ostudents screen student1
return

COMMENT **********************************SEMSUM**********************************
COMMENT PRODUCE SEMINAR SUMMARY REPORT
COMMENT **********************************SEMSUM**********************************
procedure semsum
  menu print 20 58 4 15 Sorting by seminar
  COMMENT SORT BY SEMINAR, RANK (KIND OF) - 1ST IS MOST POTENTIAL 9L, LAST IS LEAST
  sort predefined slorder%9 index temp
  order index temp
  COMMENT DETERMINE WHICH OUTPUT MEDIA
  call whichcut
  report print semstat%9 %4 %5
return

COMMENT ************************COUNT*******************************
COMMENT UPDATE ALL THE ATTRIBUTE COUNTS BASED ON THIS RECORD
COMMENT ************************COUNT*******************************
procedure count
  COMMENT IGNORE DELETED RECORDS
  if deleted then return
  COMMENT EACH "$C" COUNTER IS A COUNT OF THAT ATTRIBUTE
  COMMENT %1 INDICATES WHICH UNIT (SQUADRON OR SEMINAR)
  %c1_%1=%c1_%1+if [usaf]="y" then 1 else 0
  %c2_%1=%c2_%1+if [nomast]="y" then 1 else 0
  %c3_%1=%c3_%1+if [sorg]="y" then 1 else 0
  %c4_%1=%c4_%1+if [pilot]="y" then 1 else 0
  %c5_%1=%c5_%1+if [navigator]="y" then 1 else 0
  %c6_%1=%c6_%1+if [single]="y" then 1 else 0
  %c7_%1=%c7_%1+if [usafl]="y" then 1 else 0
  %c8_%1=%c8_%1+if [army]="y" then 1 else 0
  %c9_%1=%c9_%1+if [othercompl]="y" then 1 else 0
  %c10_%1=%c10_%1+if [minority]="y" then 1 else 0
  %c11_%1=%c11_%1+if [female]="y" then 1 else 0
  %c12_%1=%c12_%1+if [captain]="y" then 1 else 0
  %c13_%1=%c13_%1+if [nonline]="y" then 1 else 0
  %c14_%1=%c14_%1+if [stratopsk]="y" then 1 else 0
  %c15_%1=%c15_%1+if [tacopsk]="y" then 1 else 0
```

61
**** Project File SMSS ****

```plaintext
$c16_11=%c16_11+if [ppbsk]="y" then 1 else 0
$c17_11=%c17_11+if [acqlogsk]="y" then 1 else 0
$c18_11=%c18_11+if [spacesk]="y" then 1 else 0
$c19_11=%c19_11+if [tbd1]="y" then 1 else 0
$c20_11=%c20_11+if [tbd2]="y" then 1 else 0
$c21_11=%c21_11+if [tbd3]="y" then 1 else 0
$c22_11=%c22_11+if [tbd4]="y" then 1 else 0
$c23_11=%c23_11+if [tbd5]="y" then 1 else 0
return

COMMENT ******************CZERO**********************
COMMENT ZERO ALL THE COUNTERS
COMMENT COUNTERS ARE %CIj WHERE i RANGES FROM 1 TO 23 AND
COMMENT j CAN BE 1 TO 4 FOR SQUADRONS IN THE SCHOOL SUMMARY
COMMENT OR ANY SEMINAR NUMBER IN THE SQUADRON SUMMARY
COMMENT ON INPUT, $FRSTUNIT & $LASTUNIT ARE 1 AND 4 FOR SCHOOL SUMMARY
COMMENT OR THE HIGHEST & LOWEST SEMINAR NUMBER FOR SQUADRON SUMMARY
COMMENT ******************CZERO**********************

procedure czero
value1=$frstunit
while value1 <= $lastunit
%1=value1
value2=1
COMMENT 23 COUNTERS FOR EACH UNIT
  while value2<=23
    %2=value2
    *c%2_%1=0
    value2=value2+1
  endwhile
value1=value1+1
endwhile
return

COMMENT ******************CLEAR***********************
COMMENT CLEAR ALL THE COUNTERS FROM MEMORY
COMMENT ******************CLEAR***********************

procedure clear
value1=$frstunit
while value1 <= $lastunit
%1=value1
value2=1
COMMENT 23 COUNTERS FOR EACH UNIT
  while value2<=23
    %2=value2
    clear *c%2_%1
    value2=value2+1
  endwhile
```

62
**Project File SMSS**

value1=value1+1
endwhile
return

COMMENT ****************GETSEM************
COMMENT RUN APPROPRIATE QUERY BASED ON TEXT1 INPUT
COMMENT RETURN WITH TEMP INDEX FILE SET
COMMENT **********************
procedure getsem
if text1="all"
COMMENT SORT WHOLE FILE ON SEMINAR ORDER. REMEMBER %9 IS BASED ON $MIX
else if left(text1,2)="sq"
COMMENT SELECTING ONE WHOLE SQUADRON
   text1=mid(text1,3,1)
   menu print 20 59 4 15 Selecting Seminar(s)
   query predefined squadron index temp2
   order index temp2
COMMENT SORT BY SEMINAR/ALPHA
else
COMMENT SINGLE SEMINAR
COMMENT MAKE SURE IT'S IN RANGE
   if val(text1)<1 or val(text1) > 50
      text1="O"
   return
endif
COMMENT IF HE ENTERED SINGLE DIGIT, FIX IT
   if len(text1)=1 then text1="O":text1
   menu print 20 56 4 15 Selecting Seminar
   query predefined mix%9 index temp2
   order index temp2
endif
   menu print 20 61 4 15 Sorting by seminar
   sort predefined semalf%9 index temp
return

COMMENT *******************WHICHOUT**********************
COMMENT USER SELECTS PRINT OUTPUT OPTION: PRINTER, DISK, OR SCREEN
COMMENT ON RETURN, %4= "PRINTER", "DISK", OR "SCREEN"
COMMENT %5= NULL, FILENAME OF DISK FILE, OR NULL
COMMENT *******************

procedure whichout
label main1311
$lv1=1311
COMMENT DISPLAY SELECTION SCREEN
menu clear 15 4
menu draw box 2 31 5 45 15 4
menu print 3 32 15 4 REPORT OUTPUT
**** Project File SMSS ****

menu print 4 34 15 4 SELECTION
menu print 6 19 15 4 Choose where you want report output sent
menu print 8 36 15 4 OPTIONS
menu print 10 24 15 4 1. PRINTER
menu print 11 24 15 4 2. DISK
menu print 12 24 15 4 3. SCREEN
menu print 14 24 15 4 Default (Esc) is PRINTER

#firstline=10
$lastline=12
$linediff=1
$colm=23
call selopt
if $option=0 and $key=27 then $option=1
if $option=0 and ($key=104 or $key=72)
call help
jump main1311
endif
%6=null
COMMENT CALCULATE %4 AND %5
if $option=1
%4="printer"
%5=null
elseif $option=2
%4="disk"
COMMENT GET DISKFILE NAME
    menu print 10 37 15 4 Enter file (or path) name--No extension
    beep
    menu input 11 37 4 15 20 text1
    if text1=null then jump main1311
    %5=text1
    %6="file, " ;"%5"; ".prt"
else
%4="screen"
%5=null
endif
menu print 20 30 4 15 Sending report to %4 %6
return
** Project File ENWRITE ****

COMMENT WRITE THE ASCII ENVIRONMENT RECORD
COMMENT FIRST ERASE ENV.ASC IF IT ALREADY EXISTS
goto file environ screen environ
if file("env.asc")=1.0 then file erase env.asc
COMMENT REBUILD ONE-RECORD ASCII ENVIRONMENT FILE
COMMENT THIS ORDER SEEMS A LITTLE COMPLICATED BUT IT'S THE ORDER THAT MIXER
COMMENT NEEDS THEM IN. IF THE STRUCTURE OF ENVIRON EVER CHANGES, THESE FIELD
COMMENT NUMBERS WILL PROBABLY CHANGE, BUT THE RELATIVE ORDER MUST STAY THE
COMMENT SAME. THAT ORDER IS MIX, MEDIA, NUM.BUDS, FSTSEM1, LSTSEM1, MISSEM11,
COMMENT MISSEM12, MISSEM13, (SEMINAR DATA FOR SQDN 2, 3, & 4), SEMCHANGE, BUDS, SPACESK, PPBSK, TACPSK
COMMENT STRATOPS, PILOT, ANV, SINGLE USAFA, ARMY, OTHRCOMP, MINORITY, FEMALE,
COMMENT CAPTAIN, NONLINE, SORG, SORRULE, NOMAST, USAF, IORULE, TBD1,
COMMENT TBD2, TBD3, TBD4, TBD5. NOTE ARIRULE NOT USED.
write all [1;74;2;22;27;26;45;43;42;41;44;31;40;30;24;29;28;25;46;50] ascii
equiv.env.asc
**** Project File GOMIX ****

COMMENT PROJECT FILE GOMIX
COMMENT WRAPUP SMART SESSION AND EXIT OFF TO DOS TO DO MIXING
menu clear 15 4
menu print 20 51 4 15 Exit Smart. Go off to mixer
quit
**** Project File IODEL ****

COMMENT PROJECT IODEL- DELETE ALL THE INTERNATIONAL OFFICERS
COMMENT THEY'RE GONE AFTER 2ND MIX. THIS PROCESS MARKS THEM DELETED BUT DOES
COMMENT NOT PURGE THEIR RECORDS FROM THE STUDENT FILE

menu clear 15 4
menu print 3 24 15 4 SMSS - DELETE IO'S
menu draw box 2 23 4 42 15 4
menu print 5 2 15 4 This process should be done prior to mixing for 3rd mix
menu print 6 2 15 4 since the IO's will be gone then.
menu print 8 2 15 4 ARE YOU SURE YOU WANT TO DELETE IO'S(y/n)?
label iodel1
beep 1
menu input 8 45 4 15 1 $answer
if $answer=="n" then end
if $answer<>"y" and $answer<>"Y" then jump iodel1
COMMENT USE PREDEFINED QUERY TO DELETE IOS
menu print 20 40 4 15 Deleting International Officer Records
query predefined iodel neither
end
**** Project File LOADEM ****

COMMENT ******************LOADEM**************************
COMMENT INITIAL LOAD OF STUDENT DATA BASE & HELP SCREEN FILE
COMMENT %O MUST BE SET OUT OF GETMEDIA OR NEWMIX
COMMENT ****************************************************
menu print 20 58 4 15 Loading Student file
load %O students screen student1
if cerror=3001
  beep 3
  message Error opening students database--give up
COMMENT TELL SMSS.BAT NOT TO DO MIXING
  unload all
  if file("env.asc")=1.0 then file erase env.asc
  quit
endif
activate %O students screen student2
menu print 20 54 4 15 Loading Help Screen file
activate help screen standard
**Project File NEWMIX**

COMMENT ********************************************
COMMENT PROJECT NEWMIX-UPDATE STUDENT DB WITH RESULTS OF MIXING
COMMENT "media" and "%0" set on return IAW environment file
COMMENT ********************************************

COMMENT FIRST GET MEDIA CODE OUT OF ENVIRONMENT FILE TO DETERMINE HOW WE'RE
COMMENT OPERATING. THIS WAS SET BY USER AT INITIAL ENTRY
goto file environ screen environ1
goto record rec-number 1
let "media"=[media]
COMMENT FLOPPY?
if "media"=1
%0=":b:"
COMMENT HARD DISK?
elself "media"=2
%0=null
COMMENT FLOPPY/VDISK?
elself "media"=3
%0=":d:"
COMMENT ANYTHING ELSE MEANS A PROBLEM
else
  message "ERROR in environment file media code"
  stop
endif
COMMENT READ STUOUT.ASC (MIXER OUTPUT) INTO NEWMIX DATA BASE
COMMENT NEWMIX WILL BE THE TRANSACTION FILE TO UPDATE FIELDS IN STUDENTS
load "newmix" screen standard
menu print 20 37 4 15 Reading mixer interface file (stuout.asc)
COMMENT MIXERPasses NAME, STU.NO, SQUADRON, MIXX, RULE PRIORITY, SLX, ASLX, CC, SRO
COMMENT WE'LL IGNORE NAME, CC, AND SRO
read ascii stuout.asc fields [0; stu.no; squadron; mixx; priority; slx; aslx]
COMMENT PREPARE FOR TRANSACTION UPDATE. LOAD STUDENT FILE
execute loadem in-memory
order key [stu.no]
split horizontal 11 2
goto window 2
goto file "newmix" screen standard
menu print 19 33 4 15 ***UPDATING STUDENT FILE WITH NEW MIX DATA***
COMMENT DETERMINE TIME TO RUN
if "media"=1
%1="33"
elself "media"=2
%1="9"
elself "media"=3
%1="15"
endif
menu print 20 33 15 4 This will take about %1 minutes. Please wait.
transactions predefined newmix no-audit
**** Project File NEWMIX ****

COMMENT PREDEFINED TRANSACTION SET UP TO DELETE RECORDS-NOW PURGE THEM

close
unload file %Onewmix
menu print 20 46 4 15 Purging used transaction records
utilities purge %Onewmix
COMMENT DELETE FILE STUOUT.ASC TO SHOW UPDATE HAS BEEN DONE
repaint
file erase stuout.asc
**** Project File RAMLOAD ****

COMMENT ******************RAMLOAD*************************
COMMENT LOAD STUDENT (NEWMIX) DATA BASES INTO VIRTUAL MEMORY STORAGE
COMMENT VIRTUAL MEMORY OPERATES MUCH FASTER THAN FLOPPY DISK
COMMENT SYSTEM MUST HAVE BEEN BOOTTED WITH THE FOLLOWING COMMAND IN THE ROOT
COMMENT CONFIG.SYS FILE--DEVICE=\BIN\VDISK.SYS 220
COMMENT ****************** RAMLOAD *************************

menu print 16 2 4 15 NOTE:
menu print 16 7 15 4 This option requires that VDISK be setup in the root
config.sys file
menu print 17 7 15 4 at startup time. If not, go back to DOS & do it.
menu print 18 7 15 4 Is VDISK configured? (y/n)
lable rloop
beep
menu input 18 34 4 15 1 text1
COMMENT IF USER BAILS OUT RETURN WITH TEXT1=N
if text1="n" then end
if text1<>"y" and text1<>"Y" then jump rloop
COMMENT COPY ALL STUDENTS & NEWMIX FILES TO VDISK
menu print 20 47 4 15 Loading Student file to ramdisk
file copy b:\students.db to d:students.db
file copy b:\students.dbs to d:students.dbs
COMMENT INDEX FILES-IF STRUCTURE OF STUDENTS CHANGES, THESE MAY HAVE TO CHANGE
file copy b:\students.101 to d:students.101
file copy b:\students.106 to d:students.106
file copy b:\students.121 to d:students.121
file copy b:\newmix.db to d:newmix.db
file copy b:\newmix.dbs to d:newmix.dbs
** Project File RAMSAVE 

COMMENT ******************RAMSAVE************************
COMMENT SAVE VDISK FILES BACK TO FLOPPY AFTER SMSS SESSION ENDS
COMMENT ******************RAMSAVE************************

menu print 20 46 4 15 Reclaiming vdisk files to floppy
COMMENT FILE COPY DOESN'T OVERWRITE-SO FIRST DELETE DESTINATION FILES
file erase b:students.*
file erase b:newmix.db*
COMMENT VOLATILE VDISK BACK TO PERMANENT FLOPPY
file copy d:students.db to b:students.db
file copy d:students.dbs to b:students.dbs
COMMENT INDEX FILES-NOTE IF FILE STRUCTURE CHANGES- THESE MAY HAVE TO CHANGE
file copy d:students.101 to b:students.101
file copy d:students.106 to b:students.106
file copy d:students.121 to b:students.121
file copy d:newmix.db to b:newmix.db
file copy d:newmix.dbs to b:newmix.dbs
COMMENT ERASE RAMDISK FILES SO DON'T GET "FILE EXISTS" MESSAGE NEXT TIME
file erase d:.*
repaint
PROJECT STUNUMB - CREATE STUDENT NUMBERS

COMMENT THIS SHOULD BE A ONE TIME OPERATION DONE WHEN STUDENT FILE
COMMENT IS SETTLED

COMMENT SET UP SCREEN- ISSUE WARNING MESSAGE

menu clear 15 4

menu print 3 24 15 4 SMSS - CREATE STUDENT NUMBERS

menu draw box 2 23 4 53 15 4

menu print 5 2 15 4 This process should generally be done once at the start of

menu print 6 2 15 4 the school year after all the records have been added to

menu print 7 2 15 4 the student file. Once you've assigned student numbers,

menu print 8 2 15 4 you shouldn't do this anymore.

menu print 10 2 15 4 ARE YOU SURE YOU WANT TO ASSIGN NEW STUDENT NUMBERS (y/n)?

label stunum

beep 1

menu input 10 61 4 15 1 $answer

if $answer="n" then end

if $answer(>"y" and $answer<"Y" then jump stunum

COMMENT PUT FILE IN ALPHA ORDER

order key [name]

COMMENT USE PREDEFINED QUERY TO UPDATE EACH [STU.NO] TO 10,15,20,...ETC

menu print 20 31 4 15 Assigning student number to each student record

query predefined stunum neither

COMMENT REORGANIZE KEY FOR NEW VALUES

menu print 20 44 4 15 Building student number key file

key update

end
**** Project File STUREAD ****

COMMENT PROJECT STUREAD-READ IN THE STUDENT DATABASE FROM ASCII DISK
COMMENT ON INPUT, $option IS SET TO 1 OR 2 FROM LEVEL 111 OPTION SCREEN
Call header
COMMENT READ IN ASCII FILE-FIRST GET FILENAME
Menu print 7 2 15 4 ENTER NAME OF INPUT FILE (e.g. STUDENT.ASC)
Menu print 8 2 15 4 (Escr to exit).
Beep
Menu input 7 46 4 15 12 text1
COMMENT LET HIM BAIL OUT HERE TOO
If text1=null then end
COMMENT OPTION 1 MEANS BUILD FROM SCRATCH-DELETE ALL RECORDS
If $option = 1 then call deletem
Call header
%1=text1
Message Load source diskette in drive A:-Any key when ready
Menu print 20 41 4 15 Reading source file into Student file
Commend Field list may have to change depending on AU/DP
Comment Next IS INTERIM FORMAT FROM BUCKNER'S DATABASE
Read ASCII A:\%1 Fields [1;6;2;9;8;12;19;26;3;5;7;11;18;10;20;22;27;23;25;13;17;28;31;33]
Comment Update ALL THE KEYS
Menu print 20 36 4 15 Organizing all key indexes
Key update
Comment IF FLOPPY, MAKE SURE WE GET RUNTIME DISK BACK IN A:
If $media=1 or $media=3 then message Load SMSS execution disk back in A:-Any
Key when ready
End

Comment ***************HEADER*************
Comment Paints the screen properly
Comment **********************
Procedure header
Menu clear 15 4
Menu print 3 26 15 4 SMSS - BUILD STUDENT FILE
Menu draw box 2 25 4 51 15 4
Return

Comment *************DELETEm*************
Comment Clear out existing student database by deleting all records
Comment **********************
Procedure deletem
Menu print 5 2 15 4 YOU ARE ABOUT TO ERASE ALL CURRENT DATABASE RECORDS
Menu print 6 2 15 4 ARE YOU SURE YOU WANT TO DO THIS (y/n)?
Label dell
Beep 3
Menu input 6 41 4 15 1 $answer
If $answer == "n" then end
If $answer <>"y" and $answer <>"Y" then jump dell


COMMENT THAT'S ENOUGH WARNING-HE REALLY WANTS TO DO IT
COMMENT PREDEFINED QUERY "DELALL" DELETES ALL ACTIVE RECORDS-USE IT
menu print 20 33 4 15 Deleting all old records in Student file
query predefined delall neither
unload file %Ostudents
menu print 20 33 4 15 Purging all deleted records from Student file
utilities purge %Ostudents
load %Ostudents screen studenti
return
**** Project File STUWRITE ****

COMMENT *********************
COMMENT PROJECT STUWRITE
COMMENT WRITE THE ASCII STUDENT FILE TO PASS TO THE MIXER
COMMENT *********************
goto file %0students screen student1
COMMENT ORDER BY RANK, DOR, TAFCSD
menu print 20 64 4 15 Sorting by rank
sort predefined dor index temp
order index temp
COMMENT GET RID OF LAST VERSION OF ASCII FILE IF IT EXISTS
if file ("stu.asc")=1.0 then file erase stu.asc
COMMENT BUILD THE ASCII FILE FOR MIXING
menu print 20 40 4 15 Writing mixer interface file (stu.asc)
write all [1;28;31;33;39;42;65] ascii stu.asc
The mixer is a program, written in GW-BASIC, that does student mixing according to the rules passed in the environment record.
PROGRAM NAME: STUDENT MIX SOFTWARE SYSTEM (SMSS)

FUNCTION: MIX ACSC STUDENTS

FILE NAME: MIXER.BAS DATE: 16 FEB 1988

COMPUTER: ZENITH 158 LANGUAGE: BASIC

AUTHOR: ART DECELLES

DESCRIPTION:

THIS PROGRAM OPTIMALLY MIXES ACSC STUDENTS BASED ON STUDENT CHARACTERISTICS AND RULE PRIORITIES [NINE (HIGHEST) TO ZERO (LOWEST)]. MIXING OCCURS IN THE FOLLOWING GENERAL SEQUENCE:

1. SEMINAR LEADERS AND THEIR ASSISTANTS (SL/ASL) ARE ASSIGNED IN THE FIRST PASS OF THE STUDENT DATA. ALSO, PRE-ASSIGNED STUDENTS ARE "LOCKED-IN". INTERNATIONAL OFFICERS AND SOS STUDENTS DON'T CHANGE SEMINARS IF THEY ARE ASSIGNED A PRIORITY OF "9".

2. ALL OTHER STUDENTS ARE RANDOMLY ASSIGNED TO SEMINARS BASED ON THEIR CHARACTERISTICS, EACH CONSIDERED IN ORDER OF RULE PRIORITY ESTABLISHED BY THE MIX MASTER.

A. STUDENTS CANNOT BE ASSIGNED TO A PREVIOUS SEMINAR.

B. ASSIGNMENT IS MADE IF ALLOCATIONS PERMIT.

C. IF CONFLICT OCCURS MIXER ATTEMPTS TO MAKE A ONE FOR ONE EXCHANGE WITHIN ALLOCATION CONSTRAINTS.

D. IF STILL NOT POSSIBLE A FORCED ASSIGNMENT IS MADE:

a. IF AN ALLOCATED SLOT IS AVAILABLE FOR THE RULE BEING PROCESSED.

b. IF NOT, A ONE FOR ONE EXCHANGE IS ATTEMPTED WITH ASSIGNMENTS OF EQUAL PRIORITY.

C. AS A LAST RESORT, ONE FOR ONE EXCHANGES ARE ATTEMPTED FOR INCREMENTALLY HIGHER PRIORITIES.

MAJOR VARIABLES:

ALOT(27,4,11): ARRAY CONTAINING ALLOCATIONS WHICH REPRESENT THE TARGET BALANCED CHARACTERISTICS.

STAT(27,4,11): ARRAY TO GATHER STATISTICS ON ASSIGNMENTS.
ASSIGN(44,13,5): ARRAY CONTAINING A SUBSET OF STUDENT INFO AFTER AN ASSIGNMENT IN MADE.

CHAR(27): ARRAY CONTAINING THE CHARACTERISTICS FOR THE STUDENT CURRENTLY BEING CONSIDERED.

SEMI(12): ARRAY MAINTAINING CONSTRAINT INFORMATION FOR ATTEMPTS TO ASSIGN STUDENT TO SEMINARS.

RECN(575): ARRAY INDICATING WHICH RECORDS HAVE/HAVEN'T BEEN ASSIGNED PREVIOUSLY.

MS(3): ARRAY INDICATING MISSING SEMINARS IN A SQUADRON.

SQ$(4,7): ARRAY CONTAINING SCHOOL/SQUADRON PARAMETERS.

SL(4): ARRAY USED TO TRACK SL ASSIGNMENTS.

ASL(4): ARRAY USED TO TRACK ASL ASSIGNMENTS.

STAT$(27): ARRAY CONTAINING RULE ALPHA DESCRIPTORS.

RULE$(27): ARRAY CONTAINING RULE PRIORITIES.

SQ$(4,7), STAT(27,4,11), RULE$(27), ASSIGN(44,13,5), MS(3)

DIM TEMPINS(36), STAT$(27), ALOT(27,4,11), SL(4), ASL(4)

DIM CHAR(27), SEMI(12), RECN(575)


FIELD #3, 1 AS TOPS*, 1 AS SOPS, 1 AS PPBS*, 1 AS ACQLOG*, 1 AS SPAACE*, 1 AS TBD1*, 1 AS TBD2*, 1 AS TBD3*, 1 AS TBD4*, 1 AS TBDSS, 1 AS SLS, 1 AS ASLS, 1 AS SOSS, 1 AS ARIS, 1 AS SROS, 1 AS CCS, 1 AS IOS, 1 AS FEMS,
10980 REM 1 AS RNK$, 1 AS NOLINE$
10990 REM
11000 REM
11010 REM *************
11020 REM ** PROCEDURE:
11030 REM **
11040 REM
11050 SCREEN 0,1
11060 COLOR 2,0,1
11070 OPEN "0", #1, "ERROR.SMS": PRINT #1, "SMSS MIXER PROGRAM ERROR":CLOSE
11080 DELTA = +1
11090 GOSUB 13070 :REM ** RETRIEVE SCHOOL DATA **
11100 CLS: LN = 3
11110 LOCATE 2,18: PRINT "TURN THE TRACE OUTPUT OFF AT TERMINAL? (Y/N) ";
11120 INPUT "", CHS: LOCATE LN,18
11130 IF CH$ = "Y" OR CH$ = "y" THEN TRONS = 0: TRONS$ = "OFF": GOTO 11200
11140 IF CH$ = "N" OR CH$ = "n" THEN TRONS = 1: TRONS$ = "ON": GOTO 11200
11150 IF CH$ = "D" OR CH$ = "d" THEN TRONS = 2: TRONS$ = "ON": GOTO 11200
11160 LN = LN + 1
11170 IF LN = 25 THEN GOTO 11100
11180 PRINT "INCORRECT CHOICE - TRY AGAIN? ";
11190 GOTO 11120
11200 CLS: LOCATE 2,32: PRINT "MIX ";MIX;" RULES"
11210 LOCATE 4,21: PRINT "PROCESSING MIXER INPUT WITH TRACE ": TRONS$:PRINT
11220 FOR I = 1 TO 27: INPUT #1. RULE$(I): NEXT I :REM *9 INIT RULES
11230 CLOSE #1
11240 GOSUB 12850 :REM ** INITIALIZE RULE DESCRIPTORS **
11250 GOSUB 13770 :REM ** INITIALIZE STATISTICS **
11260 IOMODE$ = "I": GOSUB 12210 :REM ** PROCESS INPUT DATA **
11270 IOMODE$ = "O": GOSUB 13980 :REM ** INITIALIZE RANDOM # GENERATOR **
11280 OPEN "0", #1, "MIXERLOG.DAT" :REM ** ALLOC SQ SLOTS **
11290 IF MIX = 1 THEN GOSUB 19900
11300 GOSUB 14100 :REM ** ALLOC SEM SLOTS **
11310 CLS: LOCATE 2,30: PRINT "SMSS MIXING ROUTINE": PRINT
11320 PRIOR = 9
11330 GOSUB 16230 :REM ** ASGN SL/ASL **
11340 FOR PRIOR = 9 TO 1 STEP -1
11350 FOR INDEX = 3 TO 27
11360 IF VAL(RULE$(INDEX)) <> PRIOR THEN GOTO 11410
11370 IF PRIOR = 9 AND INDEX = 22 THEN GOTO 11410
11380 IF PRIOR = 9 AND INDEX = 19 THEN GOTO 11410
11390 IR = INDEX: RL = INDEX: RL$ = STAT$(IR)
11400 GOSUB 18700
11410 NEXT INDEX
11420 Next Prior
11430 IR = 0: INDEX = 0: RL = 0: PRIOR = 0: RL$ = "ALL OTHER": STAT$(O) = RL$
11440 GOSUB 18700 :REM ** PROCESS REST **
11450 CLOSE #1: GOSUB 20170 :REM ** OUTPUT RSLTS **
11460 OPEN ":", #1, "MIXERLOG.DAT" :REM ** FINISHED **
11470 FOR I = 1 TO 999
11480 IF EOF(1) GOTO 11510
11490 INPUT #1, TEMP$: LPRINT TMP$
11500 NEXT I
11510 CLOSE: GOSUB 11610
11520 REM ** OUTPUT STATS **
11530 END
11540 REM
11550 REM ** SUBROUTINE TO OUTPUT MIXER STATISTICS **
11560 REM **
11570 REM **
11580 LPRINT: LPRINT: LPRINT: LPRINT
11590 STAT$(0) = "SEMINAR ASSIGNMENTS"
11600 FOR I = 0 TO 27
11610 IF I = 1 OR I = 2 THEN GOTO 11710
11620 LPRINT "* STATISTICS ON "; STAT$(I); " PRI = "; RULE$(I)
11630 FOR J = 1 TO 4
11640 FOR K = I TO 11: LPRINT STAT(I,J,K);: NEXT K
11650 LPRINT " "
11660 NEXT J: LPRINT " "
11670 NEXT I
11680 LPRINT "* ASSIGNMENT ARRAY: "
11690 FOR I = 1 TO 44
11700 IF I > 11 THEN FS = 12 ELSE FS = I
11710 IF I > 22 THEN FS = 23
11720 IF I > 33 THEN FS = 34
11730 LPRINT "SEM: I: MIX1 STNTS=";
11740 FOR J = 1 TO 11: CHAR(J) = 0: NEXT J
11750 FOR J = 1 TO 13
11760 IF ASSIGN(I,J,1) = 0 GOTO 11820
11770 CHAR(ASSIGN(I,J,1)-FS+1) = CHAR(ASSIGN(I,J,1)-FS+1) + 1
11780 LPRINT ASSIGN(I,J,1);
11790 NEXT J
11800 TEMP = 0
11810 FOR J = 1 TO 11
11820 IF TEMP < CHAR(J) THEN TEMP = CHAR(J)
11830 NEXT J
11840 LPRINT " MAX": TEMP
11850 LPRINT "SEM: I: MIX2 STNTS=";
11860 FOR J = 1 TO 11: CHAR(J) = 0: NEXT J
11870 FOR J = 1 TO 13
11880 IF ASSIGN(I,J,2) = 0 GOTO 11940
11890 CHAR(ASSIGN(I,J,2)-FS+1) = CHAR(ASSIGN(I,J,2)-FS+1) + 1
11900 LPRINT ASSIGN(I,J,2);
11910 NEXT J
11920 NEXT J
11930 NEXT J
11940
TEMP = 0
FOR J = 1 TO 11
IF TEMP < CHAR(J) THEN TEMP = CHAR(J)
NEXT J
LPRINT " MAX="; TEMP
LPRINT "SEM:"; I; " RULE#=
FOR J = 1 TO 13
LPRINT ASSIGN(I,J,5);
NEXT J
LPRINT 
LPRINT " SEM:
FOR J = I TO 13
LPRINT ASSIGN(I,J,0);
NEXT J
LPRINT 
LPRINT 
NEXT I
RETURN
REM ** **
REM *******************
REM **
REM ** SUBROUTINE TO PROCESS INPUT DATA
REM **
REM **
REM OPEN "1", #1, "STU.ASC"
IF MEDIA$ <> "1" THEN GOTO 12260
OPEN "R", #2, "B:STDNT1.DAT"
OPEN "R", #3, "B:STDNT2.DAT"
GOTO 12280
OPEN "R", #2, "STDT1.DAT"
OPEN "R", #3, "STDT2.DAT"
FIELD #2, 27 AS SNAME$, 4 AS STN$, 1 AS ASQ$, 2 AS MIX1$, 2 AS MIX2$, 2 AS MIX3$, 2 AS MIXX$, 1 AS USAF$, 1 AS NOED$, 1 AS HORS$, 1 AS PILOT$, 1 AS NAV$, 1 AS SING$, 1 AS AFA$, 1 AS ARMY$, 1 AS REST$, 1 AS RACE$, 1 AS SLX$, 1 AS ASLX$, 1 AS PRIOR$
FIELD #3, 1 AS TOPS$, 1 AS SOPS$, 1 AS PPBS$, 1 AS ACQLOG$, 1 AS SPACE$, 1 AS TD1$, 1 AS TD2$, 1 AS TD3$, 1 AS TD4$, 1 AS TBD5$, 1 AS SL$, 1 AS ASL$, 1 AS SOS$, 1 AS ARI$, 1 AS SRO$, 1 AS CC$, 1 AS IO$, 1 AS FEM$, 1 AS RNK$, 1 AS NOLINE$
SI = 0: KO = 1: PRINT: PRINT " READING ";
FOR KS = 1 TO 575
IF KS/10 >= KO THEN PRINT ".": KO = KO + 1
STDT = SI
FOR J = 1 TO 36
IF EOF (1) THEN GOTO 12680
INPUT #1, TEMPIN$(J)
NEXT J
LSET SNAME$ = TEMPIN$(1): LSET IO$ = TEMPIN$(21)
12390 LSET ASQ$ = TEMPIN$(2): LSET USAF$ = TEMPIN$(22)
12400 LSET MIX1$ = TEMPIN$(3): LSET NOED$ = TEMPIN$(23)
12410 LSET MIX2$ = TEMPIN$(4): LSET HORG$ = TEMPIN$(24)
12420 LSET MIX3$ = TEMPIN$(5): LSET PILOT$ = TEMPIN$(25)
12430 LSET STN$ = TEMPIN$(6): LSET NAV$ = TEMPIN$(26)
12440 LSET MIX2$ = TEMPIN$(7): LSET SING$ = TEMPIN$(27)
12450 LSET AR1$ = TEMPIN$(8): LSET AFA$ = TEMPIN$(28)
12460 LSET CC$ = TEMPIN$(9): LSET ARMY$ = TEMPIN$(29)
12470 LSET SRO$ = TEMPIN$(10): LSET REST$ = TEMPIN$(30)
12480 LSET SL$ = TEMPIN$(11): LSET RACE$ = TEMPIN$(31)
12490 LSET ASL$ = TEMPIN$(12): LSET FEM$ = TEMPIN$(32)
12500 LSET PPBS$ = TEMPIN$(13): LSET RNK$ = TEMPIN$(33)
12510 LSET ACQLOG$ = TEMPIN$(14): LSET NOLINE$ = TEMPIN$(34)
12520 LSET SPACE$ = TEMPIN$(15): LSET SOPS$ = TEMPIN$(35)
12530 LSET TBD1$ = TEMPIN$(16): LSET TOPS$ = TEMPIN$(36)
12540 LSET TBD2$ = TEMPIN$(17): LSET SLX$ = "O"
12550 LSET TBD3$ = TEMPIN$(18): LSET ASLX$ = "O"
12560 LSET TBD4$ = TEMPIN$(19): LSET MIXX$ = "O"
12570 LSET TBD5$ = TEMPIN$(20): LSET PRIOR$ = "O"
12580 IF (10$="Y" OR 10$="y") AND RULE$(22)="O" AND MIX=3 GOTO 12670
12590 JJ = 0: KK = 0: GOSUB 13270: REM ** RECORD SCHOOL STATS **
12600 IF ASQ$ = " " GOTO 12650
12610 FOR J = 1 TO 4 : REM ** POST SQUADRON STATS **
12620 IF VAL(ASQ$) <> VAL(SQ$(J,1)) GOTO 12640
12630 JJ = J: KK = 0: GOSUB 13270: GOTO 12650
12640 NEXT J
12650 SI = SI + 1
12660 PUT #2,SI: PUT #3,SI
12670 NEXT KS
12680 STDNT = STAT(0,0,0): CLS
12690 LOCATE 2,25: PRINT "PROCESSED":KS-1;"STUDENTS INTO MIXER"
12700 LOCATE 3,20: PRINT STDNT; "STUDENTS KEPT FOR FURTHER PROCESSING"
12710 PRINT
12720 FOR I = 3 TO 27
12730 IF STAT(I,0,0) = 0 THEN GOTO 12750
12740 PRINT " THERE WERE "; STAT(I,0,0); "; "; STAT$(I); " STUDENTS"
12750 NEXT I
12760 PRINT " ***** ALL OTHER CHARACTERISTICS WERE NOT REPRESENTE"}
12770 CLOSE #1: RETURN
12780 REM
12790 REM **
12800 REM ***
12810 REM **
12820 REM **
12830 REM **
12840 REM
12850 STAT$(1) = "SEMINAR CHANGE": STAT$(2) = "BUDDY RULE"
12860 STAT$(3) = "SPACE SKILL": STAT$(4) = "PPBS SKILL"
12870 STAT$(5) = "TAC OPS SKILL": STAT$(6) = "STRAT OPS SKILL"
12880  STAT$(7) = "ACQ/LOG SKILL":  STAT$(8) = "PILOT"
12890  STAT$(9) = "NAVIGATOR":  STAT$(10) = "SINGLE/UNACCOMP"
12900  STAT$(11) = "USAFA GRADUATE":  STAT$(12) = "ARMY"
12910  STAT$(13) = "RES/NG/USN/USMC/CIV":  STAT$(14) = "MINORITY"
12920  STAT$(15) = "FEMALE":  STAT$(16) = "CAPTAIN"
12930  STAT$(17) = "NON LINE":  STAT$(18) = "SR ORG EXPERIENCE"
12940  STAT$(19) = "SOS":  STAT$(20) = "NON MASTER DEGREE"
12950  STAT$(21) = "USAFA":  STAT$(22) = "10"
12960  STAT$(23) = "UNLABELED":  STAT$(24) = "UNLABELED"
12970  STAT$(25) = "UNLABELED":  STAT$(26) = "UNLABELED"
12980  STAT$(27) = "UNLABELED"
12990  RETURN
13000  REM
13010  REM **
13020  REM ***SUBROUTINE TO RETRIEVE SCHOOL DATA***
13030  REM **
13040  REM **
13050  REM **
13060  REM
13070  OPEN "I",#1,"ENV.ASC"
13080  INPUT #1, MIX$, MEDIA$, PREV$
13090  FOR I = 1 TO 4
13100    ADJ = 0
13110    INPUT #1, SQ$(1,2), SQ$(1,3), SQ$(1,4), SQ$(1,5), SQ$(1,6)
13120    SQ$(1,1) = STR$(I)
13130    IF VAL(SQ$(1,4)) <> 0 THEN ADJ = ADJ + 1
13140    IF VAL(SQ$(1,5)) <> 0 THEN ADJ = ADJ + 1
13150    IF VAL(SQ$(1,6)) <> 0 THEN ADJ = ADJ + 1
13160    SQ$(1,7) = STR$(VAL(SQ$(1,3)) - VAL(SQ$(1,2)) - ADJ + 1)
13170  NEXT I
13180  MIX = VAL(MIX$)
13190  RETURN
13200  REM
13210  REM **
13220  REM ***SUBROUTINE TO UPDATE STATISTICS***
13230  REM **
13240  REM **
13250  REM **
13260  REM
13270  IF IOMODE$ = "O" THEN GOTO 13290
13280  IF JJ = 0 THEN STAT(0,0,0) = STAT(0,0,0) + DELTA
13290  IF KK <> 0 THEN STAT(0,JJ,KK) = STAT(0,JJ,KK) + DELTA
13300  IF JJ <> 0 THEN STAT(0,JJ,0) = STAT(0,JJ,0) + DELTA
13310  GOSUB 13440
13320  FOR II = 3 TO 27
13330    IF CHAR(II) = 0 THEN GOTO 13350
13340    STAT(II,JJ,KK) = STAT(II,JJ,KK) + DELTA
13350  NEXT II
13360  RETURN

84
REM SUBROUTINE TO INITIALIZE CHARACTERISTICS ARRAY

IF SPAACE$ = "Y" OR SPAACE$ = "y" THEN CHAR(3) = 1 ELSE CHAR(3) = 0
IF PPBS$ = "Y" OR PPBS$ = "y" THEN CHAR(4) = 1 ELSE CHAR(4) = 0
IF TOPS$ = "Y" OR TOPS$ = "y" THEN CHAR(5) = 1 ELSE CHAR(5) = 0
IF SOPS$ = "Y" OR SOPS$ = "y" THEN CHAR(6) = 1 ELSE CHAR(6) = 0
IF ACQLOG$ = "Y" OR ACQLOG$ = "y" THEN CHAR(7) = 1 ELSE CHAR(7) = 0
IF PILOTS = "Y" OR PILOTS = "y" THEN CHAR(8) = 1 ELSE CHAR(8) = 0
IF NAV$ = "Y" OR NAV$ = "y" THEN CHAR(9) = 1 ELSE CHAR(9) = 0
IF SINGS = "Y" OR SINGS = "y" THEN CHAR(10) = 1 ELSE CHAR(10) = 0
IF AFA$ = "Y" OR AFA$ = "y" THEN CHAR(11) = 1 ELSE CHAR(11) = 0
IF ARMYS = "Y" OR ARMYS = "y" THEN CHAR(12) = 1 ELSE CHAR(12) = 0
IF REST$ = "Y" OR REST$ = "y" THEN CHAR(13) = 1 ELSE CHAR(13) = 0
IF RACES = "Y" OR RACES = "y" THEN CHAR(14) = 1 ELSE CHAR(14) = 0
IF RNK$ = "Y" OR RNK$ = "y" THEN CHAR(16) = 1 ELSE CHAR(16) = 0
IF NOLINE$ = "Y" OR NOLINE$ = "y" THEN CHAR(17) = 1 ELSE CHAR(17) = 0
IF HORG$ = "Y" OR HORG$ = "y" THEN CHAR(18) = 1 ELSE CHAR(18) = 0
IF NOED$ = "Y" OR NOED$ = "y" THEN CHAR(19) = 1 ELSE CHAR(19) = 0
IF USAF$ = "Y" OR USAF$ = "y" THEN CHAR(21) = 1 ELSE CHAR(21) = 0
IF 10$ = "Y" OR 10$ = "y" THEN CHAR(22) = 1 ELSE CHAR(22) = 0
IF TBD1$ = "Y" OR TBD1$ = "y" THEN CHAR(23) = 1 ELSE CHAR(23) = 0
IF TBD2$ = "Y" OR TBD2$ = "y" THEN CHAR(24) = 1 ELSE CHAR(24) = 0
IF TBD3$ = "Y" OR TBD3$ = "y" THEN CHAR(25) = 1 ELSE CHAR(25) = 0
IF TBD4$ = "Y" OR TBD4$ = "y" THEN CHAR(26) = 1 ELSE CHAR(26) = 0
IF TBD5$ = "Y" OR TBD5$ = "y" THEN CHAR(27) = 1 ELSE CHAR(27) = 0

RETURN

REM SUBROUTINE TO INITIALIZE/OUTPUT STATS & SEMINAR ASSIGNMENTS

PRINT " INITIALIZING ";
FOR I = 0 TO 27
  PRINT ";.";
  FOR J = 0 TO 4
    FOR K = 0 TO 11: STAT(I,J,K) = 0: NEXT K
    NEXT J
  NEXT I
FOR J = 0 TO 13
  PRINT ";.";
FOR I = 0 TO 44
FOR K = 0 TO 5: ASSIGN(I, J, K) = 0: NEXT K
NEXT I
NEXT J
RETURN

REM ********************************************************************
REM SUBROUTINE TO INITIALIZE RANDOM NUMBER GENERATOR
REM ********************************************************************

XT = TIME: XT = XT - (INT(XT/100) * 100)
FOR I = 1 TO XT: X = RND: NEXT I
X = X * 10000: RANDOMIZE X
STONT = STAT(0, 0, 0)
RETURN

REM ********************************************************************
REM SUBROUTINE TO RANDOMLY ALLOCATE CHARACTERISTICS TO SEMINARS
REM ********************************************************************

PRINT: PRINT " ALLOCATING SLOTS TO SEMINARS ";
FOR J = 1 TO 4
GOSUB 14870: KO = 1
FOR I = 3 TO 27
IF I/4 > KO THEN PRINT ";": KO = KO + 1
IF MIX = 1 THEN STAT(I, J, 0) = ALOT(I, J, 0)
ALOT(I, 0, 0) = STAT(I, 0, 0)
IF STAT(I, J, 0) = 0 THEN GOTO 14390
TA = STAT(I, J, 0) / NS : REM ** TEMPORARY ALLOCATION **
ALOT(I, J, 0) = STAT(I, J, 0)
IF TA < 1 THEN TA = 0: GOTO 14270
FOR K = 1 TO LS - FS + 1
IF K = MS(1) - FS + 1 THEN GOTO 14260
IF K = MS(2) - FS + 1 THEN GOTO 14260
IF K = MS(3) - FS + 1 THEN GOTO 14260
ALOT(I, J, K) = INT(TA) : REM ** GIVE SEMINAR SHARE **
NEXT K
IF STAT(I, J, 0) - (NS * INT(TA)) = 0 THEN GOTO 14390
ADJ = 0
FOR K = 1 TO STAT(I, J, 0) - (NS * INT(TA)) : REM ** GIVE REST **
ADJ = ADJ + 1
IF ADJ > 11 THEN ADJ = 1
IF ADJ = MS(1) - FS + 1 THEN GOTO 14300
IF ADJ = MS(2) - FS + 1 THEN GOTO 14300
IF ADJ = MS(3) - FS + 1 THEN GOTO 14300
ALOT(I,J,ADJ) = ALOT(I,J,ADJ) + 1
ALOT(O,J,ADJ) = ALOT(O,J,ADJ) + 1

NEXT K
IF MIX = 1 THEN STAT(1,J,0) = 0
NEXT I
ALOT(O,J,0) = STAT(O,J,0)
TA = 25: ADJ = 0
FOR K = 1 TO NS
ADJ = ADJ + 1
IF ADJ = MS(1) - FS + 1 THEN GOTO 14430
IF ADJ = MS(2) - FS + 1 THEN GOTO 14430
IF ADJ = MS(3) - FS + 1 THEN GOTO 14430
ALOT(O,J,ADJ) = INT(ALOT(O,J,ADJ)/NS)
IF TA = 25 THEN TA = ALOT(O,J,0) - (ALOT(O,J,ADJ) * NS)
IF TA = 0 THEN GOTO 14520
ALOT(O,J,ADJ) = ALOT(O,J,ADJ) + 1
TA = TA - 1
NEXT K
STAT(O,J,0) = 0
NEXT J
PRINT
RETURN

REM
**
**
SUBROUTINE TO RANDOMLY ASSIGN STUDENTS TO TEMPORARY SEMINARS
**

RS = RND
FOR LL = 1 TO LS - FS + 1
IF RS > LL/(LS - FS + 1) GOTO 14750
TSEM = FS + LL - 1: PS = LL
FOR R = 1 TO 3
IF TSEM = MS(R) THEN GOTO 14720
NEXT R
GOTO 14760
IF LL > 11 THEN LL = 1
GOTO 14670
NEXT LL
IF MIX = 1 THEN KK = 0 ELSE KK = PS
IF MIX = 1 THEN JJ = PS ELSE JJ = J
IF TRONS > 1 THEN PRINT "* DEBUG: J=";J;" JJ=";JJ;" KK=";KK;" PS=";PS
RETURN

REM
**
**

SUBROUTINE TO GET SEMINAR DATA FOR SQUADRON

FS = VAL(SQ$(J,2)) :REM ** FIRST SEMINAR IN SQ **
LS = VAL(SQ$(J,3)) :REM ** LAST SEMINAR IN SQ **
MS(1) = VAL(SQ$(J,4)) :REM ** MISSING SEMINAR IN SQ **
MS(2) = VAL(SQ$(J,5)) :REM ** MISSING SEMINAR IN SQ **
MS(3) = VAL(SQ$(J,6)) :REM ** MISSING SEMINAR IN SQ **
NS = VAL(SQ$(J,7)) :REM ** # SEMINARS IN SQ **
TS = STAT(0,J,0) :REM ** TOTAL STUDENTS IN SQ **
RETURN

SUBROUTINE TO DETERMINE CURRENT SQUADRON ASSIGNMENT

FOR J = 0 TO 4
IF VAL(ASQ$) = VAL(SQ$(J,1)) THEN GOSUB 14870: GOTO 15060
NEXT J
J = 0
JJ = J
IF TRONS > 1 THEN PRINT "* DEBUG: FINDING CORRECT SQUADRON - J="; J
RETURN

SUBROUTINE TO FIND OFFSET FOR PRE-ASSIGNED STUDENTS

TSEM = 0
IF MIX = 1 THEN TSEM = VAL(MIX1$)
IF MIX = 2 THEN TSEM = VAL(MIX2$)
IF MIX = 3 THEN TSEM = VAL(MIX3$)
GOSUB 15290
RETURN

SUBROUTINE TO CHECK AGAINST MISSING SEMINARS

IF TSEM = 0 THEN GOTO 15370
KK = TSEM - FS + 1
IF KK >= 0 THEN GOTO 15340
PRINT "*** ERROR *** SQUADRON ";ASQ$;" AND SEMINAR ";TSEM;" MISMATCH"
PRINT "WILL REASSIGN TO NEW SEMINAR ": GOSUB 14560
IF TSEM = MS(1) AND TSEM <> 0 THEN GOSUB 14640: GOTO 15370
IF TSEM = MS(2) AND TSEM <> 0 THEN GOSUB 14640: GOTO 15370
IF TSEM = MS(3) AND TSEM <> 0 THEN GOSUB 14640
IF TSEM > 1 THEN PRINT ": DEBUG: TSEM=";TSEM;" FINDING CURRENT SMNR"
RETURN

* SUBROUTINE TO CHECK ON PREVIOUSLY ASSIGNED STUDENTS *
IF SWAPFLG$ = "Y" GOTO 15480
IF SEMI(KK) = 100 GOTO 15680
MXIFLG = 0: MX2FLG = 0: MX3FLG$ = "N": SASN$ = "Y"
CN = 0: MIX1CNT = 0: MIX2CNT = 0
IF SWITCH = 0 THEN SEMI(KK) = 100
IF TSEM = VAL(MIX1$) AND MIX = 1 THEN GOTO 15680
IF TSEM = VAL(MIX2$) AND MIX = 3 THEN GOTO 15680
FOR AB = 1 TO ASSIGN(TSEM,O,O)
  IF MIX <> 1 AND ASSIGN(TSEM,AB,1) = VAL(MIX1$) THEN MX1FLG = 1
  IF MIX = 3 AND ASSIGN(TSEM,AB,2) = VAL(MIX2$) THEN MX2FLG = 1
  IF MX1FLG = 1 AND MX2FLG = 0 THEN GOTO 15620
  IF MX1FLG = 1 AND MX2FLG = 1 THEN MX3FLG$ = "Y": GOTO 15680
  IF SLFLAG$ = "Y" AND AB = 1 THEN GOTO 15680
  IF MX1FLG = 1 THEN MIX1CNT = MIX1CNT + 1
  IF MX2FLG = 1 THEN MIX2CNT = MIX2CNT + 1
  CN = CN + 1: MX1FLG = 0: MX2FLG = 0
NEXT AB
IF SLFLAG$ = "Y" GOTO 15690
IF CN >= MAXSTDNTS THEN GOTO 15670
IF SWITCH = 0 THEN SEMI(KK) = 0: SEMI(12) = 0
GOTO 15690
IF SWITCH = 0 THEN SEMI(KK) = CN + 300
SASN$ = "N"
IF TONS > 1 THEN PRINT ": DEBUG: Prev=";SASN$;SLFLAG$;MX3FLG$;CN
RETURN

* SUBROUTINE TO CHECK ALLOCATION RESTRICTIONS *
IF SWAPFLG$ = "Y" GOTO 15810
IF SEMI(KK) > 300 THEN SASN$ = "N": GOTO 16000
IF SEMI(KK) = 100 THEN SASN$ = "N": GOTO 16000
SASN$ = "Y"
15820 GOSUB 13440
15830 FOR II = 0 TO 27
15840 IF (II = 1 OR II = 2) THEN GOTO 15990
15850 IF II = 0 GOTO 15870
15860 IF CHAR(II) = 0 THEN GOTO 15990
15870 IF STAT(II,JJ,KK) < ALOT(II,JJ,KK) THEN GOTO 15990
15880 FOR LL = 1 TO 11
15890 IF ALOT(II,JJ,LL) <= ALOT(II,JJ,KK) THEN GOTO 15950
15900 IF STAT(II,JJ,LL) >= ALOT(II,JJ,KK) THEN GOTO 15950
15910 TEMP = ALOT(II,JJ,KK)
15920 ALOT(II,JJ,KK) = ALOT(II,JJ,LL)
15930 ALOT(II,JJ,LL) = TEMP
15940 GOTO 15990
15950 NEXT LL
15960 IF SWITCH = 0 THEN SEMI(KK) = II + 200
15970 IF SWITCH = 0 AND II = 0 THEN SEMI(KK) = 100
15980 SASN$ = "N": GOTO 16000
15990 NEXT II
16000 RETURN
16010 REM
16020 REM **
16030 REM ********************************************
16040 REM **
16050 REM ** SUBROUTINE TO ASSIGN TO NEW MIX **
16060 REM **
16070 REM
16080 IF TSEM < 10 THEN Z = 1 ELSE Z = 2
16090 TSEM$ = MID$(STR$(TSEM),2,Z)
16100 IF TSEM < 10 THEN TSEM$ = "0"+MID$(TSEM$,1,1)
16110 RSET MIXX$ = TSEM$
16120 RSET PRIOR$ = STR*(PRIOR)
16130 IF MIX = 1 THEN LSET ASQ$ = SQS(PS,1)
16140 SASN$ = "Y"
16150 RETURN
16160 REM
16170 REM **
16180 REM ********************************************
16190 REM **
16200 REM ** SUBROUTINE TO ASSIGN SL & ASL TO EACH SEMINAR **
16210 REM **
16220 REM
16230 PRINT: PRINT " ASSIGNING SL & ASL ( PRIORITY = 9 )": PRINT
16240 SLFLAG$ = "Y": ARFLAG$ = "N"
16250 SLASLF = 0: MAXSTDNTS = 1
16260 FOR J = 1 TO 4
16270 GOSUB 14870: SL(J) = NS: ASL(J) = NS
16280 NEXT J
16290 FOR J = 1 TO 44
16300 ASSIGN(J,0,0) = 2
16310 NEXT J
16320 FORSSI = 1 TO STDNT
16330 S1 = SS1
16340 GET 2, S1: GET 3, S1
16350 IF TRAN S > 1 THEN PRINT "* DEBUG: RCRD="; S1; SNAME$ 16360 IF MIX = 1 GOTO 16410
16370 IF VAL(RULE$(22)) = 9 AND (10$ = "Y" OR 10$ = "y") GOTO 16400
16380 IF VAL(RULE$(19)) = 9 AND (SOS$ = "Y" OR SOS$ = "y") GOTO 16400
16390 GOTO 16410
16400 LSET MIX3$ = MIX1$
16410 GOSUB 15020 :REM ** FIND CORRECT SQUADRON **
16420 GOSUB 14870 :REM ** FIND SEMINAR DATA **
16430 GOSUB 15160 :REM ** FIND OFFSET **
16440 LIS = 0
16450 IF TSEM > 0 THEN LIS = 1: KK = TSEM - FS + 1 :REM ** LOCK-IN **
16460 IOH$ = "
16470 IF 10$ = "Y" THEN IOH$ = " INTERNATIONAL OFFICER "
16480 IF 10$ = "y" THEN IOH$ = " INTERNATIONAL OFFICER "
16490 IF SOS$ = "Y" THEN IOH$ = " SOS INSTRUCTOR "
16500 IF SOS$ = "y" THEN IOH$ = " SOS INSTRUCTOR "
16510 IF SLASLF = 1 GOTO 17080 :REM ** DONE **
16520 IF IOH$ = "Y" OR IOH$ = "y" THEN GOTO 17080
16530 IF SL$ = "Y" OR SL$ = "y" THEN GOTO 17080
16540 IF CC$ = "Y" OR CC$ = "y" THEN GOTO 17080
16550 IF SR0$ = "Y" OR SR0$ = "y" THEN GOTO 17080
16560 IF MIX < 3 AND (SOS$ = "Y" OR SOS$ = "y") THEN GOTO 17080
16570 IF MIX = 1 AND (USAF$ <> "Y" AND USAF$ <> "y") THEN GOTO 16800
16580 IF MIX = 1 AND (NOLINES = "Y" OR NOLINES = "y") THEN GOTO 16800
16590 IF TRAN S > 1 THEN PRINT "* DEBUG: SL/ASL - LIS="; LIS; SNAME$
16600 SSL = 0
16610 FOR I = 0 TO 12: SEMI(I) = 299: NEXT I
16620 IF MS(1) <> 0 THEN SEMI(MS(1) - FS + 1) = 100
16630 IF MS(2) <> 0 THEN SEMI(MS(2) - FS + 1) = 100
16640 IF MS(3) <> 0 THEN SEMI(MS(3) - FS + 1) = 100
16650 IF LIS = 0 THEN GOSUB 16440 :REM ** RANDOMLY ASSIGN **
16660 IF TRAN S > 1 THEN PRINT "* DEBUG: TSEM="; TSEM; JJ; KK; SSL; FS; LIS
16670 IF SEMI(KK) = 100 AND LIS = 0 THEN GOTO 16650
16680 IF SSL = 1 GOTO 16800
16690 IF SL(JJ) = 0 GOTO 16800
16700 IF ASSIGN(TSEM, 1, 0) > 0 GOTO 16750
16710 SWITCH = 2: GOSUB 15460 :REM ** CHECK CLASSMATES **
16720 IF SASN$ = "N" GOTO 16750
16730 SWITCH = 2: GOSUB 15780 :REM ** CHECK CONSTRAINTS **
16740 IF SASN$ = "Y" GOTO 16960 :REM ** ASSIGN AS THE SL **
16750 IF LIS = 1 THEN GOTO 16800 :REM ** TRY TO USE AS ASL **
16760 SEMI(KK) = 100
16770 FOR I = 1 TO 11
16780 IF SEMI(I) = 299 GOTO 16650 :REM ** TRY ANOTHER SEMINAR **
16790 NEXT I
16800 SSL = 1: GOTO 16610
16810 IF ASLS = "Y" OR ASL$ = "Y" THEN GOTO 16830
16820 IF ASL(JJ) <> 0 GOTO 16840 :REM ** DONE WITH ASL'S? **
16830 IF ASSIGN(TSEM,2,0) > 0 GOTO 16900
16840 IF ASSIGN(TSEM,1,0) = 0 GOTO 16900
16850 SWITCH = 2: GOSUB 15460 :REM ** CHECK CLASSMATES **
16860 IF ASLS = "N" GOTO 16900
16870 SWITCH = 2: GOSUB 15780 :REM ** CHECK CONSTRAINTS **
16880 IF ASL(JJ) = 0 GOTO 16900
16890 FOR I = 1 TO 11
16900 SSL = 0 THEN ASG$ = "ASL" ELSE ASG$ = "SL"
16910 FOR I = 1 TO 4
16920 IF SL(I) <> 0 GOTO 16770
16930 IF ASL(I) = 0 GOTO 16770
16940 NEXT I
16950 FOR I = 1 TO 11
16960 IF SSL = 0 THEN ASG$ = "SL" ELSE ASG$ = "ASL"
16970 IF TRONS = 0 THEN GOTO 17000
16980 PRINT "ASSIGNING ";SNAME$;" TO ";SI;" AS ":
16990 PRINT ASG$;" FOR SEMINAR ";TSEM;"" :REM ** UPDATE RECORDS **
17000 GOSUB 16080: GOSUB 17220 :REM ** UPDATE RECORDS **
17010 IF SSL = 0 THEN SL(JJ) = SL(JJ) - 1 ELSE ASL(JJ) = ASL(JJ) - 1
17020 FOR I = 1 TO 4
17030 IF SL(I) <> 0 GOTO 17070
17040 IF ASL(I) <> 0 GOTO 17070
17050 NEXT I
17060 GOTO 17120
17070 IF TSEM = 0 GOTO 17120 :REM ** CHECK FOR LOCK-INS **
17080 IF TRONS > 0 THEN PRINT ";LOCK IN "; TSEM; I0H$;
17090 PRINT ";LOCK IN "; TSEM; I0H$;
17100 GOSUB 15020: GOSUB 14870: GOSUB 16080: GOSUB 17390
17110 NEXT SSI
17120 SLFLAG$ = "N" : ARFLAF$ = "N" :REM ** ALL SL'S & ASL'S ASSIGNED **
17130 RETURN
17140 REM
17150 REM **
17160 REM **
17170 REM ******************************************
17180 REM **
17190 REM ** SUBROUTINE TO UPDATE STUDENT RECORDS **
17200 REM **
17210 REM **
17220 RETEMP = ASSIGN(VAL(MIXX$),0,0)
17230 IF SSL = 1 GOTO 17270
17240 LSET SLX$ = "Y" :REM ** UPDATE SL **
17250 ASSIGN(VAL(MIXX$),0,0) = 0
17260 GOTO 17290
17270 LSET ASLX$ = "Y" :REM ** UPDATE ASL **
17280 ASSIGN(VAL(MIXX$),0,0) = 1
GOSUB 17390
ASSIGN(VAL(MIXX$),0,0) = RETEMP
RETURN
REM **
REM # SUBROUTINE TO WRITE OUT STUDENT RECORDS
REM **
REM
RSET PRIOR$ = STR$(PRIOR)
ASSIGN(VAL(MIXX$),0,0) = ASSIGN(VAL(MIXX$),0,0) + 1
ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),0) = SI
IF MIX=2 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),1)=VAL(MIX1$)
IF MIX=3 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),1)=VAL(MIX1$)
IF MIX=3 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),2)=VAL(MIX2$)
IF MIX=1 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),1)=VAL(MIX1$)
IF MIX=2 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),2)=VAL(MIX2$)
IF MIX=3 THEN ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),3)=VAL(MIX3$)
ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),4) = PRIOR
ASSIGN(VAL(MIXX$),ASSIGN(VAL(MIXX$),0,0),5) = IR
RECNUM(SI) = PRIOR + 50
IF PRVFLGS = "Y" THEN Prev$ = STR$(MAXSTDNTS); PRVFLGS = "N"
FSFLG$ = "IN"
PUT #2,SI: PUT #3,SI
DELTA = +1: GOSUB 13270
IF TRONS < 2 GOTO 17620
PRINT ";# DEBUG: SLT="; SLT; "SWTCH"
IF TRONS > 1 THEN PRINT ";# DEBUG: SLT="; SLT; "SWTCH"
IF STAT(IR,JJ,SLT) < ALOT(IR,JJ,SLT) THEN GOTO 18580
IF SEMI(SLT) < 200 OR SEMI(SLT) > 300 THEN GOTO 18590
TEMPI = SEMI(SLT) - 200
TSEM = SEMI(SLT) - 1
17780     SWITCH = 2: GOSUB 15460: SWITCH = 0
17790     IF MX3FLG$ = "Y" GOTO 18580
17800     FOR PTR = ASSIGN(TSEM,0,0) TO 3 STEP -1
17810     IF TRONS > 1 THEN PRINT "* DEBUG: PTR="; PTR; SEMI(SLT)
17820     IF TRONS > 1 THEN LPRINT "* DEBUG: PTR="; PTR; SEMI(SLT)
17830     IF ASSIGN(TSEM,PTR,4) > PRIOR + OMEGA THEN GOTO 18580
17840     SI = ASSIGN(TSEM,PTR,0)
17850     GET #2,SI: GET #3,SI
17860     GOSUB 13440
17870     IF OMEGA <> 0 GOTO 17900
17880     IF FSFLG$ = "Y" AND IR <> 0 AND CHAR(IR) <> 1 GOTO 18570
17890     IF FSFLG$ = "N" AND CHAR(TEMP1) <> 1 GOTO 18570
17900     DELTA = -1: KK = SLT: GOSUB 13270 :REM ** REMOVE **
17910     GET #2,SK: GET #3,SK
17920     GOSUB 13440
17930     IF SWITCH = 0 GOTO 17970
17940     FOR YY = 3 TO 27
17950     IF VAL(RULE$(YY)) = 0 THEN CHAR(YY) = 0
17960     NEXT YY
17970     SWITCH = 2: KK = SLT
17980     GOSUB 15780: SWITCH = 0: TEMP$ = SNAME$: TMP$ = STN$
17990     GET #2,SI: GET #3,SI
18000     GOSUB 13440
18010     DELTA = +1: KK = SLT: GOSUB 13270
18020     IF FSFLG$ = "Y" GOTO 18040
18030     IF SASN$ = "N" GOTO 18570
18040     IF SWITCH = 0 GOTO 18080
18050     FOR YY = 3 TO 27
18060     IF VAL(RULE$(YY)) = 0 THEN CHAR(YY) = 0
18070     NEXT YY
18080     SWAPFLGS = "Y"
18090     FOR SMNR = FS TO LS
18100     IF TRONS > 1 THEN PRINT "* DEBUG: SMNR="; SMNR
18110     IF TRONS > 1 THEN LPRINT "* DEBUG: SMNR="; SMNR; FSFLG$; FS
18120     IF SMNR = TSEM GOTO 18550
18130     IF SMNR = MS(1) GOTO 18550
18140     IF SMNR = MS(2) GOTO 18550
18150     IF SMNR = MS(3) GOTO 18550
18160     IF (MIX = 2 OR MIX = 3) AND SMNR = VAL(MIX1$) GOTO 18550
18170     IF MIX = 3 AND SMNR = VAL(MIX2$) THEN GOTO 18550
18180     XX = SMNR - FS + 1
18190     SWITCH = 2: KK = XX
18200     GOSUB 15780: SWITCH = 0
18210     IF TRONS > 1 THEN LPRINT "* DEBUG: ALOC="; SASN$; KK; SNAME$
18220     IF FSFLG$ = "Y" GOTO 18250
18230     IF SASN$ = "N" GOTO 18550
18240     TSEM = SMNR
18250     SWITCH = 2: GOSUB 15460: SWITCH = 0: TSEM = SLT + FS - 1
18260     IF TRONS > 1 THEN LPRINT "* DEBUG: PRV="; SASN$; MX3FLG$; CN
18270 IF MX3FLG$ = "Y" GOTO 18550
18280 IF SASN$ = "N" THEN MXFLG$ = "Y": GOTO 18550
18290 IF TRONS = 0 GOTO 18320
18300 PRINT " *** EXCHANGING *** " ; TEMP$; " WITH " ; SNAME$; " ;
18310 IF LOGFLG$ <> "Y" THEN GOTO 18350
18320 PRINT #1, "EXCHANGING " ; (STU # ; TMP$ ; ;) " ; TEMP$; " WITH " ; (STU # ; STN$ ; ;) " ; SNAME$;
18330 FOR YY = PTR TO 13
18340 TEMPA = ASSIGN(TSEM, PTR, 5)
18350 FOR R = 0 TO 5
18360 IF YY = 13 THEN ASSIGN(TSEM, 13, R) = 0: GOTO 18400
18370 ASSIGN(TSEM, YY, R) = ASSIGN(TSEM, YY + 1, R)
18380 IF R = 5 AND ASSIGN(TSEM, YY, 0) = 0 THEN GOTO 18430
18390 NEXT R
18400 NEXT YY
18410 ASSIGN(TSEM, 0, 0) = YY - 1
18420 TSEM = SMNR: IR = TEMPA
18430 IF LOGFLG$ <> "Y" THEN GOTO 18470
18440 PRINT #1, "ASSIGNING " ; SNAME$ ; ; ; S1 ; ; TO SEMINAR " ; TSEM
18450 IF TRONS = 0 THEN GOTO 18490
18460 PRINT " ASSIGNING " ; SNAME$ ; ; ; S1 ; ; TO SEMINAR " ; TSEM
18470 GOSUB 16080: GOSUB 17390
18480 IR = INDEX
18490 DELTA = -1: KK = SLT: GOSUB 13270
18500 TSEM = SLT + FS - 1
18510 SWAPFLG$ = "N"
18520 GOTO 18610
18530 NEXT SMNR
18540 SWAPFLG$ = "N"
18550 NEXT PTR
18560 SEMI(SLT) = 100
18570 NEXT SLT
18580 EXIT
18590 NP = STAT(IR, 0, 0)
18600 PRINT " ASSIGNING REMAINING " ; RL$ ; " STUDENTS " ;
18610 PRINT " ( PRIORITY = " ; PRIOR ; ")"
18620 RETURN
18630 REM
18640 REM **
18650 REM **
18660 REM **
18670 REM ** SUBROUTINE TO PROCESS REMAINING STUDENTS **
18680 REM **
18690 REM
18700 NP = STAT(IR, 0, 0)
18710 PRINT
18720 PRINT " ASSIGNING REMAINING " ; RL$ ; " STUDENTS " ;
18730 PRINT " ( PRIORITY = " ; PRIOR ; ")"
18740 PRINT
18750 NP = STAT(IR, 0, 0)
18760 IF NP = 0 THEN GOTO 19820
18770 ARFLAG$ = "N"; SLFLAG$ = "N"; MXFLG$ = "N"; PRVFLG$ = "N"; FSFLG$ = "N"
18780 FOR SSI = 1 TO STDNT
18790 SI = SSI
18800 IF RECNUM(SI) <> 0 THEN GOTO 19810
18810 GET #2, SI: GET #3, SI
18820 LOGFLGS = "N": OMEGA = 0
18830 IF IR = 0 THEN GOTO 18860
18840 GOSUB 13440
18850 IF CHAR(IR) = 0 THEN GOTO 19810
18860 GOSUB 15020: REM ** FIND SQUADRON DATA **
18870 GOSUB 14870: REM ** FIND SEMINAR DATA **
18880 GOSUB 15160: REM ** FIND OFFSET **
18890 FOR I = 0 TO 12: SEMICI) = 0: NEXT I
18900 IF MS(1) <> 0 THEN SEMI(MS(1) - FS + 1) = 100
18910 IF MS(2) <> 0 THEN SEMI(MS(2) - FS + 1) = 100
18920 IF MS(3) <> 0 THEN SEMI(MS(3) - FS + 1) = 100
18930 GOSUB 14640: REM ** RANDOMLY ASSIGN **
18940 IF TRONS > 1 THEN PRINT "* DEBUG: TSEM=": TSEM; JJ; KK
18950 SWITCH = 0: GOSUB 15460: REM ** CHECK PREV CLASSMATES **
18960 IF SASN$ = "N" THEN GOTO 18990
18970 SWITCH = 0: GOSUB 15780: REM ** CHECK ON CONSTRAINTS **
18980 IF SASN$ = "Y" THEN GOTO 19740
18990 FOR I = 1 TO 11
19000 IF SEMI(I) = 0 THEN GOTO 19030
19010 IF SEMI(I) > 300 THEN SEMI(0) = 1: GOTO 19040
19020 IF SEMI(I) > 200 THEN SEMI(12) = 1
19030 NEXT I
19040 IF TRONS < 2 GOTO 19090
19050 PRINT "* DEBUG: SEMI=";
19060 FOR I = 0 TO 12: PRINT SEMI(I);: NEXT I
19070 PRINT SI; SK: PRINT
19080 IF SEMI(12) = 1 THEN GOTO 19680
19090 IF SEMI(0) = 1 GOTO 19140
19100 IF PRVFLG$ = "N" GOTO 19130
19110 PREV$ = STR$(MAXSTDNTS): PRVFLG$ = "N": GOTO 19340
19120 IF MXFLG$ = "Y" THEN GOTO 19200 ELSE GOTO 19330
19130 SEMI(0) = 0
19140 FOR I = 1 TO 11
19150 IF SEMI(I) = 100 GOTO 19190
19160 IF MAXSTDNTS <> SEMI(I) - 300 GOTO 19190
19170 SEMI(I) = 0
19180 NEXT I
19190 MAXSTDNTS = MAXSTDNTS + 1
19200 IF MXFLG$ = "Y" THEN MXFLG$ = "N": GOTO 18890
19210 IF MAXSTDNTS <= VAL(PREV$) GOTO 18890
19220 IF MAXSTDNTS > 9 THEN MAXSTDNTS = VAL(PREV$): GOTO 18990
19230 PRVFLG$ = "Y": LOGFLG$ = "Y"
**WARNING *** UNABLE TO ASSIGN ( *: ; $; ) ; NAME$ TO SEMINAR WITH LESS THAN**;

IF TRONS = 0 THEN GOTO 18890
PRINT " *** WARNING *** UNABLE TO ASSIGN ( *: ; $; ) ; NAME$
PRINT " TO SEMINAR WITH LESS THAN**;
PRINT MAXSTDNTS - 1;"PREVIOUS CLASSMATES"
GOTO 18890
IF LOGFLGS = "Y" THEN GOTO 19390
IF TRONS = 0 THEN GOTO 19360
PRINT " *** FORCING ASSIGNMENT FOR *; NAME*; (STU #:; STN$; )"
PRINT NI,1.(STU N; STNS; )-RULE FOR ":.STAT$(IR)
LOGFLG$ = "Y
SWTCH 0: KK = 0
TEMP = 15: LL = 0
FOR PTR = 1 TO 11
IF SEMI(PTR) = SWTCH GOTO 19470
IF MIX <> 1 AND VAL(MIX1$) = PTR + FS - 1 THEN GOTO 19470
IF MIX = 3 AND VAL(MIX2$) = PTR + FS - 1 THEN GOTO 19470
IF STAT(0, JJ, PTR) >= ALOT(0, JJ, PTR) GOTO 19470
IF TEMP > STAT(0, JJ, PTR) THEN TEMP = STAT(0, JJ, PTR): LL = PTR
NEXT PTR
KK = LL: TSEM = KK + FS - 1
IF TRONS > 1 THEN PRINT*" DEBUG: TSEM"; TSEM; KK; SWTCH; FSFLG$; OMEGA
IF FSFLG$ = "N" GOTO 19600
IF KK <> 0 THEN GOTO 19670
IF OMEGA + PRIOR < 5 THEN OMEGA = OMEGA + 1: GOTO 198890
PRINT "."; NAME$; " ( *: ; ) ; STN$; ")
PRINT "."; NAME$; " ( *: ; ) ; STN$; ")
PRINT "."; NAME$; " ( *: ; ) ; STN$; ")
GOTO 19810
IF SWTCH > 1 THEN FSFLG$ = "Y": SWTCH = 0: GOTO 18860
IF TEMP = 15 THEN SWTCH = SWTCH + 1: GOTO 19410
IF STAT(IR, JJ, KK) = ALOT(IR, JJ, KK) THEN SEMI(KK) = SWTCH: GOSUB 15460
SEMI(KK) = SWTCH: GOSUB 15460
IF MX3FLG$ = "Y" THEN SEMI(KK) = SWTCH: GOTO 19400
IF SWTCH > 0 GOTO 19670
IF SASN$ = "N" GOTO 19400
SWTCH = 0: GOTO 19740
SK = S#; SWTCH = 0
GOSUB 17700
:REM ** TRY EXCHANGING **
S# = SK
GET #2, S#; GET #3, S#
CIF
LO3GFLG$
'"Y"
THEN GOTO 19770

PRINT #1, "ASSIGNING ";SNAME$; ":":SI:" TO SEMINAR ";
PRINT #1, TSEM;
IF TRNS = 0 THEN GOTO 19800
PRINT " ASSIGNING ";SNAME$; ":":SI:" TO SEMINAR ";
PRINT TSEM;
GOSUB 13440: GOSUB 16080: GOSUB 17390 :REM ** UPDATE RECORDS **
NEXT SSI
RETURN
REM **
REM ******************************************************
REM **
REM ** SUBROUTINE TO RANDOMLY ALLOCATE CHARACTERISTICS TO SQUADRONS **
REM **
REM **
PRINT " ALLOCATING SLOTS TO SQUADRONS ";
GOSUB 14870
ADJ = 0
FOR I = 3 TO 27
PRINT ".";
IF STAT(1,0,0) = 0 THEN GOTO 20070
TA = STAT(1,0,0) / 4 :REM ** TEMPOARY ALLOCATION **
IF TA < 1 THEN TA = 0; GOTO 20010
FOR J = 1 TO 4
ALOT(I,J,0) = INT(TA) :REM ** GIVE SQUADRON FAIR SHARE **
NEXT J
IF STAT(1,0,0) - (4 * INT(TA)) = 0 THEN GOTO 20070
FOR L = 1 TO STAT(1,0,0) - (4 * INT(TA)) :REM ** GIVE REST **
ALOT(I,ADJ,0) = ALOT(I,ADJ,0) + 1
ADJ = ADJ + 1
IF ADJ > 4 THEN ADJ = 1
NEXT L
NEXT I
PRINT
RETURN
REM **
REM ******************************************************
REM **
REM ** SUBROUTINE TO OUTPUT RESULTS TO SMART IN ASCII **
CLS: LOCATE 2,25: PRINT "MIXER OUTPUTTING TO SMSS": PRINT: PRINT
STDNT = STAT(O0,,)
OPEN "O", #1, "STUOUT.ASC"
FOR KO = 1 TO STDNT
FOR SSI = 1 TO STDNT
IF SSI/10 >= KO THEN PRINT ".":; KO = KO + 1
20230 S1 = SSI
20240 GET #2.SI: GET #3.SI
20250 OTREC$ = CHR$(34)+SNAME$+CHR$(34)+","
20260 OTREC$ = OTREC$+CHR$(34)+STN$+CHR$(34)+""
20270 OTREC$ = OTREC$+CHR$(34)+ASQ$+CHR$(34)+""
20280 OTREC$ = OTREC$+CHR$(34)+MIXX$+CHR$(34)+""
20290 OTREC$ = OTREC$+CHR$(34)+PRIORS$+CHR$(34)+""
20300 OTREC$ = OTREC$+CHR$(34)+SLX$+CHR$(34)+""
20310 OTREC$ = OTREC$+CHR$(34)+ASLX$+CHR$(34)+""
20320 PRINT #1,OTREC$
20330 NEXT SSI
20340 PRINT #1,CHR$(26)
20350 CLOSE: RETURN
20360 REM
20370 REM **
20380 REM ***********************************************
Appendix Four

SMSS.BAT
Program Listing

SMSS.BAT is an MS-DOS batch file. When the user enters "smss" to start the system, this file of MS-DOS commands is called. Its major function is to pass control between SMART and the mixer program.

SMSS.BAT

echo off
echo Calling SMART--Standby Please
rem SMSS.BAT--this is the DOS batch file used to call SMSS and control entry and exit between SMART and the BASIC program, MIXER
rem
rem After operator enters "smss" from dos, entry is here
rem First call the SMART datamanager and project file smss
smart d -psmss
:smloop
rem Return from SMART either because the user hit Esc at the main menu
rem or because the user has chosen to call the mixer.
rem Existence of file env.asc will be used to test for which.
if not exist env.asc goto smexit
echo call mixer
mixer
rem do test for error exit
if exist error.sms goto smerr
echo Calling SMART--Standby
smart d -psmss
rem return could be from exit or mixer call. go back and check
goto smloop
:smerr
echo error exit from mixer
:smexit
echo exit from smss
END
DATE
FILMED
6-88
DTIC