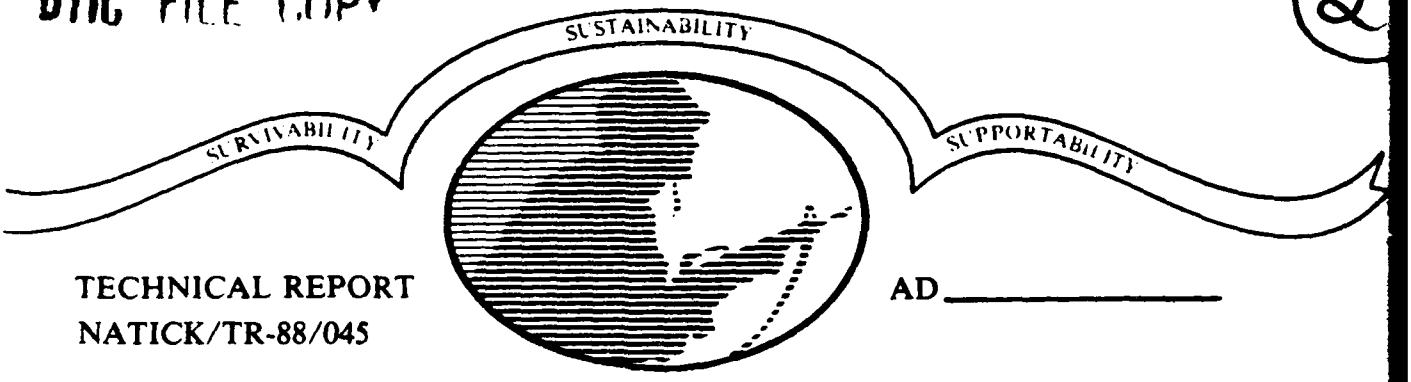
(2)



TECHNICAL REPORT
NATICK/TR-88/045

AD_____

COMPUTER SOFTWARE USED IN U.S. ARMY ANTHROPOMETRIC SURVEY 1987-1988

BY

THOMAS D. CHURCHILL
BRUCE BRADTMILLER
CLAIRE C. GORDON*

ANTHROPOLOGY RESEARCH PROJECT, INC.
YELLOW SPRINGS, OHIO 45387

JUNE 1988
FINAL REPORT FOR THE PERIOD
SEPTEMBER 1986 TO MARCH 1988

Prepared For

UNITED STATES ARMY NATICK
RESEARCH, DEVELOPMENT AND ENGINEERING CENTER
NATICK, MASSACHUSETTS 01760-5000

*SCIENCE AND ADVANCED TECHNOLOGY DIRECTORATE

DISCLAIMERS

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

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) NATICK/TR-88/045
6a. NAME OF PERFORMING ORGANIZATION Anthropology Research Project, Inc. — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U.S. Army Natick Research, Development and Engineering Center
6c. ADDRESS (City, State, and ZIP Code) 503 Xenia Avenue Yellow Springs, Ohio 45387	7b. ADDRESS (City, State, and ZIP Code) Natick, Massachusetts 01760-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable) STRNC-YBF	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER Contract No. DAAK60-86-C-0128
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 728012-12 — PROJECT NO. OMA — TASK NO. — WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) COMPUTER SOFTWARE USED IN THE U.S. ARMY ANTHROPOMETRIC SURVEY 1987-1988	
12. PERSONAL AUTHOR(S) Thomas D. Churchill, Bruce Bradtmiller and Claire C. Gordon	
13a. TYPE OF REPORT Final — 13b. TIME COVERED FROM Sept 86 TO Mar 88 — 14. DATE OF REPORT (Year, Month, Day) 1988, June, 30 — 15. PAGE COUNT 117	
16. SUPPLEMENTARY NOTATION On this contract, Dr. Gordon, Chief of the Anthropology Section, Human Factors Branch, performed the dual functions of Project Officer and co-author.	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	HARDWARE, ANTHROPOMETRY, ~~SURVEY~~ SOFTWARE, EDITING, MEASUREMENT ERROR.
			(Objective equipment)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes the hardware and software used in the 1987-1988 anthropometric survey of the U.S. Army. It includes a background section outlining how anthropometric data can be edited mathematically, and how such editing has been done in surveys of the past. It then describes the data entry and editing software which was created especially for the 1987-1988 survey, and how that software is based on historically proven techniques. Programs which randomly select subjects for measurement from among those who are screened and which calculate interobserver measurement error are also described. The appendices include a self-contained hardware/software user's guide, and complete program listings for all software used in the survey.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Claire C. Gordon	22b. TELEPHONE (Include Area Code) 617-651-5429 — 22c. OFFICE SYMBOL STRNC-YBF

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

This report was written to fulfill requirements of contract DAAK60-86-C-0128 with the Science and Advanced Technology Directorate, U.S. Army Natick Research, Development and Engineering Center, Natick, Massachusetts. The contract monitor was Dr. Claire C. Gordon, Chief of the Anthropology Section, Human Factors Branch.

The majority of the survey software was written by Thomas D. Churchill. The program, SURVEY38, was written by Paul F. Abendroth, to whom the authors are greatly indebted. The program, DELTAS, was written by Bruce Bradtmiller who also wrote the text of the report.

The authors are grateful for the advice and editorial improvements contributed by Ilse Tebbetts, John T. McConville, Carolyn Bensel, Lawrence E. Symington, Abner S. Salant and Edna Albert. Additionally, we appreciate the many suggestions for the program improvements from individuals on the survey team who, in a real sense, became "program testers extraordinaire" while at the same time learning other aspects of their jobs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

COMPUTER SOFTWARE USED IN THE U.S. ARMY ANTHROPOMETRIC SURVEY 1987-1988

INTRODUCTION

The purpose of an anthropometric survey is to produce data which are ultimately used in the design of uniforms, equipment, workspaces, and computer models of humans. Because large numbers of subjects are measured for large numbers of dimensions, computers have been employed for many years in the analysis of survey data. In the 1987-1988 U.S. Army Anthropometric Survey computers were used, for the first time, for data collection.

In the past, when the only computers available were room-sized mainframe machines, the data were collected in the field, hand-written on data forms, and only later punched onto cards or disk files. After the data were punched or entered on computer files, they were edited to identify values which might be in error. In that system, there were essentially three sources of error: (1) anthropometrist error (i.e., mismeasuring or calling out the incorrect value); (2) recorder error (i.e., writing down a value different from that called out); and (3) punching error (i.e., punching a different value from that written on the data form). When a value was identified, through editing, as a possible error, the data analysis team had to decide whether to modify the value, discard it, or retain it as a likely correct value.

In the 1980s, when computers are more likely to be lap-sized than room-sized, they can be taken into the field to record the data points as they are called out by the anthropometrist, or even to record the points directly from the measuring instruments. Such a system has a number of advantages. First, data from the survey are ready for analysis much sooner after the last subject is measured than had previously been the case, because it is not necessary to wait for punching and verification. Second, data on diskettes take up much less space than the same quantity of data on paper data forms. Further, when the software used for data collection includes data editing features, the advantages over the previous system multiply. If a suspicious data value is identified while the subject is still available, then that value can be checked against the subject to determine whether it was indeed incorrect or whether the subject is simply anomalous in that dimension. This feature dramatically increases confidence in the data because it reduces the number of occasions in which data editors must make judgments about the disposition of an aberrant value. Software with just these features was developed for the U.S. Army Anthropometric Survey of 1987-1988. This report describes that software and its operation, as well as other associated software developed for the survey.

The report is organized into three sections. The first is a background and general introduction to the editing of anthropometric data. Outlined in this section are the statistical methods appropriate to editing this particular kind of data. The second section describes how the software was designed to perform an edit function and record the data onto diskettes. Other features of the software are also described in the second section. The third section of the report discusses two other programs used in the survey. These are the subject selection program that was used to randomly select individuals who were fully measured from among those who were screened only

and the program that calculated interobserver differences in measurement for those subjects who were remeasured at one or more stations.

A brief mention should be made about what this report does not contain. The 1987-1988 survey made use of an automated device for collecting three-dimensional data of the head and face. Data from this device were input, via cable, directly into a personal computer. While software for this computer and this device were created expressly for the survey, this software is not described in this report. The reason for this is that the software for the automated head measuring device is an integral part of that system and cannot be adequately understood without a simultaneous understanding of the associated hardware. The software and the hardware are completely described in a technical report on the automated headboard device.[1] The interested reader is referred to that document for further information.

BACKGROUND

The purpose of this section is to provide a short discussion of how and why this kind of data editing works on anthropometric data, how such editing has been done in the past, and how these features were combined to yield the specific editing scheme used in the 1987-1988 Army survey software.

Characteristics of Anthropometric Data

Students in any introductory statistics course soon learn about the normal bell-shaped curve of frequency distributions. This is taught early because many of the more advanced statistical techniques assume a normally distributed data set. Anthropometric data, when properly collected from a large sample, unlike some other kinds of natural data, almost always approach a normal distribution. The only real exceptions are skinfold measures and to a certain extent weight. A typical normal curve showing the stature distribution from the 1970 survey of Army aviators[2] is shown in Figure 1.

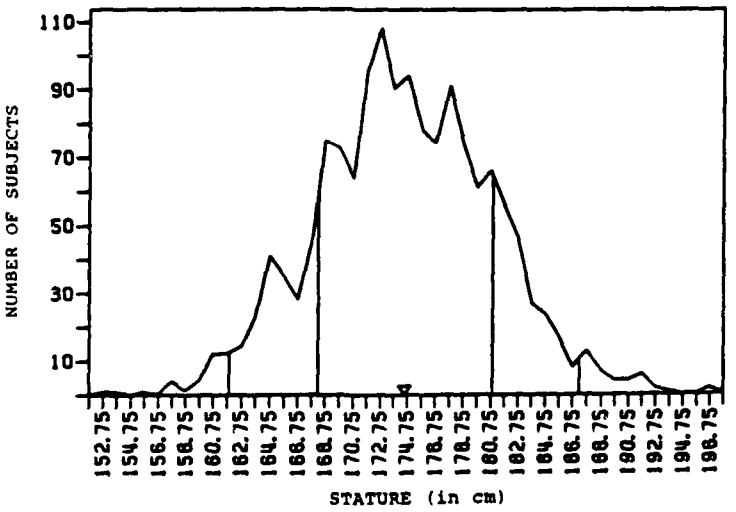


Figure 1. Distribution of stature from the 1970 Army aviator survey. Source: Churchill, et al. (1971).

While the characteristic shape of the curve has important functional implications for product design and sizing, its importance here is that it can be used in a first-level, very coarse data editing scheme in which values at the extremes of the distribution are flagged as possible errors. By flagging only the very large and very small values, relatively few correct values are falsely challenged, and the efficiency of the editing remains fairly high.

Just as anthropometric data have useful univariate characteristics in most cases, so do they often have useful multivariate features, in that many dimensions are related anatomically, hence mathematically, to each other. It is obvious from casual observation that those males who have large neck circumferences usually also have large chest circumferences. Similarly, those with small statures often also have small cervicale heights. If such individuals were plotted on a graph, the points generally tend to form a linear cluster, as can be seen in Figure 2.

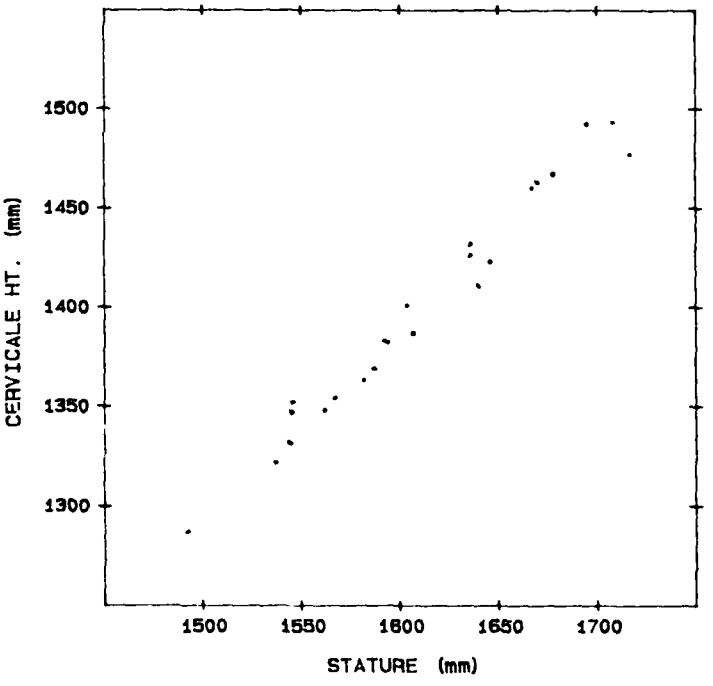


Figure 2. Hypothetical plot of Stature vs. Cervicale Height.

The linear cluster is a visual representation of what is essentially a statistical relationship between the two dimensions. The statistical relationship can be described by a line drawn through those points in such a way that the summed distances between the points and the line are minimized. This line is called a regression line, and is pictured in Figure 3.

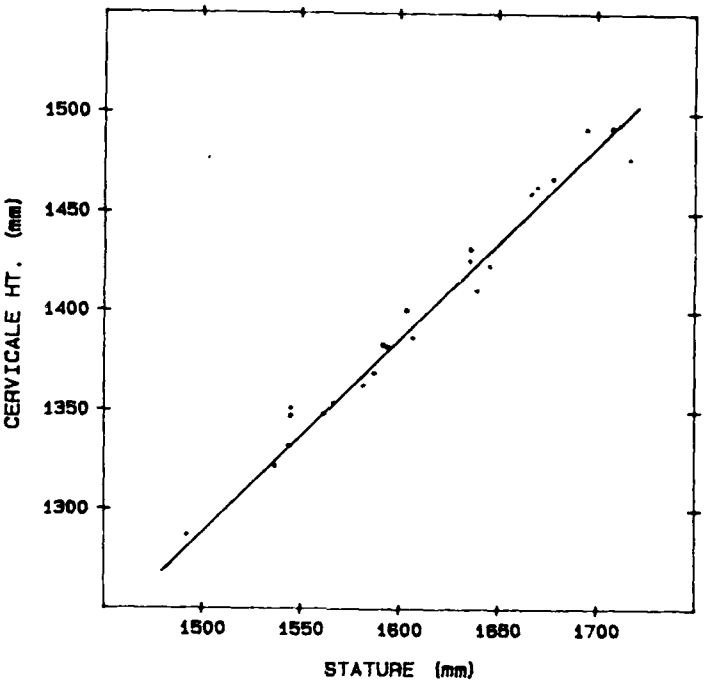


Figure 3. Regression line using data from Figure 2.

A regression line is expressed mathematically by an equation, of the general form y=a + bx, in which **y** is the dimension on the vertical axis (in Figure 3, cervicale height) and **x** is the dimension on the horizontal axis (in Figure 3, stature). The dependent variable is said to be **y** and the independent variable, **x**. In the equation **a** refers to the intercept on the vertical axis and **b** is the slope of the line. For any given population of x's and y's, there is only one regression line, represented by a single equation.

Using the regression equation, one can estimate, or predict, the value of the dependent variable by inserting a single value for the independent variable. Thus in the present case:

$$y = a + b * x$$

$$\text{cervicale height} = (-117.03) + 0.94 * (\text{stature})$$

If a person's stature is 1650 millimeters, that value can be substituted in the equation, and that person's cervicale height can be estimated as 1434 mm.

A regression estimate has a standard error which is, in effect, the standard deviation of the predicted value. The larger the standard error, the more variability there is in the dependent variable with respect to the independent variable. From a practical point of view, the larger the standard error, the less confidence one can place in the predicted value. The standard error is therefore useful in assessing the importance of a predicted value which might be used in data editing.

While a simple regression equation in which a single variable (stature) acts as a predictor is illustrated above, it is statistically possible to have two or more predictors. Such equations are called multiple regression equations. A disadvantage of such equations is that they are difficult to visualize and graph on paper. A significant advantage, however, is that the ability of the equation to predict a given dependent variable is often enhanced by the addition of other independent variables.

The close mathematical relationship which many anthropometric dimensions have with each other, and which are so well described by regression equations, make regression equations a very useful tool in a second level of data editing. This second level is a considerably finer sieve than the first level in which only extreme values are questioned. In using regression equations for data editing, a value for each dimension can be predicted using measured values of other dimensions as in the above example in which cervicale height was predicted on the basis of stature. Comparing the predicted value of cervicale height with the measured value of cervicale height is a very effective way of testing the validity of the measured data point. A different equation, using other variables, can be constructed to check the value of the measured stature dimension.

In short, the fact that anthropometric data tend to be normally distributed on a statistical curve, with very few values at the extreme ends of the distribution, makes it possible to catch some kinds of errors by querying all extreme values obtained for a given dimension. An even finer editing system is made possible because pairs and classes of dimensions are closely related to each other in size and, thus, the size of one dimension can be reliably predicted from the size of another. Measured values are compared with predicted values and significant discrepancies queried for possible error.

Data Editing in Anthropometric Surveys of the Past

Computer aided data editing was first used in U.S. military surveys in the 1950 survey of Air Force men.[3] In that case, various dimensions were subtracted from or added to other dimensions, and the resulting distributions were examined. For example, if eye height subtracted from stature resulted in a difference of zero or less, the values for both stature and eye height were scrutinized. This approach to editing was crude by today's standards and extremely time-consuming.

In 1965, Edmund Churchill, a mathematics professor at Antioch College, Yellow Springs, Ohio, under contract to the U.S. Air Force, developed the two-pronged approach which forms the basis of the editing routines in use today.[4]

The first level of editing is the X-Val (extreme values) computer program. This program lists many univariate statistics for each of the dimensions measured in a given survey, but more important for data editing, it lists the 10 largest and 10 smallest values for each dimension. In past surveys, data editing began after large quantities of data had been obtained. Upon receiving the X-Val printout, the data analyst looked for several things.

After checking the mean, standard deviation, and coefficient of variation, the analyst checked for the skewness and kurtosis of the distribution. These characteristics refer to the overall shape of the "normal" bell curve. They help answer the questions, "Is the bell of the right height?", "Is the bell of the right breadth?", and "Is the bell centered over the mean value?" An oddly shaped distribution is one indication that one or more values at the extreme ends of the distribution are distorting the shape of the whole curve. After examining these measures of the distribution of the whole population for a given variable, the data analyst examined the ten largest and ten smallest values, which were located nearby on the printout. Beside each of the extreme values was the subject number of the individual who had that value. Thus, if an extreme value was by its magnitude clearly an error, and if the nature of the error was obvious (e.g. transposed figures), it was a simple matter to locate the individual record (at that time on a punched computer card) and make an appropriate change. In this way, it was possible to make use of the normal distribution characteristic of anthropometric data to edit the data and correct any erroneous points.

Appropriate distributions, however, tell only part of the story when data are edited. It is entirely possible for an individual data point to be in error even if it is not at the end of the distribution. For example, a value for a stature of 1687 could be and sometimes undoubtedly was misentered as 1867. A digit transposition is a common error, easily made, but in this case an error in which neither the correct value nor the erroneous value would be at the end of the distribution where it would be flagged by the X-Val editing system. Yet the error is a substantial 180 mm. Numerous errors of this kind could be embedded in any group of measured values.

It was recognition of this problem that led Churchill to develop the second phase of editing for anthropometric data -- a program called EDIT based on the multivariate characteristics of anthropometric data. Because regression equations can be used to predict values for the dependent variable, they can also be used in editing to predict a measured value. The predicted value for a given dimension for a given individual can be compared to the actual measured value for that dimension for that individual, as a way of gauging whether the measured value is likely to be correct. Any values which seem seriously discrepant are flagged and can be examined more closely.

The EDIT program was used in the following way. After initial editing with the X-Val program was completed, a correlation matrix for the data set was produced. Using the correlation matrix, and a knowledge of the anatomical and statistical relationship of anthropometric dimensions to each other, a series of editing combinations was developed. These editing combinations were triplets of dimensions, each of which was a fairly reliable predictor of the other two. An example of such an editing combination is stature, suprasternale height, and acromion height. Editing combinations were developed so that each dimension was involved in at least one triplet. Some dimensions were involved in many triplets. The editing combinations and a check value, which is a multiple of the standard error of the predicted value, were entered into the program. The program then calculated regression estimates for each individual value in the data set. For example, as above, regression equations were developed so that stature and suprasternale height were used to predict acromion height, acromion height and suprasternale height were used to predict stature, and stature and acromion height were used to predict suprasternale

height. For each individual in the data set, the predicted value of stature was subtracted from the measured value of stature. If the difference between the two values was greater than the check value multiplied by the standard error of the regression estimate, then the dimension name, subject number, measured value and predicted value were all listed on the printout. This procedure was followed for each dimension for each subject. An example of the output from an early edit of the 1970 Army Aviator survey[2] is shown in Figure 4.

It should be emphasized that if stature and suprasternale height are used to predict acromion height, and the measured value is flagged as a potential error, it does not mean that acromion height is necessarily the value in error since an erroneous value in either stature or suprasternale height would also produce a predicted value of acromion height which was different from the measured value. Thus the editing is always done in triplets so that by examining the pattern of predicted versus measured values in all three dimensions of the triplet, it is possible to identify which of the three measured values is the likely error.

When editing was done in the past, the EDIT program was used several times, decreasing the check value each time. The initial level for the check value is a matter of some judgment, but often it was set as high as 10. Depending on the number of questionable values identified on the earlier runs, the check value would be decreased on each successive run. Generally, a check value of less than 3.0 was not used, because a value lower than that often begins to flag values that are correct and are simply examples of human variability.

Both the X-Val and the EDIT programs were, and still are, excellent means for identifying values that are out of the ordinary, from the perspective of the population and the individual, respectively. When aberrant values were identified, the data analyst decided the fate of each questionable data point on a case-by-case basis. In the case of a clear transposition of digits, a correction could be made with a relatively clear conscience. Similarly a correction could be made in the case of a missing initial digit (e.g., stature recorded as 785, instead of 1785). Yet problematical values were often flagged for which no clear solution was apparent. In such cases, the analyst had to decide whether to declare the value missing, or to substitute the value predicted from the regression equation, or finally, to leave the questionable value in place. These options are always unsettling to the data analyst, but in the context of data editing long after the survey had ended, there was really no other choice.

The imprecision inherent in making subjective decisions about questionable values spurred the development of the software used in the 1987-1988 U.S. Army survey and described in this report. The goal was to perform the initial data editing at the same time the data were entered, thereby dealing with questionable values when the subject was still present. If a value appeared either out of range or inappropriate when compared to other dimensions for that individual, then the dimension could be remeasured on the same subject, and the questionable value could be either confirmed or replaced with a correct value.

SUBJECT NUMBER 901

	①	②	③	④			①	②	③	④			①	②	③	④
No. 35 Chest Circ	1068.	931.	4.3	***	33 Neck Circ		365.	380.	-1.1	***	34 Shoulder Circ		1108.	1212.	-3.6	***
No. 34 Shoulder Circ	1108.	1292.	-4.2	***	35 Chest Circ		1068.	1010.	2.2	***	36 Waist Circ		1030.	950.	1.8	***
No. 35 Waist Front	441.	379.	3.5	***	56 Waist Back		453.	478.	-1.1	***	13 Trunk Height		559.	608.	-2.4	***

SUBJECT NUMBER 924

No. 37 Hip Circ	1008.	1113.	-3.6	***	35 Chest Circ		1155.	1085.	2.8	***	36 Waist Circ		1059.	981.	2.3	***
No. 36 Waist Circ	1059	905	3.9	***	37 Hip Circ		1008.	1039.	-1.5	***	38 Upper Thigh Circ		603.	620.	-0.9	***

1 = the variables in the editing triplet
2 = measured value
3 = predicted value
4 = $\frac{\text{(measured - predicted)}}{\text{standard error}}$

Figure 4. Sample output from EDIT of 1970 Army aviator data.
Source: Churchill, et al. (1971).

DATA ENTRY AND EDITING

This section describes how the survey software worked. It is not a user's manual, which appears in Appendix A, nor is it a listing of the program source code, which is Appendix B. Instead, the section describes, in a textual form, how the programs asked for data, how the data were edited, and the disposition of the data. All on-line editing was done separately for males and females.

The standard dimensions and the automated headboard dimension were divided into eight measuring stations, which the subjects visited sequentially. The rationale and methods for this division are presented in Clauser et al. (1988).[5]

Measuring Stations #1 - #6

The software for Stations #1 through #6 was essentially identical. The only substantive differences between the stations were that different dimensions and different numbers of dimensions were measured at each station. (A unique feature of Station #2 is discussed later in this section.) The program operating at these stations was called INED (Data INput and EDiting).

As it was used in the field, INED was invoked by an AUTOEXEC.BAT file on the boot diskette. This means that the diskette could be placed in the computer before it was turned on, and the program came up automatically on the screen. It was possible to invoke the program by the command INED if the AUTOEXEC.BAT version was not available.

The software system was set up as a two-disk system. One disk contained the program and several files necessary for its function. The program/station disk remained in the station computer whenever the program was running. The second disk was unique to each subject, and was carried by the subject from one station to another.

The structure of the program was essentially interactive in that the program posed a series of questions and the user responded. Because the survey was set up to measure all males or all females in half-day increments, the sex of the subjects was sought before the first subject of a session was measured. Because each station had two measurers, either of whom could measure a given subject, the measurer's name was asked for before each subject.

When a subject first appeared, his or her diskette was inserted into the drive and the program asked for a subject number. The subject's number was stamped onto his/her original biographical data form and was written on the subject diskette, the diskette envelope, and the computer data sheet which was carried from station to station. Since it was critical to keep the subject's form with the diskette until the data from the diskette were entered, the subject number was also written on a file on the disk. Thus when the program asked for the subject number, it checked the number in the disk file to make certain that the diskette still accompanied the proper form.

After the subject number was checked, the program asked for the measurement of the first dimension. At each station, the names of each dimension were stored on the station/program disk, and each dimension was called out by name, so it was not necessary to remember the order of the measurement. When a measurement was entered, it was displayed on the screen and the next measurement was requested.

The first phase of the on-line data editing began at this point. This phase was analogous to examining data on the X-Val program in previous surveys. In a step that was blind to the user, after a measurement was entered, but before the next measurement was displayed, the value was compared with the largest previously-measured value for that dimension and with the smallest previously-measured value for that dimension. These largest and smallest values were from the population being measured (i.e., the U.S. Army survey subjects to date). If the measurement was larger or smaller than the largest or smallest values to date, the program moved to a subroutine which challenged the value. The screen notified the user that the value was out of range, and gave the user the chance to either change the value or to declare that the subject was unusually large or unusually small for this dimension. If the value was changed because the user realized that an entry error had been made, then the new value was checked before going on to the next dimension. If the user specified that the subject was large or small, the entered value was checked to ensure that it was correspondingly large or small before going on to the next dimension.

This first phase of on-line editing occurred for each variable, just as it was entered. As with the X-Val when used in post-survey editing, INED screened out values that were gross entry errors, based on the population distribution. When a large or small value was a "true" value and confirmed by the regression check used in the next phase of editing, then that value was substituted for the previous large or small limit. (These values were maintained in a file on the station disk.) If an individual had a large or small value which was beyond the limits, and the regression check found it aberrant after three measures, then that value was not used to update the limits. Using limits set by individuals of unusual shape or proportion was thought to defeat the purpose of limit checking during data entry.

A second phase of on-line editing occurred when the last dimension of a given subject at a given station was entered. Each station/program disk contained a file with regression equations for that station. The regression equation predicting an individual variable had, as its two independent variables, the best two predictors chosen from among the dimensions at that station. These independent variables had been selected after computing the multiple correlation coefficient (a measure of how well two or more variables predict another variable) for every possible combination of variables at that station. The pair with the highest multiple correlation coefficient had been chosen as the two predictor variables for that dimension. These equations were calculated on all subjects measured in the Army survey. At the beginning of the survey the equations were calculated and updated daily until a sufficient sample size was achieved. After measuring had been completed at two Army posts, equations were updated at each new post to include the cumulative data through the previous post.

After the last dimension value for a subject was entered by the operator, and the operator had a chance to review all the entered values, the regression phase of editing began. Using the regression equations contained in the file on the program disk, a predicted value for each dimension for that subject was calculated. The predicted value was subtracted from the measured value, just as in the EDIT program used in post-survey editing. The differences were computed in this manner for all the measured values. If any of the differences was greater than three times the standard error, the program instructed the operator to remeasure one or more dimensions. As in the case with the EDIT program, the first predicted value to be flagged was not necessarily the one which had been misentered; either of the two dimensions used in the prediction was a possible culprit as well.

The program decided which dimension to remeasure first by arranging, in ascending order, all of the differences divided by the standard errors. The dimension with the largest discrepancy between its measured and predicted value was remeasured first. If there was a significant change* between the originally measured value and the remeasured value, then all the regression estimates were recalculated. If, on recalculation, none of the differences was greater than three times the standard error, then the program moved beyond the data editing phase, If, on recalculation, there were still one or more differences greater than three times the standard error, the program asked for a remeasure, again of the dimension with the highest difference divided by the standard error. If that dimension had already been remeasured, the second highest difference divided by the standard error was remeasured. The program continued down the list of those dimensions where the differences divided by the standard error were decreasing. This continued for all differences until the differences were less than two times the standard error.

A word of explanation is necessary to account for the apparent discrepancy between starting the remeasuring sequence at three times the standard error, but, in fact, remeasuring those differences down to two times the standard error. This method was chosen because, although a predicted value may be well above three standard errors different from the measured value, the fault may be in the mismeasurement of one of the independent (predictor) variables. Where this is the case, the predicted value of that independent variable also shows a difference from its measured value. That difference, however, might not be as large, relative to its standard error. Therefore it is necessary to go somewhat below the initial check value if any remeasuring is necessary.

During the course of remeasuring, if differences between predicted and measured values that exceeded the limits persisted after all differences divided by standard errors exceeding two had been remeasured, the remeasuring sequence was begun again, following the same process as before. The rationale behind this was that, while a measurer might measure a dimension incorrectly a

---

* A significant change, in this context, is defined as the lesser of either one-half the standard error, or one-half the allowable measurer error. Allowable measurer errors are those values which define "acceptable" or "not acceptable" levels of difference between two measurers each measuring the same subject on the same day. A full discussion of these allowable errors can be found in the Measurer's Handbook.[5] The actual values can be found in Appendix C.

first time, and possibly even a second time, it was extremely unlikely that a measurer would measure a dimension incorrectly three times in a row. This was especially true because the measurer's awareness of his/her actions was increased by virtue of having to remeasure. No dimension was measured more than three times on any subject. At this point it was assumed that the reason for the discrepancy, if it persisted, lay elsewhere.

Occasionally it happens that a subject is proportioned differently from many people. Such an individual would be located at some distance from a regression line (see Figure 3). In these cases one or more of that individual's measurements would be flagged for remeasure, because they would appear to the software as possible measurement errors. When this occurs, the subject will likely have been measured once and remeasured twice for some dimensions. Because the measured values are not errors, even after the third measurement, the editing procedure would continue to identify values which it regarded as suspicious. In such a case, the operator in the Army survey was instructed to enter an explanation of why the data appeared as unusual. This explanation was written to the subject's disk (although not printed out on the hard copy of the data sheet), and was of help to the data analysts who examined the data when all the subjects had been measured.

After all measuring and remeasuring had been completed, the program wrote the data to the subject disk and to the station disk. The manner in which the data were written to the subject disk depended, to a large extent, on whether or not the subject was remeasured, and how often. On the subject disk, the data were written in rows and columns. There was one column for each dimension at the particular station, and there were four rows. The fourth row always contained the difference between the predicted and measured value divided by the standard error for the dimension. If a subject was not remeasured, all the measurements were placed in row one, and rows two and three contained zeros.

In the case where a subject was remeasured, values were placed in rows two and three according to the following guidelines: (1) if the remeasured value was significantly different (as defined above) from the original value, the new value was placed in row one, and the original value was moved to row two; (2) if the remeasured value was not significantly different from the original value, the original value remained in row one and the new value was placed in row two; (3) this proceeded until all the dimensions which needed to be remeasured had been remeasured one time. If these remeasurements resolved all the discrepancies, the data were written to the disk in that form, and row three remained with zeros. If discrepancies still remained after the first remeasure, then these additional steps took place: (4) all non-zero values in row two were moved to row three; (5) the second remeasuring sequence began, and values were placed either into row one or the now-empty row two, according to the rules in (1) and (2) above.

The purpose for moving the data from one row to another was to keep the most likely correct value in row one. In those cases where a subject is aberrant, and a second remeasure has not resolved the problem, the program has no basis for selecting the "most likely correct" value. In such cases, the value in row one is the mean of all three measured values. The value in row

two is the lowest of the three values and that in row three is the highest. Thus when this value is flagged in post-survey editing, the data analyst will have a complete measurement history on the dimension for that subject and can make decisions accordingly.

Since the most likely correct value is in column one, the writing of data to the station disk was much simplified. After data were written to the subject disk as described above, the data from row one were written to the station disk and printed out. The station disk served as a backup in case of loss or damage to the subject disk, but the subject disk, with its complete data record, was the primary source of data for compilation of all the survey data.

The data collection and disposition described above occurred identically at stations #1 through #6. There was a minor addition at Station #2, which contained three special dimensions: thumbtip reach, wrist-wall length, and wrist-wall length, extended. These dimensions have traditionally been measured several times because they involve subject motivation and considerable subject participation in establishing subject position. Here they were measured three times, and the software for station two was modified to accommodate this procedure. The program selected which of the three values to record in the following way: (1) it calculated the difference between the largest value and the middle value, and then between the smallest value and the middle value; (2) those differences were compared with the allowable error for that dimension; (3) if both differences were smaller than the allowable error, the median value of the three was retained and recorded; (4) if both differences were larger than the allowable error, the dimension was flagged and all three variables were remeasured three times each; (5) if only one value (either the largest or the smallest) was distant from the median by more than the allowable error, then it was dropped, and the mean of the remaining two values was recorded.

Station #7

Station #7 housed the automated headboard device for the collection of three-dimensional data on the head and face. As noted, the software for the headboard was intimately connected with the operation of the device, and it is described fully in Annis and Gordon, 1988.[1]

Station #8, In-processing and Out-processing

The software written for this station (program INOUT8 listed in Appendix B) had several functions. For that reason, it is menu driven, and the operator always had a choice of which function to perform next. The Station #8 portion of the software was identical to that at Stations #1 through #6. Three foot dimensions were measured at Station #8.

The in-processing operations, which took place physically at the same site were very different in nature. The in-processing function created, for the first time, the subject disk that was subsequently carried to all the other stations. At the in-processing station, the operator took from the subject his/her biographical data questionnaire (see Appendix D), and copied the subject number from the upper right hand corner onto a blank formatted

disk, onto the diskette sleeve, and finally onto a blank data sheet. The diskette in its sleeve and the data sheet were carried from station to station, while the completed biographical data questionnaire remained at the in-processing station. The in-processing software asked for the subject number and wrote it onto a file on the diskette. That file also contained the sex of the subject and the subject's stature and weight copied from the biographical data questionnaire. Stature and weight were used only to separate scrambled subjects in the case where there was a mixup about subject numbers. Two individuals might, in an unlikely event, be assigned the same subject number, but it was deemed very unlikely indeed that these two persons would also have exactly the same stature and weight.

The entry of all the information from the biographical data form was also performed at this station. This menu item was interactive, as at the other stations, and followed exactly the order of questions on the biographical survey form. These data were entered into a separate file on the subject's diskette.

The other major function of the software at this station was the out-processing of subjects. In this program feature, the subject's diskette was entered, and each data point checked to make certain that it was readable. If there was a problem with reading measurements from a particular station, the subject was either sent back to the station for remeasurement, or the values were entered from the hard copy data sheet. In addition, the data from the subject number file and the data from row one were written to a file on the station data disk. In this case, each subject's data were contained in a separate file. Because this took up more space on the diskette than one large file with all the data, the station diskette was changed weekly. It should be noted again that these cumulative records served only as a backup in the case of loss of the subject diskettes, which contained all the information about each subject, and remained as the primary data source in compiling survey data as they come in from the field.

Summary

The data input and editing software for this survey functioned on the basis of a two-diskette system. One diskette stayed with the station and kept a cumulative record of the subjects processed. The second diskette was unique to the subject and was carried by the subject from station to station.

As data were entered by the measurer and recorder, they were checked, first for their position in the population of all individuals measured to date, and second for their position in relation to other measured values for that subject. When, by means of these checks, values appeared to be aberrant, the operator was asked to remeasure the subject. This ensured that questionable data values were checked on the subject while he/she was still at the measuring location.

The advantages of this computerized system over earlier data collection methods are: (1) increased speed of data entry; (2) reduced volume of physical data records; (3) increased confidence in the resulting data set.

OTHER SURVEY SOFTWARE

Subject Selection Software

One of the features that distinguished the 1987-1988 survey of Army personnel from its predecessors was the use of a stratified random sample, with preset sampling cells describing specific race and age groups. Thus, for the first time, this survey required the selection of subjects from a larger group of potential subjects who presented themselves to be measured. Although it would have been possible to eventually fill each sampling cell by measuring every soldier available, that was considered to be an extremely inefficient and expensive approach to data acquisition. The challenge therefore, was to find a way to screen a large number of individuals, select those who were most needed to fill the sampling cells, and do so in a way that would not bias the sample.

The entire sampling strategy, the rationale behind it, and the mechanics of using it are described in detail in Gordon and Bradtmiller, 1985.[6] Described here is the software that was used to assist in the implementation of that sampling strategy. In general, one or more military units were screened at once. The subject selection team knew in advance the age and racial composition of the units to be screened. Considering that composition in light of the sampling needs for the survey as a whole, the subject selection team made determinations, before screening began, of how many in each sampling cell were to be selected from the companies at hand. The purpose of this program was to use that information to make the appropriate number of random selections. The assumption was made that subjects would present themselves to be screened randomly with respect to age and race. This was a reasonable assumption because, although the subjects were told that they would be selected according to race and age, they did not know how age was divided into categories, and they did not know which sampling fractions were to be used. Thus, any attempt by an individual subject to determine whether or not he would be selected by positioning himself in a particular place in the line was very unlikely to be successful.

The subject selection procedure (named "Survey38" and listed in Appendix B) was written in BASIC and was designed to operate on a Radio Shack TRS-80 Model 100, a small laptop portable computer which can run on battery or AC power. This kind of portability was required because often the subject screening took place out of doors, on training fields, and in gymnasia where electrical outlets were not readily accessible. The menu-driven program was quite easy to operate.

The menu allowed the user to choose between entering parameters, actually selecting the sample, and finally, displaying various values associated with the subject selection operation. When entering parameters, the user was asked how many would be selected for each age-race sampling cell. If the user selected more than half of those screened in a category, the program asked what proportion from that category should be rejected. Specifically the form of the question was "Reject 1 in ?", and the user filled in the blank. Conversely, if less than half of those screened in that category were to be selected for measurement, the question was of the form "Accept 1 in ?". This procedure was repeated for each of the age-race sampling cells.

After each of the parameters had been entered, the user was returned to the menu and could choose the subject selection routine. Under this item, the user was asked to enter the age of the subject, and then the race of the subject. These were both obtained from the subject's biographical data form. After both items were entered, the program determined whether the subject was selected, and gave an audible and on-screen signal to the operator indicating whether the subject had been chosen for complete measurement. The actual program operation was a matter of counting the number of individuals in each cell. For example, if a Hispanic subject between the ages of 21 and 24 presented himself, and the entered parameters directed the acceptance of one in two individuals in that age-race sampling cell, then that subject would be selected if he were the first to have been screened in that category, and rejected if he were the second. The third subject would have been selected, and so on. The program kept track of the counts in each age-race category, and either selected or rejected according to the parameters entered. Unless all individuals in a category were to be rejected, the first subject in that category was accepted, and others accepted or rejected according to the parameters of each cell.

The user entered, during the input of parameters, the total number of individuals to be selected during that session. The program kept track, across all age-race categories, of how many subjects had been selected and notified the user when the total number of subjects had been selected. This was an advantage when the user had entered parameters assuming a certain distribution of race and age in the companies to be screened, when, in fact, the actual distribution of personnel present was different. It allowed the operator to stop screening subjects if the maximum for that session had been reached.

The third menu item allowed various values to be displayed. A secondary menu within this item enabled the user to view the parameters which had been input, the individuals screened in each category, and the individuals selected for measurement in each category. These could be seen for both the day's and the week's total. At the end of each week those files were deleted so that the computer could begin a fresh count the next week. Using standard communication software, these totals were transmitted to other computers which ran standard spreadsheet software and kept track of sample acquisition for the survey as a whole.

Delta Program

One of the unique features of the 1987-1988 survey of Army personnel was that 10 subjects were processed twice each week. These repeat subjects were scheduled so that once each morning, and once each afternoon, a subject was measured by both measurers instead of just by one measurer at that station. The purpose of this was to maintain the accuracy of the data being gathered, and to immediately identify situations in which the two measurers' techniques began subtly to differ from each other. The Delta program is a simple program, written in BASIC, which calculates the differences between two measurements for a given dimension, and prints them out, along with the mean for all the differences, and the allowable error for that dimension (the program listing is in Appendix B).

This program was menu driven and had three main items. The first was to read values from the subject diskettes. Under this item, the operator inserted a subject diskette, entered the subject number, and entered the station at which the subject was remeasured. If the subject was remeasured at more than one station (a rarity), the entry process was repeated for the second and subsequent stations. This menu item could be used on a daily basis, or the diskettes could be accumulated and a whole week's worth done at once. The program operation under this first menu item was simply to open the file containing the first-time measurements, and read them in; open the file containing the second-time measurements, and read them in; subtract one from the other, and write the differences to a file containing deltas.

The second menu item was used at the end of the week to calculate the mean differences and print the results. After selecting this menu item, no further user input was required. The output included the dimension name, each inter-measurer difference for the week, the mean of the differences, and the allowable error for that dimension. Each station was printed on a separate page. A sample output from this program is reproduced as Figure 5.

Dimension	Deltas	N	Mean	AE
Head Circumference	1 0 0 1	4	0.50	5
Bitrag Coronal Arc	0 1 6 1	4	2.00	7
Bitrag Crinion Arc	1 2 1 2	4	1.50	5
Bitrag Frontal Arc	1 4 1 5	4	2.75	5
Bitrag Subnasl Arc	1 0 1 1	4	0.75	6
Bitragion Chin Arc	2 5 0 1	4	2.00	8
Bitrag Submand Arc	1 7 8 1	4	4.25	6
Bizygomatic Brdth	0 0 1 0	4	0.25	2
Head Length	1 3 1 3	4	2.00	2
Head Breadth	0 0 1 1	4	0.50	2
Menton-Sellion L	3 1 3 0	4	1.75	3
Ear Length	1 2 0 1	4	1.00	2
Ear Lgth abve Trag	4 2 0 2	4	2.00	2
Ear Breadth	1 1 0 2	4	1.00	3
Ear Protrusion	1 1 2 0	4	1.00	3
Interpupil Breadth	0 0 1 0	4	0.25	0
Thumb Breadth	1 0 0 0	4	0.25	2
Wrist-Thmbtip Lgth	2 0 2 2	4	1.50	3
Wrst-Ctr of Grip L	2 4 1 1	4	2.00	4
Hand Length	2 0 3 1	4	1.50	3
Wrist-Index Fingr L	1 1 4 0	4	1.50	4
Hand Breadth	1 1 1 0	4	0.75	2
Hand Circumference	1 2 0 2	4	1.25	4

Figure 5. Sample output from the Delta program. This page shows the deltas for Station #4, at a time when four subjects had been remeasured.

The third menu item was simply the deletion of all files so that the next week's data could be accumulated. It, too, required no user input beyond the initial selection of that item.

REFERENCES

1. Annis, James and Claire Gordon. 1988. The Development and Validation of an Automated Headboard Device for the Measurement of Three-dimensional Coordinates of the Head and Face. Technical Report NATICK/TR-88/048, U.S. Army Natick Research, Development and Engineering Center, Natick, Massachusetts.

2. Churchill, Edmund, John T. McConville, Lloyd L. Laubach and Robert M. White. 1971. Anthropometry of U.S. Army Aviators - 1970. Technical Report 72-52-CE (AD 743 528). U.S. Army Laboratories, Natick, Massachusetts.

3. Hertzberg, H.T.E., G.S. Daniels and Edmund Churchill. 1954. Anthropometry of Flying Personnel-1950. Technical Report WADC TR 52-321 (AD 47 953). Wright Air Development Center, Wright-Patterson Air Force Base, Ohio.

4. Kikta, Paul and Thomas Churchill. 1978. Editing Procedure for Anthropometric Data. Technical Report AMRL-TR-78-38 (AD A060 393). Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

5. Clauser, Charles, Ilse Tebbetts, Bruce Bradtmiller, John McConville and Claire Gordon. 1988. Measurer's Handbook: U.S. Army Anthropometric Survey, 1987-1988. Technical Report NATICK/TR-88/043, U.S. Army Natick Research, Development and Engineering Center, Natick, Massachusetts.

6. Gordon, Claire C. and Bruce Bradtmiller. 1985. Proposed Sampling Strategy for a U.S. Army Anthropometric Survey. Technical Report. U.S. Army Natick Research, Development and Engineering Center, Natick, Massachusetts. (In press.)

This document reports research undertaken in cooperation with the US Army Natick Research, Development and Engineering Center under Contract No. DAAK60-86-C-0128 and has been assigned No. NATICK/TR-88/045 in the series of reports approved for publication.

APPENDIX A.

Instruction Manual for the Operation of the Hardware and Software Used in the 1987-1988 U.S. Army Survey

APPENDIX A.

Instruction Manual for the Operation of the Hardware and Software Used in the 1987-1988 U.S. Army Survey

This appendix contains step-by-step instructions for describing the operation, care and maintenance of the portable computers used for data collection in the field. The standard anthropometric stations #1 to #6 use Compaq Portable Computers, and the software is similar for each of those stations. The first section below describes the operation of the software and hardware for the six standard data stations. A separate section describing the procedures required at the automated headboard station and the in- and out-processing station follows.

MEASUREMENT STATIONS #1 THROUGH #6

Hardware

1. The hardware used at stations #1 to #6 consists of the Compaq Portable Computer and the Brother M-1109 printer (see Figure A-1).

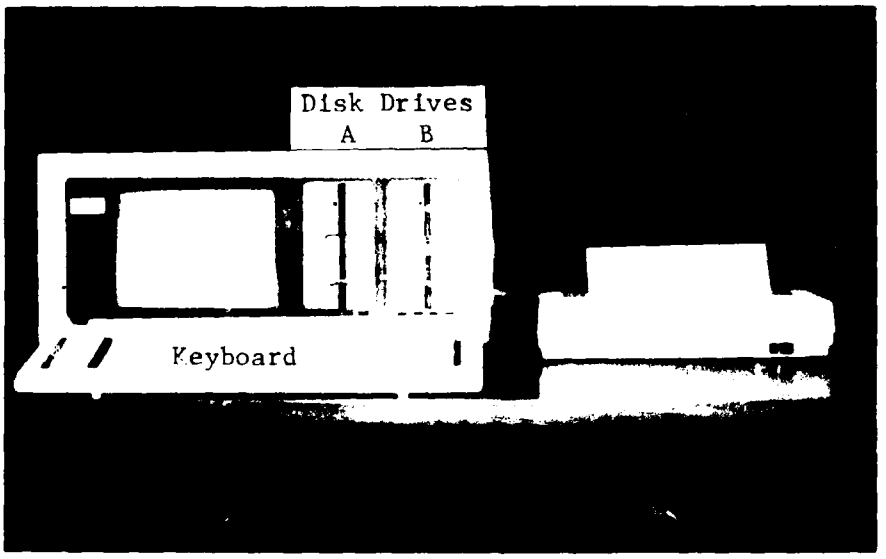


Figure A-1. Compaq Portable Computer in the open operating position with the Brother M-1109 printer.

2. Lift the computer onto the table or desk surface with the handle on the top. Locate the side of the computer with two fold-out tabs near the bottom. Flip these tabs out. Keeping the computer upright (with the handle on top), rotate it, if necessary, so that the side containing the tabs is away from you. On each end (to your right and left) near the top, is a door. Open each door by pushing in slightly on the bottom of the door and then sliding the door down. The door on the left contains the power cord for the unit. Remove the cord and affix the female end onto the prongs located inside the door. The door on the right contains the slots for attaching accessories, such as the grey printer cable. The printer cable has two ends -- one slightly smaller than the other. Attach the smaller end to the printer port inside the right-hand door. When the correct end is selected, it will fit into only one plug in the side of the computer. It will also fit only one way, so that if it does not go in smoothly on the first attempt, flip it over and try it again. After the cable is snugly attached, secure it by screwing in the screws with a small screwdriver. Now lay the computer so that the side with the tabs is down, and the handle is in back, and place it on the table resting on the tabs. The front of the computer, now facing you, contains small tabs on the right and left sides, near the bottom. Flip these tabs out. Above each tab is a keyboard lock. When you move these locks to the open position, the keyboard will fall out toward you. Position your hands in such a way as to catch the keyboard when you move the locks to the "open" position.

3. The Brother printer should be placed in a convenient location relative to the computer and to you. The printer end of the printer cable should be attached to the port on the rear of the printer with the wire clips. It, too, will fit only one of the plugs, and will fit only one way. When it is snugly attached, flip the wire clips toward each other until they lock the cable into place.

4. Both the printer and the computer power cords should be plugged into the multiple outlet power strip. The power strip is then plugged into a three-pronged wall outlet, or into a three-pronged adapter for a two-pronged wall outlet.

5. The computer has two disk drives (vertical slots above the keyboard on the right side of the machine) which accommodate standard 5 1/4-inch floppy diskettes. The drive on the left is known as Drive A; the drive on the right is Drive B. Flip up the levers on each drive and remove the cardboard disk drive protector from each drive. (These cardboard protectors should be retained and reinserted to protect the disk drives whenever the computer is moved.)

The station diskette (see Figure A-2) which is labeled, for example, "Station 1 Disk" should be inserted with the double-notched edge leading and the single notch on the bottom. Insert the station diskette into Drive A (on the left) until it doesn't go any farther. Move the drive handle clockwise until it closes the drive and retains the disk in place.

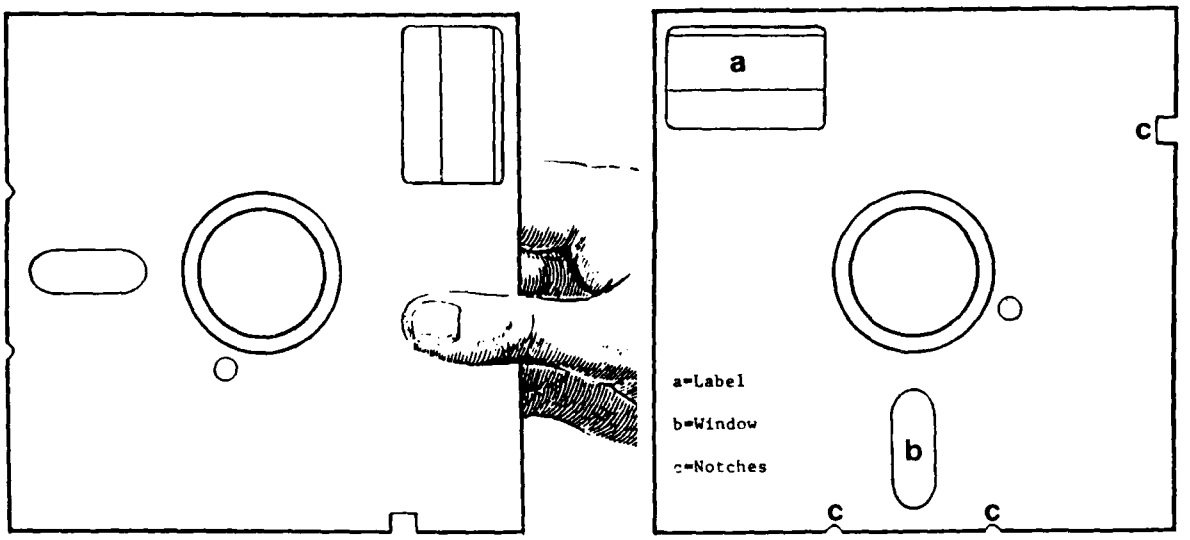


Figure A-2. A typical diskette (right), positioned for insertion into disk drive (left).

6. Turn on the switch on the power strip. Turn on the computer with the switch located inside the door where the power cord is attached. Turn on the printer with the switch located on the front panel, near the bottom right-hand corner.

Software

1. The computer will "boot" automatically. This means that the operating system software required for the operation of the machine will be read from the station diskette. In addition, it will make a number of automatic checks, including a check on the connection with the printer and confirmation that all memory locations are functional. The computer will also read, from the diskette, the program that will accept the anthropometric data. When all the automatic booting, loading, and reading is finished, the computer system is ready to accept input from the operator. During this process, the red light on each drive will come on periodically. Never insert or remove a diskette while the red light is on.

2. The screen will prompt for the current date with **ENTER NEW DATE (MM-DD-YY):** You respond with the correct date, in numbers, with the month, day, and year, using two characters for each entry. Separate the numbers with hyphens, for example, 08-15-87. Follow this entry with a carriage return (see Figure A-3 for location of RETURN key). The return key must be pressed after

each entry from now on. Next, the prompt will be for the time, **ENTER NEW TIME:** You respond with the time of day, for example, 09:30. Use 24-hour military time; for example, 2:30 in the afternoon would be entered as 14:30. (See Figure A-3 for location of the shift key to type a colon.)

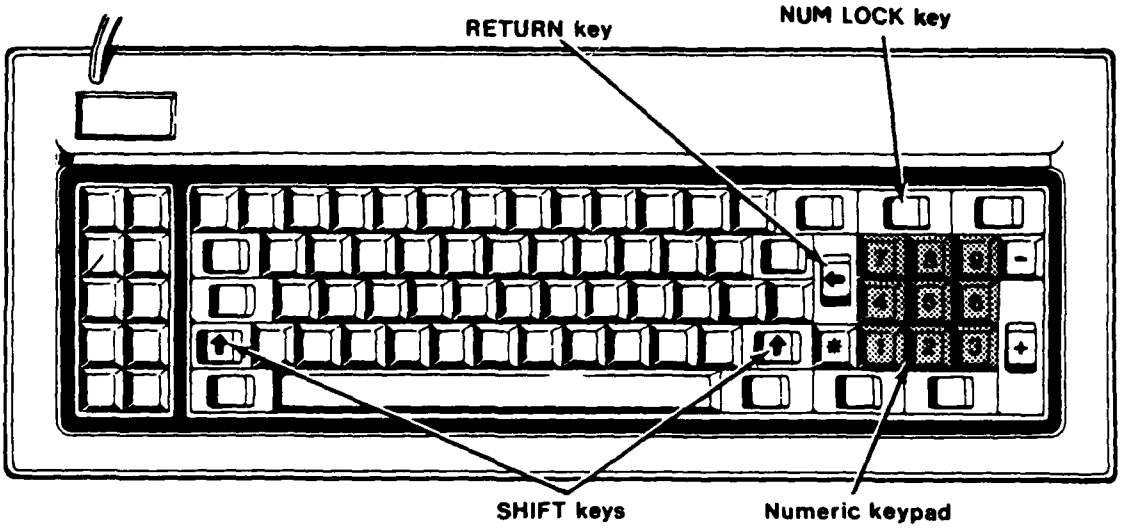


Figure A-3. Compaq computer keyboard.

3. After the recorder enters the date and time, the program will respond with **GOOD MORNING. NOW IT IS TIME TO MEASURE SOME SOLDIERS.** It then requests the sex of the current group of subjects, with **WHAT SEX IS TO BE MEASURED? (M/F).** You respond with M when males are to be measured and F when females are to be measured. The program verifies the input by responding **WE NOW MEASURE WOMEN** or **WE NOW MEASURE MEN**, as appropriate.

4. The program next informs you of the status of the session so far. It first tells you what sex is being measured: **WE ARE MEASURING MEN** or **WE ARE MEASURING WOMEN**, as appropriate. It then says **MEASURER'S NAME IS XXX**, where **XXX** is the measurer's name. A third line says **CHANGE MEASURER'S NAME? (Y/N)** and the appropriate responses are Y for yes and N for no, or -1 to return to the previous step. This is used to set the measurer's name when the program is first begun, and it is used subsequently when the measurer and recorder switch roles between subjects. When using the program for the first time each day, either the Y or N response will elicit the next prompt, **ENTER THE MEASURER'S NAME.** Later, only the Y response will elicit the prompt for the measurer's name. Your first name is sufficient as a response to the prompt. After these initial questions, the program is ready to accept data from each subject. At this point, insert the subject's diskette into disk drive B if you have not already done so.

5. The subject's diskette should be inserted into Drive B (on the right) in the same manner as the station's diskette was inserted into Drive A. Then insert the subject's data recording form into the printer. It should be centered between the marks drawn on the top panel of the printer. Insert the form as you would a paper into a typewriter and move the paper by moving the thumb roller away from you. The thumb roller is located on the left side of the top panel. Roll the paper in until the top edge is just at the edge of the clear plastic shield. Press the orange on-line button on top of the printer. The yellow light should be lit.

6. The next prompt is **TYPE IN THE SUBJECT NUMBER**, and the appropriate response is the subject number on the subject's diskette and his/her data form. Numeric data may be entered from the numbers on the top row of the keyboard (as on a typewriter) or from the numeric keypad on the side of the keyboard (see Figure A-3) which is more efficient to use when only numbers are being punched in. Note that the *, +, and - keys used in this software can all easily be reached from the numeric keypad. If the keypad is to be used, it is necessary to press the NUM LOCK key at the top of the keypad (see Figure A-3).

7. After typing in the subject number, the program is ready for the anthropometric data. The next prompt will depend on which station is being used. Assuming station #1, for example, the prompt is **PLEASE MEASURE #1 STATURE**. The correct response is the subject's stature, in millimeters (mm). No decimal points will be used. Thus if the person's stature is 185.7 cm or 1857 mm, you should enter 1857. At the same time the entry is handwritten on a data form provided at each station as insurance against a power failure. After each measurement is entered, the prompt for the next measurement will be displayed.

If, after entering a value, you realize you have entered it incorrectly, enter a -1 for the next value. This will cause the program to prompt **RESTART MEASURING SEQUENCE AT VARIABLE NUMBER**. You can then enter the number of the dimension whose correct value you wish to enter, and the program will ask for that dimension again, as in **PLEASE MEASURE #2 CERVICALE HT**. If you enter a -1 for the variable number, you will be returned to the prompt **CHANGE MEASURER'S NAME**.

Occasionally it will not be possible to make a measurement. This might occur if the subject has had surgery or some disfigurement which would make the measurement meaningless. In such a case, the correct value to enter is zero (0). Since this is not a usual response, however, when a zero is entered the program will respond, for example, **BITRAGION-CRINION ARC IS ZERO!! IS THIS CORRECT? (Y/N)**. If the value is actually missing, simply respond Y or +, and the program will move to the next dimension. If the zero was entered in error, an N or - response will cause the program to ask for that dimension again, as **PLEASE MEASURE #3 BITRAGION-CRINION ARC**.

8. The data entry software contains a number of checks to verify that the data are entered accurately. The first of these is a range check to see if the entered value is within the range (minimum and maximum) of previously measured subjects for that dimension. If the entered value is outside that range, an auditory signal will be given and the screen will display a number of choices. Using CERVICALE HEIGHT as an example, let us say that the

correctly measured value is 1585. Let us further suppose that the recorder has entered the value 1855, in error. The screen will display the following:

MIN = 1424.6 MAX = 1671.6

THIS VALUE IS NOT IN RANGE!

PLEASE CHECK THE INPUTED VALUE FOR CERVICALE HT

IF YOU WANT TO CHANGE THE VALUE TYPE IN C OR *

IF THE SUBJECT IS SMALL TYPE IN S OR -

IF THE SUBJECT IS LARGE TYPE IN L OR +

This alerts you to the possibility of an error. If the recorder realizes his/her error, he/she can enter a C or *. When one of those is entered, the response is **CERVICALE HT =** , the corrected value can be entered and the program goes on to the next dimension.

To continue with the CERVICALE HEIGHT example, if the correct value were 1685 and that were entered, the same display would appear on the screen. Since in this case the subject is simply large, and the value is therefore out of range, the correct response is L or + (the subject is large). Similarly, if the correct value were below the bottom of the range, you would respond S or -.

After all the dimensions have been measured and entered, the program will display a table showing the names of all dimensions and the recorded values of all dimensions. At the bottom of the table, the following prompt is displayed: **DO YOU WANT TO CHANGE ANY VALUES? (Y/N).** If all the values are appropriate you respond, simply, N. However, if upon checking values against the handwritten entries on the backup data forms, discrepancies are apparent, you respond Y. The screen displays: **TYPE IN THE VARIABLE NUMBER AND THE NEW VALUE WITH A SPACE IN BETWEEN.** One could respond, for example, 2 1588, which would refer to variable number 2, CERVICALE HEIGHT, and the new measurement for that dimension. If you change a value, the table is displayed again, and the question is again posed, **DO YOU WISH TO CHANGE ANY VALUES?**

In addition to the measured dimensions, the display will also show nonmeasured variables. These are sex (number 31), measurer's name (number 32), system (number 33) and erase (number 34). If at this time you discover that the sex or measurer's name is wrong, it can be changed. Note that at this point the sex is numerically coded so that 1 = male and 2 = female. If you change variable 33 (system) to 1, you will have access to DOS and can enter any DOS command. (The software is currently operating in a DOS 3.2 environment.) After entering the command, enter a blank line (the RETURN key) to return to the program. If you set variable 34 (erase) to 1, all data for that subject are eliminated, and the program is ready for a new subject. This should be used when, from reviewing the table, it is clear that there has been a serious confusion and most of the values are incorrect. It is also used in the unlikely event that a subject is not able to complete the measuring sequence.

If, on reviewing the table, it is clear that there has been a serious confusion and most of the values are incorrect, the best approach is to start with that subject again. In such a case, respond Y or + to the question about changing values. When the program asks for a variable number and the new value, type 34 1. This will produce the prompt, **DO YOU WANT TO ERASE ALL THE DATA ENTERED FOR THIS SUBJECT AND ENTER A NEW SUBJECT? (Y/N)**. If you respond yes here, the program will return to **CHANGE MEASURER'S NAME?**. A response of no will, of course, cause a return to the table of displayed values.

If there are no more values to be changed, enter N. This causes the program to enter the second phase of data editing. In this phase, the value for each dimension is checked against the values of other dimensions for that subject, to verify that they all "make sense."

9. If there are one or more values which do not correspond with other values for that individual, an audible signal is given, followed by the prompt **PLEASE REMEASURE WAIST HT**, or whatever dimension might appear to be in error. The proper response here is to actually remeasure the dimension indicated. Do not attempt to remember if the original value was entered in error. Simply ask the measurer to retake the measurement and enter the new value. After the new value is entered, each value is again checked. Sometimes several values will need to be remeasured. Occasionally some values may be remeasured a second time. It may occur that several remeasures are caused by a misentered value early in the remeasuring sequence. When this occurs, a -1 entered as a value for any dimension will cause the program to return to the table of measured values. From there, the errant value can be corrected. After the correction is made, and you respond N (or -) to a request for more changes, the new values are edited again.

Sometimes, even after a second remeasure, the program still finds the value out of the ordinary. In such a case the screen will prompt with **WRITE AN EXPLANATION PLEASE.** Respond to this by typing in a few words of explanation about why the measurement(s) appear(s) exceptional. Do not use a comma in the text of your explanation. This may result in other survey software having difficulty reading data from the subject's diskette.

Although it is unlikely to be needed, it is even possible at the explanation stage to return to the table of measured values. In such a case, the first two characters of the explanation must be -1. This response will return you to the table of measured values, which will allow the change of any value, or exit via variable 34. It should be noted, however, that if you return to the table of measured values from the explanation (using the -1) and change a value, the new set of values is edited only one final time. If you attempt to return to the table from the explanation line a second time, you can change values but these values will not be edited. Such an approach is not recommended because it circumvents the editing procedures and could result in bad data points creeping into the data set. The preferred action is to notify your supervisor of the problem so that the necessary corrections can be made when the data are compiled.

10. After all the data have been entered, corrected, and/or explained, the data are written onto both the station diskette and the subject diskette. This occurs automatically and requires no action on your part.

11. The printer then prints the values on the subject's hard copy data sheet. Occasionally the program will appear to "lock up" at this point. Usually this means that it is trying to send data to the printer and is unsuccessful for some reason. If printing does not start right away, first check to see if the ONLINE light is lit. If it is not, press the orange ONLINE button to light it, and printing will proceed. If no lights on the printer control panel are on, this means that there is no power to the printer. Check the ON/OFF switch or power cord, as appropriate. After restoring power to the printer, printing will proceed. When the printing is finished, remove the paper from the printer, remove the subject's diskette from Drive B (on the right), and give both to the subject for transporting to the next station.

12. The program will prompt you with **CHANGE THE MEASURER'S NAME? (Y/N)**, and you can repeat from Step 4. If, at this point, you need to measure subjects of a different gender, do so by replacing the measurer's name with the word "Sex". This will return you to the question **WHAT SEX IS TO BE MEASURED**. If you have measured the last subject for the day, you can exit the program in the following way: respond Y to the question about changing the measurer's name, and when you are prompted for the new name, respond with END. This tells the program you are finished measuring for the day, and stops execution of the program.

13. Turn off the machine, turn off the printer, and turn off the power strip. If you are leaving the measuring location and need to prepare the computer equipment for travel, perform steps 1 through 5 of the Hardware section in reverse order.

MEASUREMENT STATION #7

As at stations #1 through #6, the software to be used at station #7 is specifically designed to accommodate the dimensions at that station. The measurement system at station #7 is the automated headboard device (AHD). The software associated with the AHD is described in detail in a technical report describing the development of the headboard,[1] and Appendix D of the measurer's handbook,[5] and will not be repeated here. The hand photographic system is also located at this station but does not require computer software.

IN/OUT-PROCESSING STATION AND STATION #8 (FOOT MEASURING)

The software at this station serves a variety of functions. First, it establishes the initial data record for the subject, collects his or her biographical information from the data sheet, accepts three foot measurements, and finally, verifies that the subject has visited every station, and that the other station computers have correctly written data onto the diskette. To activate this software, insert the program disk into drive A and type in INOUT8.

The screen at this station initially displays 8 choices:

1 IN PROCESS

2 MEASURE FEET

3 OUT PROCESS

4 ENTER QUESTIONNAIRES

5 COPY QUESTIONNAIRES

6 USE DOS SYSTEM

7 END PROGRAM

8 WRITE SUBJECT'S FILE TO SCREEN

WHAT DO YOU NEED TO DO?

In-process

This choice is used to create a subject disk for each subject. The screen will display **PUT A FORMATTED DISK IN DRIVE B PLEASE** and then ask you to do the following four things (one at a time):

TYPE IN THE SUBJECT'S NUMBER

TYPE IN THE SUBJECT'S SEX (M/F)

TYPE IN THE SUBJECT'S ACTUAL WEIGHT

TYPE IN THE SUBJECT'S ACTUAL HEIGHT

The subject's actual height and actual weight are found on the back of the Biographical Data form. After you have entered the subject's actual weight, the screen will display all the values you have entered and give you an opportunity to change what you have entered. For example, the display might show:

NSUB = 1234

SEX = M

ACTUAL HEIGHT = 1567.0

ACTUAL WEIGHT = 876.0

TYPE E OR * TO EXIT TO MENU

IS ALL THE ABOVE INFORMATION CORRECT? (Y/N)

If the information is not correct, respond N to the question, and you will be given the opportunity to correct the entries. If you wish to continue in-processing, respond Y to the final question and you will be prompted for another subject number. If you respond N, you will be returned to the menu and can make another selection. If you enter -1 for a subject number, you will also be returned to the menu.

Measure Feet

The software for this menu item is nearly identical to that of the anthropometric data collection software at stations #1 through #6. It is for the collection of the three dimensions taken in the footboxes. After the data from a subject have been entered and edited, the program will ask **DO YOU HAVE MORE FEET TO MEASURE (Y/N)?** As before, if you respond Y, you will go to the beginning of the anthropometric data entry section, and enter a new subject number and the foot measurements. If you respond N, you will be returned to the menu, and can select other operations.

Out-process

The function of the software controlled under this menu item is to verify that the software at the other stations has functioned correctly and to compile a cumulative data record of all subjects measured.

When this item is selected, the screen displays:

PLEASE PUT THE NEXT SUBJECT'S DISK IN DRIVE B.

PLEASE TYPE IN THE SUBJECT NUMBER.

As with all other features in this software, the program will check the entered subject number with that on the disk and alert you if they do not match. If the subject numbers do not match, you will be asked to retype the number, or insert another disk, whichever is appropriate. You are then asked to type in the subject number again.

When the subject number you have typed in agrees with the subject number on the disk, the program functions essentially automatically. In the case where all the stations have been recorded on the disk correctly, the screen will display the subject number and inform you of the progress of the program:

SUBJECT 1234

STATION.1 HAS BEEN READ IN.

STATION.2 HAS BEEN READ IN.

STATION.3 HAS BEEN READ IN.

STATION.4 HAS BEEN READ IN.

STATION.5 HAS BEEN READ IN.

STATION.6 HAS BEEN READ IN.

STATION.8 HAS BEEN READ IN.

THE HEAD X-Y-Z COORDINATES HAVE BEEN READ IN.

If the subject has been remeasured, the screen will display the number of the station at which remeasurement occurred. If the subject was not remeasured, the screen will also display that information.

The screen will then ask you whether or not you have more disks to enter. If you answer yes, the screen will indicate that you should put the next subject disk in and ask you to type the next subject number. If you answer no, you will be returned to the menu.

Occasionally, there will be subjects whose disks are, for one reason or another, incomplete. If data from station #1, for example, are missing from the disk, the following message will be displayed:

THE FILE FOR STATION.1 IS MISSING OR DAMAGED.

THERE IS A PROBLEM; PLEASE CHECK THE SUBJECT'S PRINTOUT.

M- THIS STATION IS MISSING FROM THE SUBJECT PRINTOUT.

H- THIS SUBJECT WAS MEASURED ONLY AT THE HEADBOARD.

E- THIS STATION WAS PRINTED ON THE SUBJECT PRINTOUT. ENTER DATA VALUES.

P- PAUSE TO USE DOS SYSTEM COMMANDS.

C- CANCEL THIS SUBJECT.

PLEASE TYPE IN THE APPROPRIATE LETTER.

As the screen indicates, you should check the subject's printout which accompanies the diskette and verify whether the subject was measured at the station in question. If he or she was measured, then type in E, which will prompt you for the values of dimensions measured at that station. If he or she was not measured at that station, then type in M, and the program will go on to check the next station. If the printout shows that the subject was measured only at the headboard, or at the headboard and station #4 (traditional head and hand measurements), type in H, and the program will look only for data from the headboard and Station 4. Occasionally it will be useful to use the DIR command from DOS to view the directory on a subject disk. When this is the case, type in P, which will allow you to enter any DOS command. After entering the DOS command, you will be returned to the program. If, for any reason, you wish to stop entering data from the subject's disk, type in C, for cancel.

If the headboard data are missing from the disk, there are several things that happen. First, you will be instructed to check the subject's printout

to see if the individual was measured at the headboard. If the subject was measured at the headboard, enter +. If the subject was not measured at the headboard, enter -. If the subject was not measured at the headboard, and you enter the minus sign, the program goes on to check for remeasured stations. If the subject was measured at the headboard, enter the plus, and the following is displayed:

ENTER THE FOLLOWING COMMAND: DIR B:*.YS

When you enter the DIR command as indicated, all the headboard files on the subject's disk will be displayed on the screen, such as:

```
VOLUME IN DRIVE B: HAS NO LABEL
DIRECTORY OF B:/
SN03456          YS       234          1-17-88      2:24P
1 FILE(S)              307869 BYTES FREE
```

The program will then ask:

PLEASE TYPE IN THE NUMBER OF FILES DISPLAYED ON THE SCREEN.

In general, there will be either one headboard file, or none at all. If there is one, enter 1, and the screen will then ask you to type in the name of the file as it appears on the directory that has just been listed on the screen. In the example above, you would type in: SN03456. You may include .YS following the name, but it is not necessary. The program will then read in the file, and proceed to check for remeasured stations.

In the case that the directory shows no files, the display will be:

```
VOLUME IN DRIVE B:  HAS NO LABEL
DIRECTORY OF B:/
FILE NOT FOUND
```

Then you would type in 0 when the program asks you to type in the number of files displayed on the screen. The program will then say:

```
PLEASE TYPE IN THE X-Y-Z COORDINATES.
TYPE IN THE THREE COORDINATES WITH A SPACE IN BETWEEN.
R.  TRAGION      =
```

You then type in the X-Y-Z coordinates which are found on the back of the subject's printout. After you type in those for R. Tragion, the next point will be provided as a prompt. After you have entered the last point, the screen will show: **THE HEAD X-Y-Z COORDINATES HAVE BEEN READ IN** and will indicate whether or not the subject has been remeasured at one or more stations. After that display, you are asked, as always, whether you have more data to enter. If you answer yes, you are directed to the next subject. If you answer no, you are returned to the menu.

The data for each subject are compiled into large files on the disk in Drive A. This disk should be replaced with a new formatted disk at the beginning of each week of measuring. Failure to do this will result in loss of data.

Enter Questionnaires

This menu item is used to enter data from the Biographical Data forms. The screen will first display a list of the measuring locations and ask for your selection. Specifically, it will show:

1. **FORT MCCLELLAN**

2. **FORT CAMPBELL**

3. **FORT BRAGG**

4. **FORT STEWART**

5. **FORT DIX** . . . . . etc.

PLEASE ENTER YOUR SELECTION FOR TODAY'S POST.

After you have entered the appropriate number, the name of that post will be displayed at the top of the screen, beside the day's date. It will then ask, **DO YOU WISH TO CHANGE TODAY'S DATE? (Y/N).** If the date is correct, you respond N. If you are entering questionnaire data on a day other than when the form was filled out, respond Y and you will be asked for a new date. Enter the date that is written on the form. As in In-process, the program will next ask you to put a diskette into Drive B, and then display **TYPE IN THE SUBJECT NUMBER PLEASE.** When you enter the number, the program will check to see if the subject file has been previously created with In-process. If the file has previously not been created on the diskette, it will be created. The program then follows exactly the order of questions on the biographical data questionnaire (see Appendix D), except that the subject's name (Question No. 1) is not entered to protect the subject's privacy.

2. **UNIT TO WHICH YOU ARE ASSIGNED AT THIS POST:**

   COMPANY OR BATTERY

   BATTALION

   REGIMENT OR GROUP

   BRIGADE

   DIVISION

Each of the parts in Question 2 is responded to exactly as the subject has filled out his/her questionnaire. Because subjects often appear in groups from a single unit, the program was designed to minimize entry errors. After the first subject, the unit data for that subject are displayed automatically, followed by the question, **DO YOU WISH TO CHANGE UNIT FOR THIS SUBJECT? (Y/N).** If the unit is correct (i.e., the same as for the previous subject), respond N and proceed to Question 3. If the unit is different from the previous subject, respond Y, and the program will ask for each item in turn.

3. **MILITARY COMPONENT:**

1-REGULAR ARMY 2-ARMY RESERVE 3-NATIONAL GUARD 4-MISSING

ENTER THE APPROPRIATE NUMBER:

4. **MILITARY PERSONNEL CLASS:**

E-ENLISTED W-WARRANT OFFICER C-COMMISSIONED OFFICER M-MISSING

ENTER THE APPROPRIATE LETTER:

If either W or C is entered, a follow-on question is displayed: **SPECIFY BRANCH**, and you enter exactly what the subject has written on his or her form.

5. **RANK/GRADE RANK: GRADE:**

You enter the rank from the form using three characters (e.g., PVT for Private, 1LT for First Lieutenant, and so on), and then the grade consisting of two characters (e.g., E3, O5, etc.).

6. **TIME IN SERVICE YEARS: MONTHS:**

7. **MOS PRIMARY: SECONDARY:**

After you enter the primary MOS, as the subject has written it, the program will ask for a secondary MOS. If this is recorded, enter it as written. If there is none, just hit the carriage return for the next question.

R-RIGHT L-LEFT E-EITHER

8. **WITH WHICH HAND DO YOU USUALLY FIRE A WEAPON?**
ENTER THE APPROPRIATE LETTER:

9. **WITH WHICH EYE DO YOU USUALLY SIGHT YOUR WEAPON?**
ENTER THE APPROPRIATE LETTER:

At the bottom of the screen, the program asks **DO YOU WISH TO CHANGE ANY OF THE ABOVE INFORMATION? (Y/N).** If you respond Y, you will be asked which question to go back to and given the opportunity to change your response. If you answer N, the program will go to the next series of questions, found on the second page of the biographical data questionnaire.

The title at the top of the next display corresponds to page 2 of the biographical data questionnaire, Biographical Data: Personal History. As before, the questions on the screen follow almost exactly the questions on the form.

1. **TYPE IN THE SUBJECT'S BIRTHDATE PLEASE.**
(MONTH, DAY, YEAR)

The appropriate response is the birthdate of the subject, using numbers for the months. Each part of the date consists of two characters and is separated by a comma and/or a space.

2. TYPE IN THE SUBJECT'S AGE PLEASE.

Respond with the subject's age as he or she has written it on the form.

3. TYPE IN M FOR MALE OR F FOR FEMALE PLEASE.

4. TYPE IN A LETTER FOR THE SUBJECT'S RACE PLEASE.

W-White, not of Hispanic origin

B-Black, not of Hispanic origin

H-Hispanic

A-Asian/Pacific Islander

I-American Indian/Alaskan Native

M-Mixed

O-Other

If M or O are entered, the program will respond **PLEASE SPECIFY:**, and you should enter what the subject has written on his or her form. In the case of Mixed, the format is generally one of the above races, followed by a slash, and then another of the races. For example, "White/Hispanic" would be a valid response.

5. HOW TALL IS THE SUBJECT IN BARE FEET?

FEET:

INCHES:

6. HOW MUCH DOES THE SUBJECT WEIGH, WITHOUT CLOTHES, IN POUNDS?

Both 5. and 6. are entered directly as the subject has written them on the form.

7. DOES THE SUBJECT WEAR:

G FOR PRESCRIPTION GLASSES C FOR PRESCRIPTION CONTACT LENSES

B FOR BOTH N FOR NEITHER M FOR MISSING

TYPE IN A LETTER.

8. WITH WHICH HAND DOES THE SUBJECT USUALLY WRITE?

ENTER THE APPROPRIATE LETTER.

R-RIGHT L-LEFT E-EITHER M-MISSING

After answering questions 1 through 8, the program displays the entered answers in an easily readable format at the bottom of the screen and asks the question, **DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMATION (Y/N)?**, to which you answer N if all the information is correct and you want to go on to the next page, or Y if one or more items need to be changed. If you respond Y, the program will prompt for the number of the item to which you wish to return, and the correct entry may be made.

On the third page of the form, the first question is **9. DOES THE SUBJECT CURRENTLY PARTICIPATE IN RESISTANCE OR FREE WEIGHT TRAINING AT LEAST ONCE A WEEK (Y/N)?** If the subject has responded N, enter that and the program will proceed to question 10. If you enter Y, the program will display:

A. HOW LONG HAS THE SUBJECT BEEN INVOLVED IN THIS TRAINING?

YEARS: MONTHS:

B. HOW MANY DAYS PER WEEK DOES THE SUBJECT NOW TRAIN?

UPPER BODY: DAYS PER WEEK:

LOWER BODY: DAYS PER WEEK:

C. ON THE DAYS THAT THE SUBJECT DOES TRAIN, HOW MANY HOURS PER DAY DOES THE SUBJECT TRAIN?

UPPER BODY: HOURS PER DAY:

LOWER BODY: HOURS PER DAY:

In each case, you enter the numbers shown on the subject's form.

10. DOES THE SUBJECT RUN ON A REGULAR BASIS (Y/N)?

As in question 9 if the subject does not run, enter N, and the program will proceed to the next item. If the subject does run, enter Y, which will invoke the following prompts:

A. HOW LONG HAS THE SUBJECT BEEN RUNNING?

YEARS: MONTHS:

B. HOW MANY DAYS PER WEEK DOES THE SUBJECT RUN?

DAYS PER WEEK:

C. HOW MANY MILES PER DAY DOES THE SUBJECT USUALLY RUN?

MILES:

After questions 9 and 10 have been answered, the screen displays the information from those two questions in an easily readable format and prompts: **DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMATION (Y/N)?** As before, a Y response allows a change, and the N response moves to questions on the last page of the form.

The first three questions on the last page of the form are:

11. SUBJECT'S BIRTHPLACE:

12. MOTHER'S BIRTHPLACE:

13. FATHER'S BIRTHPLACE:

and the correct response is the two-letter postal abbreviation for the state in which the subject, the mother, and the father were born. The subject will often include the name of the city or the county, and will often write out the name of the state. The only thing that should be entered is the abbreviation for the state. For example, if the subject has written that his birthplace is "Little Rock, Arkansas", you would enter "AR". Where the response is a foreign country, enter the complete name of the country, but not the city or state within that country. For example, if the subject's mother were born in Ontario, Canada, you would enter "Canada" for question 12.

The screen next prompts with:

14. TYPE IN A LETTER FOR THE SUBJECT'S MOTHER'S RACE PLEASE.

W-White, not of Hispanic origin

B-Black, not of Hispanic origin

H-Hispanic

A-Asian/Pacific Islander

I-American Indian/Alaskan Native

M-Mixed

O-Other

As before, if you enter M or O, you will be prompted to specify the other race or the racial mix.

Similarly, question 15 is:

15. TYPE IN A LETTER FOR THE SUBJECT'S FATHER'S RACE PLEASE.

W-White, not of Hispanic origin

B-Black, not of Hispanic origin

H-Hispanic

A-Asian/Pacific Islander

I-American Indian/Alaskan Native

M-Mixed

O-Other

and you respond in like fashion.

The responses to the three parts of Question 16 on ethnicity/national extraction have been obtained by interview and are quite consistent in format. For these questions, you type in exactly the same information that is written on the form. The format of the question on the screen is:

16. ETHNICITY/NATIONAL EXTRACTION

SUBJECT:

MOTHER:

FATHER:

Next, the following appears:

17. BODY DIMENSIONS

ACTUAL HEIGHT: **ACTUAL WEIGHT:**

And you respond with the same information that is written on the form. Again, these have been obtained by team members, not specified by the subjects themselves, and they are to be entered exactly as written, which is in the form of three or four digits, as appropriate, without a decimal point.

Finally, the program will ask, one last time, if you want to change any of the information entered from the last page. *If you do not*, respond N and the next question will be **DO YOU HAVE MORE DATA TO ENTER (Y/N)?** If there is another subject biographical form you wish to enter, respond Y, and the screen will show **PUT THE SUBJECT'S DISK IN DRIVE AND THEN PRESS ENTER.** This will allow you to go on to the next subject's form. If you do not intend to enter another form, respond N to the question, and you will be returned to the menu.

Copy Questionnaires

This program is used when questionnaire data have been entered while the subject is being measured at Stations #1 through #7. In such a case, the subject will have his/her diskette at the stations, away from the In- and Out-Processing station. Thus the questionnaire data would not have been entered on the subject's disk but on a temporary storage disk kept at the In- and Out-Processing station. When such a subject returns for out-processing, the questionnaire data will need to be copied from the temporary storage disk onto his subject disk. That is the purpose of this menu item. When this is selected, the screen will display:

PLEASE PLACE THE TEMPORARY STORAGE DISK IN DRIVE A.

PLEASE PLACE THE SUBJECT'S DISK IN DRIVE B.

TYPE IN THE SUBJECT NUMBER PLEASE.

After you type in the subject number, the rest of the program's function is automatic and does not require operator input. When the copying is complete, the screen will display:

REMOVE THE SUBJECT'S DISK FROM DRIVE B.

DO YOU HAVE MORE QUESTIONNAIRE DATA TO COPY (Y/N)?

If you respond Y, you can copy data onto the next subject's disk. If you respond N, you will be returned to the menu.

Use DOS System

This item on the menu makes it possible to temporarily "jump out" of the program and execute a DOS command. Although any DOS command could be used, probably the most useful one in the context of this software would be DIR, the command which lists a directory of files on the disk.

To use this item, simply enter 6 from the menu. The screen will display:

PAUSE - PLEASE ENTER A BLANK LINE (TO CONTINUE) OR A DOS COMMAND.

You can then enter whichever DOS command you need. After the DOS command has been executed, the program will prompt:

PRESS THE ENTER KEY TO RETURN TO THE MENU.

and you can easily return to the program by pressing the ENTER key.

End Program

This item is self-explanatory. It will return the **A>** prompt.

Write the Subject's File to Screen

The last item on the menu is used when you wish to check that the out-processing is correctly writing information to the data disk in Drive A. This reads data from that disk and displays it on the screen. When you select this menu item, the screen prompts with **TYPE IN THE SUBJECT NUMBER YOU WANT TO SEE.** After you enter the subject number, the display is **STATION 1 PRESS THE ENTER KEY TO VIEW THE NEXT STATION** followed by all the data collected at Station #1 in an abbreviated tabular format. This includes the subject number, the measurer's name, the sex of the subject (1=male, 2=female), any comments or explanations, all the anthropometric data, remeasure values for all dimensions which were remeasured due to the regression equations, and the standard errors of estimates calculated by the regression equations. That display remains on the screen until you press ENTER, which will display Station #2, and so on proceeding through Station #6, then #8 (feet), and ending with Station #7 (the headboard data). At the bottom of the display of headboard data, the prompt is **READ ANOTHER SUBJECT'S FILE? (Y/N).** If you respond Y, you get the prompt for a new subject number; if you respond N, you are returned to the menu.

CARE AND MAINTENANCE OF THE COMPUTER EQUIPMENT

The computer, while designed to be portable, is not indestructable. It should not be dropped, thrown, or have heavy equipment stacked on it. Treat it as you would any fine piece of electronic equipment. When handled with reasonable care, the computer requires very little maintenance. Periodically, when required, clean the screen with a spray window cleaner and a soft cloth.

The diskettes require no maintenance, but must be handled very carefully. The window, a cutout place in the diskette sleeve, is where the disk drive writes and reads information to and from the diskette (see Figure A-2). **IT IS VERY IMPORTANT NOT TO TOUCH THE SURFACE OF THE DISKETTE WHERE IT IS EXPOSED AT THE WINDOW.** To do so could damage the diskette and result in loss of data. When the diskette is not in the drive, it should be inserted into its protective jacket. Insert the diskette so that the window goes into the jacket first, and is completely protected. The diskettes should be stored upright, and kept away from magnetic fields and extreme temperatures.

APPENDIX B.

Program Source Code Listings: Page No.

PROGRAM INED

```
C     PROGRAM INED IS A DATA ENTRY ( IN ) AND DATA EDITING ( ED ) PROGRAM
      COMMON/ M / MINMAX(30,2,2), DATA(30,3), IS
      REAL MINMAX
      COMMON/ REG / EST(30), IVN(30,2,2), EQC(30,3,2), SE(30,2),
     + ER(0:30) ,KLM(0:30), NVS
      COMMON/ T / RME, NAME
      CHARACTER*18 NAME(30), MNAME, MN, A*1, BEEP*1, EXP*127,
     + STANUM*1, SUBNUM*5, SN*2, RE*2, SR*2, ISNAME(2)*6
      DIMENSION DAT(30), RME(30), IDAT(30)
      DATA NST / 1 /,MNAME /'    '/, ISNAME / ' MALE ', 'FEMALE' /
C     READ FILE REGEQ TO GET
      OPEN(9,FILE='REGEQ',STATUS='OLD')
C     STATION NUMBER, NUMBER OF VARIABLES, WERE TO PRINT OUTPUT ON PAGE,
C       CHECK VALUE
      READ(9,*) NSTA, NVS, NLD, CK
C     REMEASURE ERROR,
      READ(9,*) (RME(I),I = 1, NVS)
C     NAMES, MALE REGRESSION EQUATIONS,
      READ(9,'(A18,2I4,4F15.8)') (NAME(I), IVN(I,1,1), IVN(I,2,1),
     + (EQC(I,J,1), J = 1, 3), SE(I,1), I = 1, NVS)
C     AND FEMALE REGRESSION EQUATIONS.
      READ(9,'(18X,2I4,4F15.8)') (IVN(I,1,2), IVN(I,2,2), (EQC(I,J,2),
     + J = 1, 3), SE(I,2), I = 1, NVS)
      CLOSE(9)
      DO 2 I = NVS
    2 RME(I) = MIN( RME(I), SE(I,1), SE(I,2) ) / 2.0
      WRITE(*,'('' GOOD MORNING IT IS TIME TO MEASURE SOME SOLDIERS
     +'')')
C  READ IN THE CURRENT MINS AND MAXS
      OPEN(9,FILE='MINMAX',STATUS='OLD',ACCESS='DIRECT',
     + FORM='UNFORMATTED', RECL=120)
      DO 4 I = 1, 30
      DO 4 J = 1, 2
      DO 4 IS = 1, 2
      IREC = I + 30 * ( J - 1 ) + 60 * ( IS - 1 )
C     INITIALIZE VARIABLES AND OPEN FILES
    4 READ(9,REC=IREC) MINMAX(I,J,IS)
      CALL WSEX(IS)
      BEEP = CHAR(7)
      SN = 'SN'
      RE = 'RE'
      OPEN(6,FILE='PRN')
      WRITE(STANUM,'(I1)') NSTA
      OPEN(10,FILE='STATION.'//STANUM,MODE='READWRITE',STATUS='OLD')
C     CALL SUBROUTINE APPEND TO POSITION STATION FILE AT THE END
C       OF THE FILE
      CALL APPEND(NVS)
C     * * * * * *
C       THE PROGRAM COMES TO HERE AT THE START OF EACH SUBJECT
C     * * * * * *
   10 IF(IS.EQ.1) WRITE(*,'(A)') ' WE ARE MEASURING MEN'
      IF(IS.EQ.2) WRITE(*,'(A)') ' WE ARE MEASURING WOMEN'
```

```
      WRITE(*,'(A,A18)') ' THE MEASURER''S NAME IS ', MNAME
      WRITE(*,'(35H CHANGE THE MEASURER''S NAME? (Y/N) ,\)')
C     CHANGE MEASURER'S NAME?
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.MNAME.EQ.'    '.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
   12 WRITE(*,'(27H ENTER THE MEASURER''S NAME  ,\)')
      READ(*,'(A18)') MNAME
      WRITE(*,'(9X,A18)') MNAME
C     END PROGRAM
      IF(MNAME.EQ.'END'.OR.MNAME.EQ.'end') THEN
        CLOSE(9)
        CLOSE(10)
        STOP
      END IF
C     CHANGE SEX BEING MEASURED
      IF(MNAME.EQ.'SEX'.OR.MNAME.EQ.'sex') THEN
        CALL WSEX(IS)
        GO TO 12
      END IF
C     USE DOS COMMAND
      IF(MNAME.EQ.'SYSTEM'.OR.MNAME.EQ.'system') THEN
        PAUSE
        GO TO 12
      END IF
      ELSE
      IF(A.NE.'N'.AND.A.NE.'-'.AND.A.NE.'n') GO TO 10
      END IF
C     READ SUBJECT NUMBER FROM SCREEN
      WRITE(*,'('' TYPE IN THE SUBJECT NUMBER '',\)')
      READ(*,*,ERR=10) NSUB
      OPEN(4,FILE='B:SUBJECT.NUM',STATUS='OLD',IOSTAT=ISTAT,ERR=14)
C     READ SUBJECT NUMBER FROM SUBJECT'S DISK
      READ(4,*,IOSTAT=ISTAT,ERR=14) ISUB
      CLOSE(4)
   14 IF(ISTAT.LT.0) THEN
C     PROBLEM ON SUBJECT'S DISK - END OF FILE
        WRITE(*,'(1X,A1,A,A1)') BEEP, ' FILE SUBJECT.NUM IS BLANK',BEEP
        GO TO 10
      END IF
      IF(ISTAT.GT.0) THEN
C     PROBLEM ON SUBJECT'S DISK - ERROR
        WRITE(*,'(1X,A1,A,A1)') BEEP, ' FILE SUBJECT.NUM IS NOT ON THIS
     +DISK OR THERE IS SOME OTHER PROBLEM', BEEP
        GO TO 10
      END IF
      IF(NSUB.NE.ISUB) THEN
C     SUBJECT NUMBERS DO NOT MATCH
      WRITE(*,'(1X,A1,61H THERE IS SOME CONFUSION ABOUT THE SUBJECTS''S
     +SUBJECT NUMBER ,A1)') BEEP, BEEP
      WRITE(*,'(A,I6,/A,I6)') ' THE SUBJECT NUMBER ENTERED =', NSUB,
     + ' THE SUBJECT NUMBER FROM THE DISK =', ISUB
       GO TO 10
      END IF
C    * * * * * *
```

```
C     ENTER THE SUBJECT'S DATA
C    * * * * * * *
       EXP = 'OK'
       II = 1
C     FOR STATION #2 ONLY
   18  IF(NSTA.EQ.2) THEN
         CALL THREE
         II = 4
       END IF
   20  CONTINUE
C     LOOP #26 IS THE MAIN DATA ENTRY AREA
       DO 26 I = II, NVS
        WRITE(*,'('' PLEASE MEASURE #'',I3,1X,A18,''   '',\)') I, NAME(I)
C     A NOT REAL ENTRY WILL SEND THE PROGRAM BACK TO 'CHANGE THE
C       MEASURER'S NAME' PROMPT
        READ(*,*,ERR=10) DATA(I,1)
       IF(DATA(I,1).LT.0.0) THEN
C     IF DATA VALUE IS LESS THAN ZERO RESTART THE MEASURING SEQUENCE AT
C       A PREVIOUS VARIABLE
   22   WRITE(*,'(A,\)')' RESTART MEASURING SEQUENCE AT VARIABLE NUMBER '
        READ(*,*,ERR=24) II
C     IF VARIABLE NUMBER IS LESS THAN ZERO RESTART SUBJECT
        IF(II.LT.0) GO TO 10
C     VARIABLE NUMBER TOO LARGE, RESTART MEASURING AT CURRENT VARIABLE
        IF(II.GT.I) II = I
C     SET ZERO TO ONE
        IF(II.LT.1) II = 1
C     STATION #2 ONLY CALL SUBROUTINE THREE TO RESTART MEASURING AT
C       THE FIRST VARIABLE
        IF(II.EQ.1.AND.NSTA.EQ.2) GO TO 18
        GO TO 20
C     INCORRECT VARIABLE NUMBER ENTERED
   24   WRITE(*,'(1X,A1,A/A,A1)') BEEP, 'THERE WAS A PROBLEM WITH THE VA
     +RIABLE NUMBER ENTERED','CHECK AND REENTER', BEEP
        GO TO 22
       END IF
       IF(DATA(I,1).EQ.0.0) THEN
C     CHECK IF ZERO WAS INTENDED DATA VALUE
        WRITE(*,'(9X,A1,A18,A,A1)') BEEP, NAME(I),
     +    ' IS ZERO!! IS THAT CORRECT?  (Y/N)', BEEP
        READ(*,'(A1)') A
          IF(A.NE.'Y'.AND.A.NE.'y'.AND.A.NE.'+') THEN
            II = I
            GO TO 20
        END IF
       END IF
C     CALL SUBROUTINE MM TO CHECK IF DATA VALUE IS WITH IN RANGE
        IF(DATA(I,1).GT.0.0) CALL MM(I,NAME(I))
   26  CONTINUE
C    * * * * * * *
C    ALL VARIABLES HAVE BEEN MEASURED
C    * * * * * * *
C    CLEAR THE SCREEN
C     THE DATA POINTS ARE WRITTEN TO THE SCREEN
```

```
C      THE DATA POINTS CAN BE CHANGED IF NEEDED
   30 WRITE(*,'(1X,A1,''[2J'')') CHAR(27)
      DO 32 I = 1, NVS, 2
   32 WRITE(*,'(2(5X,I3,2X,A18,'' ='',F7.1))') I, NAME(I), DATA(I,1),
     + I + 1, NAME( I + 1 ), DATA( I + 1, 1 )
      WRITE(*,'(A,I1,2X,A6,A,A18,A)') ' 31 SEX = ', IS, ISNAME(IS),
     + ' 32 MEASURER''S NAME = ', MNAME, ' 33 SYSTEM  34 ERASE'
   34 WRITE(*,'('' DO YOU WANT TO CHANGE ANY VALUES? (Y/N) '',\)')
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
C     SO YOU WANT TO CHANGE A VARIABLE
        WRITE(*,'(A,A)') ' FOR VARIABLE NUMBERS, 32 MEASURER''S NAME,'
     + , ' 33 SYSTEM, & 34 ERASE USE 1 AS THE NEW VALUE'
        WRITE(*,'('' TYPE IN THE VARIABLE NUMBER AND THE NEW VALUE WITH A
     + SPACE IN BETWEEN '',\)')
C   READ IN VARIABLE NUMBER AND NEW DATA VALUE
      READ(*,*,ERR=36) I, DR
      IF(I.EQ.31) THEN
C     CHANGE SEX MEASURED
        IF(DR.EQ.1.OR.DR.EQ.2) THEN
          IS = DR
          WRITE(*,'(A,A6)') ' SEX IS CHANGED TO ', ISNAME(IS)
        ELSE
          CALL WSEX(IS)
        END IF
        GO TO 30
      END IF
      IF(I.EQ.32) THEN
C     CHANGE MEASURER'S NAME
        WRITE(*,'(A\)') ' TYPE IN THE NEW MEASURER''S NAME '
        READ(*,'(A18)') MNAME
        WRITE(*,'(A,A18)') ' THE MEASURER''S NAME HAS BEEN CHANGED TO '
     +   , MNAME
        GO TO 30
      END IF
      IF(I.EQ.33) THEN
C     USE A DOS COMMAND
        PAUSE
        GO TO 34
      END IF
      IF(I.EQ.34) THEN
C     JUNK THIS SUBJECT AND START OVER?
        WRITE(*,'(A\)') ' DO YOU WANT TO ERASE ALL THE DATA ENTERED FOR
     +THIS SUBJECT AND ENTER A NEW SUBJECT ?  (Y/N)'
        READ(*,'(A1)') A
        IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') GO TO 76
        GO TO 34
      END IF

      IF(I.LT.1.OR.I.GT.NVS) THEN
C     INCORRECT VARIABLE NUMBER
        WRITE(*,'(1X,A1,A,I5)')BEEP,' INCORRECT VARIABLE NUMBER =',I,BEEP
        GO TO 34
      END IF
```

```
C     CHANGE A DATA VALUE
      DATA(I,1) = DR
C     CALL SUBROUTINE MM TO CHECK IF DATA VALUE IS WITH IN RANGE
      CALL MM(I,NAME(I))
      GO TO 30
   36 WRITE(*,'(1X,A1,A/A,A1)') BEEP, 'THERE WAS A PROBLEM WITH THE VARI
     +ABLE NUMBER OR NEW VALUE ENTERED', 'CHECK AND REENTER', BEEP
C     PROBLEM READING I OR DR MAY BE A NON-NUMERIC VALUE OR WRONG
C       SEPARATOR
      GO TO 34
      ELSE
        IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') GO TO 34
C     YOU DO NOT WANT TO CHANGE A VARIABLE
      END IF
C     * * * * * *
C     CALL SUBROUTINE REGEQ FOR REGRESSION ESTIMATES OF THE DATAPOINTS
C     * * * * * *
      CALL REGEQ
      IF(ER(KLM(NVS)).LT.CK) GO TO 60
C     KLM(NVS) IS VARIABLE NUMBER OF THE VARIABLE WITH THE LARGEST ERROR
C     IF THE DATA IS WITHIN CK STANDARD ERRORS OF ESTIMATE OF THE
C       REGRESSION ESTIMATE GO TO STATEMENT #60 IF NOT REMEASURE
   40 J = NVS
   42 WRITE(*,'(1X,A1,''PLEASE REMEASURE '',A18,A1,\)')BEEP,NAME(KLM(J))
     +, BEEP
      READ(*,*,ERR=42) DR
      IF(DR.LT.0) GO TO 30
      IF(DR.LT.DATA(KLM(J),1)-RME(KLM(J)).OR.DR.GT.DATA(KLM(J),1)+
     + RME(KLM(J)) ) THEN
C     IF THE REMEASURED DATA POINT IS OUTSIDE THE RANGE OF REMEASURE
C       ERROR THE REMEASURED DATA POINT IS THE NUMBER ONE DATA POINT
C         AND THE ORIGINAL DATA POINT IS THE NUMBER TWO DATA POINT
      DATA(KLM(J),2) = DATA(KLM(J),1)
      DATA(KLM(J),1) = DR
C     CALL SUBROUTINE REGEQ FOR REGRESSION ESTIMATES OF THE DATA POINTS
      CALL REGEQ
      IF(ER(KLM(NVS)).LT.CK) GO TO 60
C     IF THE DATA IS WITHIN CK STANDARD ERRORS OF ESTIMATE OF THE
C       REGRESSION ESTIMATE GO TO STATEMENT #60 IF NOT REMEASURE
      ELSE
C     IF THE REMEASURED DATA POINT IS WITHIN THE RANGE OF REMEASURE
C       ERROR THE REMEASURED DATA POINT IS THE NUMBER TWO DATA POINT
      DATA(KLM(J),2) = DR
      END IF
C     LOOP #44 FINDS THE VARIABLE NUMBER OF THE VARIABLE THAT HAS
C       NOT BEEN REMEASURED AND HAS THE GREATEST ERROR
C         ONLY VARIABLES WITH AN ERROR OF OVER 2 WILL BE REMEASURED
      DO 44 J = NVS, 1, -1
      IF(ER(KLM(J)).GT.2.0.AND.DATA(KLM(J),2).EQ.0.0) GO TO 42
   44 CONTINUE
C   IF WE GET HERE THERE IS A PROBLEM
C     IF THIS IS THE FIRST TIME HERE MOVE THE DATA IN THE SECOND
C       ROW OF ARRAY DATA TO THE THIRD ROW OF ARRAY DATA THEN GO
C         BACK TO STATEMENT #40 AND REMEASURE FOR A SECOND TIME
```

```
      IF(DATA(KLM(NVS),3).EQ.0.0) THEN
      DO 46 I = 1, NVS
      DATA(I,3) = DATA(I,2)
   46 DATA(I,2) = 0.0
      GO TO 40
      ELSE
C     IF DATA HAS BEEN MEASURED THREE TIMES
C      REPORT THE AVERAGE, MINIMUM, & MAXIMUM
      DO 48 I = 1, NVS
      IF(DATA(I,1).GT.0.0.AND.DATA(I,2).GT.0.0.AND.DATA(I,3).GT.0.0)
     + THEN
      DM = ( DATA(I,1) + DATA(I,2) + DATA(I,3) ) / 3.0
      DMIN = AMIN1( DATA(I,1), DATA(I,2), DATA(I,3) )
      DMAX = AMAX1( DATA(I,1), DATA(I,2), DATA(I,3) )
      DATA(I,1) = DM
      DATA(I,2) = DMIN
      DATA(I,3) = DMAX
      END IF
   48 CONTINUE
C     CALL SUBROUTINE REGEQ TO GET THE FINAL REGRESSION ESTIMATES
      CALL REGEQ
      END IF
C     WRITE AN EXPLANATION
      WRITE(*,'(1X,A1,'' WRITE AN EXPLANATION PLEASE'',A1)') BEEP, BEEP
      READ(*,'(A127)') EXP
C     IF EXPLANATION EQUALS -1 THEN GO BACK TO WRITE DATA VALUES ON THE
C       SCREEN
      IF(EXP(1:2).EQ.'-1') GO TO 30
C     * * * * * *
C     THE PROGRAM HAS FINISHED ENTERING AND EDITING DATA FOR THIS SUBJECT
C     NOW THE PROGRAM WRITES OUT THE INFORMATION FOR THIS SUBJECT
C     * * * * * *
   60 CONTINUE
      DO 61 I = 1, NVS
C     CHECK FOR ZERO DATA VALUE
      IDAT(I) = DATA(I,1)
   61 IF(DATA(I,1).EQ.0.0) A = '?'
      BACKSPACE 10
C     POSITION FILE #10
   62 READ(10,'(I5)',END=63,IOSTAT=K) J
      CONTINUE
      IF(J.GT.0) GO TO 62
      BACKSPACE 10
   63 IF(K.LT.0) THEN
        BACKSPACE 10
        BACKSPACE 10
        CONTINUE
        READ(10,'(I5)') J
        CONTINUE
      END IF
C     WRITE THE DATA TO THE STATION DISK ( FILE #10 ) IN INTEGER
      WRITE(10,'(I5,I2,30I4)') NSUB, IS, ( IDAT(I), I = 1, NVS )
      CONTINUE
      BACKSPACE 10
```

```
      CONTINUE
      READ(10,'(I5,I2,30I4)') NN, ISEX, ( IDAT(I), I = 1, NVS )
C     CHECK THAT INFORMATION WAS WRITTEN
      WRITE( *,'(I5,I2,30I4)') NN, ISEX, ( IDAT(I), I = 1, NVS )
      CONTINUE
      WRITE(10,'(I5)') -13
      IF(NN.NE.NSUB) WRITE(*,'(1X,A1,A/A,A1)') BEEP, 'WRITE DOWN THE SUB
     +JECT NUMBER ON THE SCREEN AND THE CORRECT', ' SUBJECT NUMBER',
     + BEEP
      IF(A.EQ.'?') THEN
C     ASK FOR EXPLANATION OF ZERO VALUES
        WRITE(*,'(1X,A1,A,A1/A,A1)') BEEP,
     +' THERE ARE ONE OR MORE ZERO DATA VALUES', BEEP,
     +' WRITE AN EXPLANATION PLEASE', BEEP
C     IF THERE ARE BOTH LARGE REGRESSION ESTIMATE ERRORS AND ZERO VALUES
C       THE FUNCTION 3 KEY WILL RETURN THE FIRST EXPLANATION
        READ(*,'(A127)') EXP
      END IF
      WRITE(*,'(A)') ' DO YOU WANT TO WRITE A COMMENT?  (Y/N)'
      READ(*,'(A1)') A
      IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') THEN
C     WRITE A COMMENT
        WRITE(*,'(A)') ' WRITE A COMMENT PLEASE'
        READ(*,'(A127)') EXP
      END IF
C     OPEN FILE 7 AND WRITE THE DATA TO FILE 7
C     PUT SUBJECT NUMBER IN CHARACTER VARIABLE
      WRITE(SUBNUM,'(I5.5)') ISUB
C     GET THE PRINTER READY
      WRITE(*,'(A/A/A)') ' PUT THE DATA FORM IN THE PRINTER PLEASE',
     + ' CHECK THE ON-LINE LIGHT',' PRESS THE ENTER KEY WHEN READY'
      READ(*,'(A1)') A
      SR = SN
   64 WRITE(6,'(1X,4A1)',IOSTAT=IOE,ERR=65) CHAR(27), '9', CHAR(27), '0'
   65 IF(IOE.NE.0) THEN
C     PRINTER IS NOT READY
        WRITE(*,'(A1,A,A1/A)') BEEP,
     +  ' THERE IS A PROBLEM WITH THE PRINTER, IS THE PAPER IN THE PRINT
     +ER?', BEEP,
     +  ' WHEN THE PROBLEM HAS BEEN CORRECTED PRESS THE RETURN KEY'
        READ(*,'(A1)') A
        GO TO 64
      END IF
C     OPEN FILE ON SUBJECT'S DISK
      OPEN(7,FILE='B:'//SR//SUBNUM//'.'//STANUM,STATUS='NEW',ERR=80)
C     WRITE INFORMATION ON SUBJECT'S DISK
      WRITE(7,'(I6,2X,A18,I3,2X,A127)') NSUB, MNAME, IS, EXP
      DO 66 I = 1, 3
   66 WRITE(7,'(30F7.1)') ( DATA(J,I), J = 1, NVS )
      WRITE(7,'(30F5.1)') (ER(J),J=1,NVS)
      REWIND 7
C  REWIND FILE 7 TO READ IT
      READ(7,'(I6,2X,A18,I3/30(I5,2X))') NN, MN, ISEX, (IDAT(J),J=1,NVS)
      CLOSE(7)
```

```
      WRITE(6,'(A1)') ( ' ', I = 1, NLD )
C     MOVE PAPER DOWN NLD LINES TO POSITION THE PRINTER FOR PRINTING
   68 WRITE(6,'(1X,4A1,I5,A18,I3)') CHAR(27), 'x', '0', CHAR(15),
     + NN, MN, ISEX
C     PRINT INFORMATION READ FROM SUBJECT'S DISK ONTO SUBJECT'S PRINT OUT
      DO 70 I = 1, NVS, 5
   70 WRITE(6,'(5(1X,A18,I5))')
     + ( NAME(I+J), IDAT(I+J), J = 0, 4 )
      WRITE(6,'(1X,2A1)') CHAR(27), '8'
C     CLEAR THE PRINTER
      WRITE(6,'(1H1)')
   72 WRITE(*,'(A\)') ' DO YOU WANT TO PRINT THE DATA AGAIN?  (Y/N) '
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
C     THERE WAS A PROBLEM PRINTING TO THE SUBJECT'S PRINT OUT
C      PRINT IT AGAIN
        WRITE(6,'(1X,2A1)') CHAR(27), '9'
        WRITE(*,'(A/A/A)') ' PUT A CLEAN SHEET OF PAPER IN THE PRINTER',
     +  ' CHECK THE ON-LINE LIGHT', ' PRESS THE ENTER KEY WHEN READY'
        READ(*,'(A1)') A
        GO TO 68
      ELSE
        IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') GO TO 72
      END IF
C     ADD NEW MINS AND MAXS TO FILE #9
      DO 74 I = 1, NVS
C     DO NOT USE VALUES WHICH THE REGRESSION ESTIMATE IS CK OR MORE
C      STANDARD ERRORS AWAY FROM THE MEASURED VALUE
      IF(ER(I).GE.CK) GO TO 74
C     DO NOT USE ZERO VALUES
      IF(DATA(I,1).EQ.0.0) GO TO 74
C        NEW MIN
      IF(DATA(I,1).LT.MINMAX(I,1,IS)) THEN
      IREC = 0
      IF(IS.EQ.2) IREC = 60
      IREC = IREC + I
      WRITE(9,REC=IREC) DATA(I,1)
      MINMAX(I,1,IS) = DATA(I,1)
      END IF
C        NEW MAX
      IF(DATA(I,1).GT.MINMAX(I,2,IS)) THEN
      IREC = 30
      IF(IS.EQ.2) IREC = 90
      IREC = IREC + I
      WRITE(9,REC=IREC) DATA(I,1)
      MINMAX(I,2,IS) = DATA(I,1)
      END IF
   74 CONTINUE
C     ZERO DATA ARRAY
   76 DO 78 I = 1, 30
      DO 78 J = 1, 3
   78 DATA(I,J) = 0.0
C     * * * * * *
C     THE CURRENT SUBJECT IS FINISHED
```

```
      GO TO 10
C     GO BACK TO STATEMENT # 10 AND START A NEW SUBJECT
C     * * * * * *
   80 IF(SR.EQ.SN) THEN
C     FILE HAS BEEN REMEASURED CHANGE FILE NAME PREFIX
        SR = RE
        GO TO 65
      ELSE
C     FILE HAS BEEN MEASURED THREE OR MORE TIMES
        WRITE(*,'(A/A/A/A)') ' THIS SUBJECT HAS BEEN MEASURED TWICE AT T
     +HIS STATION',' TYPE A NEW TWO LETTER PREFIX',
     +  ' IF THE PROGRAM COMES BACK HERE TRY A DIFFERENT PREFIX',
     +  ' IF YOU DO NOT WANT TO SAVE THIS DATA TYPE IN 13'
        READ(*,'(A2)') SR
        IF(SR.EQ.'13') GO TO 76
        GO TO 65
      END IF
      END
      SUBROUTINE MM(I,NAME)
C     THIS SUBROUTINE CHECKS TO SEE IF A DATA POINT IS WITHIN RANGE
      CHARACTER*18 NAME, A*1, BEEP*1
      COMMON/ M / MINMAX(30,2,2), DATA(30,3), IS
      REAL MINMAX
      BEEP = CHAR(7)
C     IF WITHIN RANGE RETURN
    2 IF(DATA(I,1).GE.MINMAX(I,1,IS).AND.DATA(I,1).LE.MINMAX(I,2,IS))
     + RETURN
C     WRITE(*,'('' MIN ='',F8.1,'' MAX ='',F8.1)') MINMAX(I,1,IS),
C     + MINMAX(I,2,IS)
C     REVERSE THE SCREEN
      WRITE(*,'(1X,A1,''[0;7m'')') CHAR(27)
    4  WRITE(*,'('' THIS VALUE IS NOT IN RANGE!'',A1/
     +  '' PLEASE CHECK THE INPUTTED VALUE FOR '',A18,A1/
     +  '' IF YOU WANT TO CHANGE THE VALUE TYPE IN C OR *'',A1/
     +  '' IF THE SUBJECT IS SMALL TYPE IN S OR -'',A1/
     +  '' IF THE SUBJECT IS LARGE TYPE IN L OR + '',A1,\)')
     + BEEP,NAME,BEEP,BEEP,BEEP,BEEP
       READ(*,'(A1)') A
C     RESET THE SCREEN
      WRITE(*,'(1X,A1,''[0;1m'')') CHAR(27)
C     DATA TOO FAR OUT OF RANGE
      IF(DATA(I,1).LT.(0.75*MINMAX(I,1,IS)).OR.
     + DATA(I,1).GT.(1.25*MINMAX(I,2,IS))) THEN
       A='C'
       WRITE(*,'(1X,A1,A18,'' = '',F8.1,A/1X,A,A1)')
     + BEEP ,NAME,DATA(I,1),'  THIS IS TOO FAR OUT OF RANGE',
     +  'PLEASE CHECK AND REENTER',BEEP
      END IF
C     SMALL DATA VALUE
       IF((A.EQ.'S'.OR.A.EQ.'s'.OR.A.EQ.'-').AND.
     + DATA(I,1).LT.MINMAX(I,2,IS)) RETURN
C     LARGE DATA VALUE
       IF((A.EQ.'L'.OR.A.EQ.'l'.OR.A.EQ.'+').AND.
     + DATA(I,1).GT.MINMAX(I,2,IS)) RETURN
```

```
C     READ NEW DATA VALUE
        IF(A.EQ.'C'.OR.A.EQ.'c'.OR.A.EQ.'*') THEN
         WRITE(*,'(1X,A18,'' = '',\)') NAME
        READ(*,*) DATA(I,1)
        GO TO 2
        END IF
         GO TO 4
       END
       SUBROUTINE REGEQ
C     THIS SUBROUTINE ESTIMATES EACH DATA POINT AND COMPARES
C       THE ESTIMATED AND MEASURED VALUES
       COMMON/ M / MINMAX(30,2,2), DATA(30,3), IS
       REAL MINMAX
       COMMON/ REG / EST(30), IVN(30,2,2), EQC(30,3,2), SE(30,2),
      + ER(0:30), KLM(0:30), NVS
       DO 2 I = 1, NVS
C     REGRESSION ESTIMATE
       EST(I) = DATA(IVN(I,1,IS),1) * EQC(I,1,IS) + DATA(IVN(I,2,IS),1) *
      + EQC(I,2,IS) + EQC(I,3,IS)
C     DIFFERENCE BETWEEN ESTIMATED AND MEASURED DATA DIVIDED BY THE
C       STANDARD ERROR OF ESTIMATE
       ER(I) = ( DATA(I,1) - EST(I) ) / SE(I,IS)
C     CHECK FOR ZERO VALUES
C     SET ER(I) TO 0.0 IF ANY OF THE VARIABLES IN THE EQUATION IS ZERO
       IF(DATA(I,1).EQ.0.0) ER(I) = 0.0
       IF(DATA(IVN(I,1,IS),1).EQ.0.0) ER(I) = 0.0
       IF(DATA(IVN(I,2,IS),1).EQ.0.0) ER(I) = 0.0
     2 IF(ER(I).LT.0.0) ER(I) = -ER(I)
       ER(0) = 0.0
       KLM(0) = 0
       KLM(1) = 1
C     ORDER BY SIZE OF ER(L), SMALLEST ER(1), LARGEST ER(NVS)
       DO 8 L = 2, NVS
       LL = L - 1
       GO TO 6
     4 KLM(LL+1) = KLM(LL)
       LL = LL - 1
     6 IF(ER(L).LT.ER(KLM(LL))) GO TO 4
     8 KLM(LL+1) = L
       WRITE(*,'(5F16.2)') ( ER(I), I = 1, NVS )
       RETURN
       END
       SUBROUTINE APPEND(NVS)
C     THIS SUBROUTINE READS TO THE END OF THE STATION DISK
C       THEN WRITES OUT THE LAST SUBJECT'S DATA
C         THE STATION FILE IS READY FOR NEW SUBJECT'S DATA
       DIMENSION IDAT(30)
       WRITE(*,'(A/A)') ' SUBROUTINE APPEND HAS BEEN CALLED',
      + ' THE SUBJECTS MEASURED PREVIOUSLY'
     2 READ(10,'(I5)',END=6) I
       WRITE(*,'(I10)') I
       IF(I.GT.0) GO TO 2
     4 BACKSPACE 10
       BACKSPACE 10
```

```
      READ(10,'(I5,I2,30I4)') NSUB, IS, ( IDAT(I), I = 1, NVS )
      WRITE(*,'(A)') ' THE LAST SUBJECT''S DATA'
      WRITE(*,'(16I5)') NSUB, IS, ( IDAT(I), I = 1, NVS )
      RETURN
    6 WRITE(*,'(A)') ' THE END OF THE FILE WAS REACHED'
      GO TO 4
      END
      SUBROUTINE THREE
C     SUBROUTINE THREE IS CALLED TO MEASURE THE FIRST THREE VARIABLES
C       AT STATION #2
C     THESE THREE VARIABLES ARE MEASURED THREE TIMES THEN PUT IN ORDER
C       OF SIZE THEN THE MIDDLE VALUE IS SELECTED IF THE DIFFERENCES
C       ARE WITH IN THE REMEASURE ERROR
C     IF ONE DIFFERENCE IS GREATER THAN THE REMEASURE DIFFERENCE THEN
C       THE OTHER TWO VALUES ARE AVERAGED
C     IF BOTH DIFFERENCES ARE GREATER THAN THE REMEASURE DIFFERENCE
C       THEN EVERY THING IS REMEASURED
      COMMON/ M / MINMAX(30,2,2), DATA(30,3), IS
      COMMON/ T / RME, NAME
      DIMENSION X(3,3), RME(30)
      CHARACTER*18 NAME(30)
    2 DO 4 J = 1, 3
C     ALTERNATE BETWEEN THE FIRST TWO VARIABLES THREE TIMES
        DO 4 I = 1, 2
          WRITE(*,'('' PLEASE MEASURE #'',I3,1X,A18,''   '',\)')I,
     +     NAME(I)
    4     READ(*,*,ERR=2) X(J,I)
    6 WRITE(*,'('' PLEASE MEASURE #'',I3,1X,A18,2X/
     + '' THREE TIMES, PUT A SPACE IN BETWEEN EACH MEASUREMENT'')')
     + I, NAME(3)
C     MEASURE VARIABLE THREE TIMES
      READ(*,*,ERR=6) X(1,3), X(2,3), X(3,3)
      DO 10 I = 1, 3
C     VALUES ARE PUT IN ORDER OF SIZE
      X1 = X(1,I)
      X2 = X(2,I)
      X3 = X(3,I)
    8 IF(X1.GT.X2) THEN
        XX = X1
        X1 = X2
        X2 = XX
      END IF
      IF(X2.GT.X3) THEN
        XX = X2
        X2 = X3
        X3 = XX
      END IF
      IF(X1.GT.X2) GO TO 8
      DATA(I,1) = 0.0
C     IF THE DIFFERENCES ARE WITH IN REMEASURE ERROR SELECT THE
C       MIDDLE VALUE
      IF(X2-X1.LE.RME(I).AND.X3-X2.LE.RME(I)) DATA(I,1) = X2
C     IF THE SMALLEST VALUE IS OUTSIDE THE REMEASURE RANGE AVERAGE
C       THE OTHER TWO
```

```
      IF(X2-X1.GT.RME(I).AND.X3-X2.LE.RME(I)) DATA(I,1) = (X2+X3) / 2.0
C     IF THE LARGEST VALUE IS OUTSIDE THE REMEASURE RANGE AVERAGE
C       THE OTHER TWO
      IF(X2-X1.LE.RME(I).AND.X3-X2.GT.RME(I)) DATA(I,1) = (X1+X2) / 2.0
   10 CONTINUE
      DO 12 I = 1, 3
C     IF BOTH THE LARGEST AND SMALLEST VALUES FOR ANY ONE OF THE
C       THREE VARIABLES WERE OUTSIDE THE REMEASURE RANGE THEN
C         REMEASURE ALL THREE VARIABLES
      IF(DATA(I,1).EQ.0.0) THEN
        WRITE(*,'(1X,A18,\)') NAME(I)
        WRITE(*,'(A)') ' THERE IS TOO LARGE A RANGE BETWEEN THESE THR
     +EE MEASUREMENTS'
        GO TO 2
      END IF
C     CALL SUBROUTINE MM TO CHECK THAT THE DATA VALUES ARE WITH IN RANGE
      CALL MM( I, NAME(I) )
   12 CONTINUE
      RETURN
      END
      SUBROUTINE WSEX(IS)
C     SUBROUTINE WSEX ASKS WHICH SEX IS TO BE MEASURED
C       THEN CHECKS FOR A CORRECT RESPONSE
      CHARACTER*1 SEX
    2 WRITE(*,'('' WHAT SEX IS TO BE MEASURED? (M/F) '',\)')
      READ(*,'(A1)') SEX
      IF(SEX.EQ.'M'.OR.SEX.EQ.'m') THEN
       WRITE(*,'('' WE NOW MEASURE MEN'')')
       IS = 1
      ELSE
       IF(SEX.EQ.'F'.OR.SEX.EQ.'f') THEN
        WRITE(*,'('' WE NOW MEASURE WOMEN'')')
        IS = 2
       ELSE
        WRITE(*,'('' ONE MORE TIME'')')
        GO TO 2
       END IF
      END IF
      RETURN
      END
```

```
C     THIS PROGRAM IS A COMBINATION OF  SEVERAL PROGRAMS
C     SO THAT STATION 8 CAN IN-PROCESS, MEASURE FEET, OUT-PROCESS,
C     OR ENTER QUESTIONNAIRE DATA WITHOUT CHANGING PROGRAMS
      CHARACTER*3 ESC, ANS*1, SEX*1, BEEP*1
      LOGICAL THERE
      BEEP = CHAR(7)
      ESC = ' '//CHAR(27)//'['
C     INITIALIZE SUBROUTINE INED
      CALL START1
C     INITIALIZE SUBROUTINE DISKIN
      CALL START3
C     THE DISK WITH THE PROGRAM ON IT IS REMOVED AND REPLACED WITH A DISK
C      TO RECEIVE THE WEEKS DATA
      WRITE(*,'(A3,A2/3X,A/3X,A/3X,A)') ESC, '2J',
     + 'REMOVE THE PROGRAM DISK PLEASE', 'PUT THIS WEEK''S DATA DISK IN
     +DRIVE A PLEASE', 'PRESS THE ENTER KEY PLEASE'
      READ(*,'(A1)') ANS
C     CLEAR THE SCREEN AND WRITE THE MENU
    2 WRITE(*,'(A3,A2/8(3X,A/),3X,A\)') ESC, '2J', '1  IN PROCESS',
     + '2  MEASURE FEET', '3  OUT PROCESS', '4  ENTER QUESTIONNAIRES',
     + '5  COPY QUESTIONNAIRES', '6  USE DOS SYSTEM', '7  END PROGRAM',
     + '8  WRITE SUBJECT''S FILE TO SCREEN', 'WHAT DO YOU NEED TO DO?  '
C     READ SELECTION
      READ(*,'(A1)') ANS
C     CHECK SELECTION
      IF(ANS.LT.'1'.OR.ANS.GT.'8') THEN
        WRITE(*,'(1X,A1)') BEEP
        GO TO 2
      END IF
C     CONVERT INTO INTEGER
      READ(ANS,'(I1)') I
C     GO TO THE LINE APPROPRIATE FOR THE PROCEDURE SELECTED
      GO TO ( 10, 20, 30, 40, 50, 60, 70, 80 ), I
C     * * * * IN PROCESSING * * * *
   10 WRITE(*,'(A3,A2/3X,A/)') ESC, '2J', 'PUT A FORMATTED DISK IN DRIVE
     +B PLEASE'
C     IN PROCESSING ASKS FOR THE SUBJECT NUMBER, SEX, ACTUAL HEIGHT, AND
C      ACTUAL WEIGHT
C     IF AN ENTRY IS LESS THAN ZERO IN PROCESSING STOPS AND RETURNS TO
C      THE MAIN MENU
   12 WRITE(*,'(3X,A\)') 'TYPE IN THE SUBJECT''S NUMBER  '
      READ(*,*,ERR=12) NSUB
      IF(NSUB.LT.0) GO TO 2
   14 WRITE(*,'(3X,A\)') 'TYPE IN THE SUBJECT''S SEX  (M/F)  '
      READ(*,'(A1)') SEX
      IF(SEX.EQ.'-') GO TO 2
      IF(SEX.NE.'F'.AND.SEX.NE.'M'.AND.SEX.NE.'f'.AND.SEX.NE.'m')GOTO 14
   15 WRITE(*,'(3X,A\)') 'TYPE IN THE SUBJECT''S ACTUAL HEIGHT  '
      READ(*,*,ERR=15) HGT
      IF(HGT.LT.0) GO TO 2
   16 WRITE(*,'(3X,A\)') 'TYPE IN THE SUBJECT''S ACTUAL WEIGHT  '
      READ(*,*,ERR=16) WGT
```

```
      IF(WGT.LT.0) GO TO 2
C     WRITE INPUT VALUES TO SCREEN
   17 WRITE(*,'(3X,A,I7/3X,A,A1/3X,A,F8.1/3X,A,F8.1/3X,A/3X,A\)')
     + 'NSUB =', NSUB, 'SEX = ', SEX, 'ACTUAL HEIGHT =', HGT,
     + 'ACTUAL WEIGHT =', WGT, 'TYPE E OR * TO EXIT TO MENU',
     + 'IS ALL THE ABOVE INFORMATION CORRECT?  (Y/N)  '
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
C     INPUT VALUES OK
      JUNK = 0
C     CHECK TO SEE IF FILE SUBJECT.NUM ALREADY EXISTS
   18   INQUIRE(FILE='B:SUBJECT.NUM',EXIST=THERE)
      IF(JUNK.NE.0) THEN
C     PROBLEM WITH DISK  REPLACE DISK AND TRY AGAIN
      WRITE(*,'(1X,A1)') BEEP
      WRITE(*,'(3X,A/3X,A/3X,A)') 'THERE IS A PROBLEM WITH THIS DISK',
     + 'USE A DIFFERENT DISK PLEASE', 'PRESS THE ENTER KEY WHEN THE DISK
     + HAS BEEN REPLACED'
      READ(*,'(A1)') ANS
      JUNK = 0
        IF(ANS.EQ.'-') GO TO 10
      GO TO 18
      END IF
        IF(THERE) THEN
C     DISK SHOULD BE EMPTY BUT IT IS NOT WRITE MESSAGE TO THE SCREEN
          WRITE(*,'(3X,A)') 'THERE IS A SUBJECT.NUM FILE ON THIS DISK'
        JUNK = 1
        GO TO 18
        ELSE
C     DISK IS EMPTY OPEN FILE
        OPEN(10,FILE='B:SUBJECT.NUM',STATUS='NEW',ERR=17,IOSTAT=JUNK)
        END IF
C     WRITE FILE SUBJECT.NUM ON SUBJECT'S DISK
        WRITE(10,'(I7/A1,2F8.1)') NSUB, SEX, HGT, WGT
        CLOSE(10)
      ELSE
C     REENTER DATA VALUES
        IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') GO TO 12
C     STOP IN PROCESSING AND RETURN TO MENU
        IF(ANS.EQ.'E'.OR.ANS.EQ.'e'.OR.ANS.EQ.'*') GO TO 2
C     AN INCORRECT RESPONSE WAS GIVEN WRITE THE INPUT VALUES AGAIN
        WRITE(*,'(1X,A1)') BEEP
        GO TO 17
      END IF
   19 WRITE(*,'(3X,A/3X,A\)') 'PLEASE REMOVE SUBJECT''S DISK FROM DRIVE
     +B', 'CONTINUE IN PROCESSING?  (Y/N)  '
      READ(*,'(A1)') ANS
C     CONTINUE IN PROCESSING
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 10
C     STOP IN PROCESSING AND RETURN TO THE MAIN MENU
      IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') GO TO 2
      WRITE(*,'(1X,A1)') BEEP
      GO TO 19
   20 CALL INED
```

```
C     CALL SUBROUTINE INED TO ENTER THE THREE FOOT DIMENSIONS
      GO TO 2
   30 CALL DISKIN
C     CALL SUBROUTINE DISKIN TO OUT PROCESS
      GO TO 2
   40 CALL QESIN
C     CALL SUBROUTINE QESIN TO ENTER QUESTIONNAIRE DATA
      GO TO 2
   50 CALL COPY
C     DUMMY SUBROUTINE
      GO TO 2
   60 PAUSE
C     USE DOS SYSTEM
      WRITE(*,'(3X,A)') 'PRESS THE ENTER KEY TO RETURN TO THE MENU'
      READ(*,'(A1)') ANS
      GO TO 2
   70 WRITE(*,'(3X,A)') 'DO YOU WANT TO END THE PROGRAM  (Y/N)  '
C     END PROGRAM INOUT8
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'y'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'+') THEN
C     IF RESPONSE IS Y THEN END PROGRAM ELSE RETURN TO MAIN MENU
        CLOSE(4)
        STOP
      ELSE
        WRITE(*,'(1X,A1)') BEEP
        GO TO 2
      END IF
   80 CALL READSML
C     CALL SUBROUTINE READSML TO READ THE FILES ON THE DATA DISK
C     THERE IS A FILE FOR EACH SUBJECT MEASURED THIS WEEK
      GO TO 2
      END
      SUBROUTINE COPY
C     DUMMY SUBROUTINE
      WRITE(*,'(3X,A)') 'SORRY SUBROUTINE COPY IS NOT READY'
      WRITE(*,'(1X,A1)') CHAR(7)
      DO 2 I = 1, 10000
C     WASTE TIME
    2 CONTINUE
      RETURN
      END
      SUBROUTINE INED
C     SUBROUTINE INED IS A DATA ENTRY ( IN ) AND DATA EDITING ( ED )
C      SUBROUTINE FOR THE THREE FOOT DIMENSIONS
      COMMON/ M / MINMAX(3,2,2), DATA(3,3), IS
      REAL MINMAX
      COMMON/ REG / EST(3), IVN(3,2,2), EQC(3,3,2), SE(3,2),
     + ER(0:3) ,KLM(0:3), NVS
      COMMON/ T / RME, NAME
      CHARACTER*18 NAME(3), MNAME, MN, A*1, BEEP*1, EXP*127,
     + STANUM*1, SUBNUM*5, SN*2, RE*2, SR*2, ISNAME(2)*6
      DIMENSION DAT(3), RME(3), IDAT(3)
      DATA NST / 1 /,MNAME /'    '/, ISNAME / ' MALE ', 'FEMALE' /
      STANUM = '8'
```

```
      GO TO 1
      ENTRY START1
C     READ FILE REGEQ TO GET
      OPEN(9,FILE='A:REGEQ',STATUS='OLD')
C     STATION NUMBER, NUMBER OF VARIABLES, WHERE TO PRINT OUTPUT ON PAGE,
C      CHECK VALUE
      READ(9,*) NSTA, NVS, NLD, CK
C     REMEASURE ERROR,
      READ(9,*) (RME(I),I = 1, NVS)
C     NAMES, MALE REGRESSION EQUATIONS,
      READ(9,'(A18,2I4,4F15.8)') (NAME(I), IVN(I,1,1), IVN(I,2,1),
     + (EQC(I,J,1), J = 1, 3), SE(I,1), I = 1, NVS)
C     AND FEMALE REGRESSION EQUATIONS.
      READ(9,'(18X,2I4,4F15.8)') (IVN(I,1,2), IVN(I,2,2), (EQC(I,J,2),
     + J = 1, 3), SE(I,2), I = 1, NVS)
      CLOSE(9)
      OPEN(9,FILE='A:MINMAX',STATUS='OLD')
      READ(9,*) MINMAX
      CLOSE(9)
      RETURN
    1 DO 2 I = NVS
    2 RME(I) = MIN( RME(I), SE(I,1), SE(I,2) ) / 2.0
      WRITE(*,'('' GOOD MORNING IT IS TIME TO MEASURE SOME SOLDIERS
     +'')')
C     INITIALIZE VARIABLES AND OPEN FILES
      CALL WSEX(IS)
      BEEP = CHAR(7)
      SN = 'SN'
      RE = 'RE'
      OPEN(6,FILE='PRN')
C     ******
C        THE SUBROUTINE COMES TO HERE AT THE START OF EACH SUBJECT
C     ******
   10 IF(IS.EQ.1) WRITE(*,'(A)') ' WE ARE MEASURING MEN'
      IF(IS.EQ.2) WRITE(*,'(A)') ' WE ARE MEASURING WOMEN'
      WRITE(*,'(A,A18)') ' THE MEASURER''S NAME IS ', MNAME
      WRITE(*,'(35H CHANGE THE MEASURER''S NAME? (Y/N) ,\)')
C     CHANGE THE MEASURER'S NAME?
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.MNAME.EQ.'   '.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
   12 WRITE(*,'(27H ENTER THE MEASURER''S NAME  ,\)')
      READ(*,'(A18)') MNAME
      WRITE(*,'(9X,A18)') MNAME
      IF(MNAME.EQ.'END'.OR.MNAME.EQ.'end') RETURN
C     CHANGE SEX BEING MEASURED
      IF(MNAME.EQ.'SEX'.OR.MNAME.EQ.'sex') THEN
        CALL WSEX(IS)
        GO TO 12
      END IF
C     USE DOS COMMAND
      IF(MNAME.EQ.'SYSTEM'.OR.MNAME.EQ.'system') THEN
        PAUSE
        GO TO 12
      END IF
```

```
      ELSE
      IF(A.NE.'N'.AND.A.NE.'-'.AND.A.NE.'n') GO TO 10
      END IF
C   READ SUBJECT NUMBER FROM SCREEN
      WRITE(*,'('' TYPE IN THE SUBJECT NUMBER '',\)')
      READ(*,*,ERR=10) NSUB
      OPEN(8,FILE='B:SUBJECT.NUM',STATUS='OLD',IOSTAT=ISTAT,ERR=14)
C   READ SUBJECT NUMBER FROM SUBJECT'S DISK
      READ(8,*,IOSTAT=ISTAT,ERR=14) ISUB
      CLOSE(8)
   14 IF(ISTAT.LT.0) THEN
C   PROBLEM ON SUBJECT'S DISK - END OF FILE
        WRITE(*,'(1X,A1,A,A1)') BEEP, ' FILE SUBJECT.NUM IS BLANK',BEEP
        GO TO 10
      END IF
      IF(ISTAT.GT.0) THEN
C   PROBLEM ON SUBJECT'S DISK - ERROR
        WRITE(*,'(1X,A1,A,A1)') BEEP, ' FILE SUBJECT.NUM IS NOT ON THIS
     +DISK OR THERE IS SOME OTHER PROBLEM', BEEP
        GO TO 10
      END IF
      IF(NSUB.NE.ISUB) THEN
C   SUBJECT NUMBERS DO NOT MATCH
      WRITE(*,'(1X,A1,61H THERE IS SOME CONFUSION ABOUT THE SUBJECT''S
     +SUBJECT NUMBER ,A1)') BEEP, BEEP
      WRITE(*,'(A,I6,/A,I6)') ' THE SUBJECT NUMBER ENTERED =', NSUB,
     + ' THE SUBJECT NUMBER FROM THE DISK =', ISUB
       GO TO 10
      END IF
C   * * * * * *
C    ENTER THE SUBJECT'S DATA
C   * * * * * *
      EXP = 'OK'
      II = 1
   20 CONTINUE
C    LOOP #26 IS THE MAIN DATA ENTRY AREA
      DO 26 I = II, NVS
       WRITE(*,'('' PLEASE MEASURE #'',I3,1X,A18,''   '',\)') I, NAME(I)
C    A NOT REAL ENTRY WILL SEND THE PROGRAM BACK TO 'CHANGE THE
C      MEASURER'S NAME' PROMPT
       READ(*,*,ERR=10) DATA(I,1)
      IF(DATA(I,1).LT.0.0) THEN
C   IF DATA VALUE IS LESS THAN ZERO, RESTART THE MEASURING SEQUENCE AT
C     A PREVIOUS VARIABLE
   22  WRITE(*,'(A,\)')' RESTART MEASURING SEQUENCE AT VARIABLE NUMBER '
       READ(*,*,ERR=24) II
C   IF VARIABLE NUMBER IS LESS THAN ZERO, RESTART SUBJECT
        IF(II.LT.0) GO TO 10
C   IF VARIABLE NUMBER IS TOO LARGE, RESTART MEASURING AT CURRENT
C     VARIABLE
        IF(II.GT.I) II = I
C   SET ZERO TO ONE
        IF(II.LT.1) II = 1
        GO TO 20
```

```
C   INCORRECT VARIABLE NUMBER ENTERED
   24   WRITE(*,'(1X,A1,A/A,A1)') BEEP, 'THERE WAS A PROBLEM WITH THE VA
     +RIABLE NUMBER ENTERED','CHECK AND REENTER', BEEP
        GO TO 22
       END IF
       IF(DATA(I,1).EQ.0.0) THEN
C   CHECK IF ZERO WAS INTENDED DATA VALUE
          WRITE(*,'(9X,A1,A18,A,A1)') BEEP, NAME(I),
     +    ' IS ZERO!! IS THAT CORRECT?  (Y/N)', BEEP
          READ(*,'(A1)') A
            IF(A.NE.'Y'.AND.A.NE.'y'.AND.A.NE.'+') THEN
              II = I
              GO TO 20
          END IF
       END IF
        IF(DATA(I,1).GT.0.0) CALL MM(I,NAME(I))
   26 CONTINUE
C  CLEAR THE SCREEN
C    THE DATA POINTS ARE WRITTEN TO THE SCREEN
C      THE DATA POINTS CAN BE CHANGED IF NEEDED
   30 WRITE(*,'(1X,A1,''[2J'')') CHAR(27)
      DO 32 I = 1, NVS, 2
   32 WRITE(*,'(2(5X,I3,2X,A18,'' ='',F7.1))') I, NAME(I), DATA(I,1),
     + I + 1, NAME( I + 1 ), DATA( I + 1, 1 )
      WRITE(*,'(A,I1,2X,A6,A,A18,A)') ' 31 SEX = ', IS, ISNAME(IS),
     + ' 32 MEASURER''S NAME = ', MNAME, ' 33 SYSTEM  34 ERASE'
   34 WRITE(*,'('' DO YOU WANT TO CHANGE ANY VALUES? (Y/N) '',\)')
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
C   SO YOU WANT TO CHANGE A VARIABLE
         WRITE(*,'(A,A)') ' FOR VARIABLE NUMBERS, 32 MEASURER''S NAME,'
     + , ' 33 SYSTEM, & 34 ERASE USE 1 AS THE NEW VALUE'
        WRITE(*,'('' TYPE IN THE VARIABLE NUMBER AND THE NEW VALUE WITH A
     + SPACE IN BETWEEN '',\)')
C   READ IN VARIABLE NUMBER AND NEW DATA VALUE
        READ(*,*,ERR=36) I, DR
        IF(I.EQ.31) THEN
C   CHANGE SEX MEASURED
           IF(DR.EQ.1.OR.DR.EQ.2) THEN
             IS = DR
             WRITE(*,'(A,A6)') ' SEX IS CHANGED TO ', ISNAME(IS)
           ELSE
             CALL WSEX(IS)
           END IF
           GO TO 30
        END IF
        IF(I.EQ.32) THEN
C   CHANGE MEASURER'S NAME
           WRITE(*,'(A\)') ' TYPE IN THE NEW MEASURER''S NAME '
           READ(*,'(A18)') MNAME
           WRITE(*,'(A,A18)') ' THE MEASURER''S NAME HAS BEEN CHANGED TO '
     +    , MNAME
           GO TO 30
        END IF
```

```
      IF(I.EQ.33) THEN
C   USE A DOS COMMAND
        PAUSE
        GO TO 34
      END IF
      IF(I.EQ.34) THEN
C   JUNK THIS SUBJECT AND START OVER?
        WRITE(*,'(A\)') ' DO YOU WANT TO ERASE ALL THE DATA ENTERED FOR
     +THIS SUBJECT AND ENTER A NEW SUBJECT ?  (Y/N)'
        READ(*,'(A1)') A
        IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') GO TO 76
        GO TO 34
      END IF

      IF(I.LT.1.OR.I.GT.NVS) THEN
C   INCORRECT VARIABLE NUMBER
        WRITE(*,'(1X,A1,A,I5)')BEEP,' INCORRECT VARIABLE NUMBER =',I,BEEP
        GO TO 34
      END IF
C   CHANGE A DATA VALUE
      DATA(I,1) = DR
C   CALL SUBROUTINE MM TO CHECK IF DATA VALUE IS WITHIN RANGE
      CALL MM(I,NAME(I))
      GO TO 30
   36 WRITE(*,'(1X,A1,A/A,A1)') BEEP, 'THERE WAS A PROBLEM WITH THE VARI
     +ABLE NUMBER OR NEW VALUE ENTERED', 'CHECK AND REENTER', BEEP
C   PROBLEM READING I OR DR, MAY BE A NON-NUMERIC VALUE OR WRONG
C     SEPARATOR
      GO TO 34
      ELSE
        IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') GO TO 34
C   YOU DO NOT WANT TO CHANGE A VARIABLE
      END IF
C   * * * * * *
C    CALL SUBROUTINE REGEQ FOR REGRESSION ESTIMATES OF THE DATAPOINTS
C   * * * * * *
      CALL REGEQ
      IF(ER(KLM(NVS)).LT.CK) GO TO 60
C    KLM(NVS) IS VARIABLE NUMBER OF THE VARIABLE WITH THE LARGEST ERROR
C    IF THE DATA IS WITHIN CK STANDARD ERRORS OF ESTIMATE OF THE
C      REGRESSION ESTIMATE GO TO STATEMENT #60 IF NOT REMEASURE
   40 J = NVS
   42 WRITE(*,'(1X,A1,''PLEASE REMEASURE '',A18,A1,\)')BEEP,NAME(KLM(J))
     +, BEEP
      READ(*,*,ERR=42) DR
      IF(DR.LT.0) GO TO 30
      IF(DR.LT.DATA(KLM(J),1)-RME(KLM(J)).OR.DR.GT.DATA(KLM(J),1)+
     + RME(KLM(J)) ) THEN
C    IF THE REMEASURED DATA POINT IS OUTSIDE THE RANGE OF REMEASURE
C      ERROR THE REMEASURED DATA POINT IS THE NUMBER ONE DATA POINT
C        AND THE ORIGINAL DATA POINT IS THE NUMBER TWO DATA POINT
      DATA(KLM(J),2) = DATA(KLM(J),1)
      DATA(KLM(J),1) = DR
C    CALL SUBROUTINE REGEQ FOR REGRESSION ESTIMATES OF THE DATAPOINTS
```

```
      CALL REGEQ
      IF(ER(KLM(NVS)).LT.CK) GO TO 60
C     IF THE DATA IS WITHIN CK STANDARD ERRORS OF ESTIMATE OF THE
C       REGRESSION ESTIMATE GO TO STATEMENT #60 IF NOT REMEASURE
      ELSE
C     IF THE REMEASURED DATA POINT IS WITHIN THE RANGE OF REMEASURE
C       ERROR THE REMEASURED DATA POINT IS THE NUMBER TWO DATA POINT
      DATA(KLM(J),2) = DR
      END IF
C     LOOP #44 FINDS THE VARIABLE NUMBER OF THE VARIABLE THAT HAS
C       NOT BEEN REMEASURED AND HAS THE GREATEST ERROR
C         ONLY VARIABLES WITH AN ERROR OF OVER 2 WILL BE REMEASURED
      DO 44 J = NVS, 1, -1
      IF(ER(KLM(J)).GT.2.0.AND.DATA(KLM(J),2).EQ.0.0) GO TO 42
   44 CONTINUE
C   IF WE GET HERE THERE IS A PROBLEM
C     IF THIS IS THE FIRST TIME HERE, MOVE THE DATA IN THE SECOND
C       ROW OF ARRAY DATA TO THE THIRD ROW OF ARRAY DATA THEN GO BACK
C         TO STATEMENT #40 AND REMEASURE FOR A SECOND TIME
      IF(DATA(KLM(NVS),3).EQ.0.0) THEN
      DO 46 I = 1, NVS
      DATA(I,3) = DATA(I,2)
   46 DATA(I,2) = 0.0
      GO TO 40
      ELSE
C     IF DATA HAS BEEN MEASURED THREE TIMES
C       REPORT THE AVERAGE, MINIMUM, & MAXIMUM
      DO 48 I = 1, NVS
      IF(DATA(I,1).GT.0.0.AND.DATA(I,2).GT.0.0.AND.DATA(I,3).GT.0.0)
     + THEN
      DM = ( DATA(I,1) + DATA(I,2) + DATA(I,3) ) / 3.0
      DMIN = AMIN1( DATA(I,1), DATA(I,2), DATA(I,3) )
      DMAX = AMAX1( DATA(I,1), DATA(I,2), DATA(I,3) )
      DATA(I,1) = DM
      DATA(I,2) = DMIN
      DATA(I,3) = DMAX
      END IF
   48 CONTINUE
C    CALL SUBROUTINE REGEQ TO GET THE FINAL REGRESSION ESTIMATES
      CALL REGEQ
      END IF
C    WRITE AN EXPLANATION
      WRITE(*,'(1X,A1,'' WRITE AN EXPLANATION PLEASE'',A1)') BEEP, BEEP
      READ(*,'(A127)') EXP
C    IF EXPLANATION EQUALS -1 THEN GO BACK TO WRITE DATA VALUES ON THE
C      SCREEN
      IF(EXP(1:2).EQ.'-1') GO TO 30
C    * * * * * *
C    THE SUBROUTINE HAS FINISHED ENTERING AND EDITING DATA FOR THIS
C      SUBJECT.  NOW THE SUBROUTINE WRITES OUT THE INFORMATION FOR THIS
C      SUBJECT
C    * * * * * *
   60 CONTINUE
C    WRITE A COMMENT
```

```
      WRITE(*,'(A)') ' DO YOU WANT TO WRITE A COMMENT?  (Y/N)'
      READ(*,'(A1)') A
      IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') THEN
        WRITE(*,'(A)') ' WRITE A COMMENT PLEASE'
        READ(*,'(A127)') EXP
      END IF
C     OPEN FILE 7 AND WRITE THE DATA TO FILE 7
C     PUT SUBJECT NUMBER IN CHARACTER VARIABLE
      WRITE(SUBNUM,'(I5.5)') ISUB
C    GET PRINTER READY
      WRITE(*,'(A/A/A)') ' PUT THE DATA FORM IN THE PRINTER PLEASE',
     + ' CHECK THE ON-LINE LIGHT',' PRESS THE ENTER KEY WHEN READY'
      READ(*,'(A1)') A
      SR = SN
   64 WRITE(6,'(1X,4A1)',IOSTAT=IOE,ERR=65) CHAR(27), '9', CHAR(27), '0'
   65 IF(IOE.NE.0) THEN
C    PRINTER NOT READY
        WRITE(*,'(A1,A,A1/A)') BEEP,
     +  ' THERE IS A PROBLEM WITH THE PRINTER, IS THE PAPER IN THE PRINT
     +ER?', BEEP,
     +  ' WHEN THE PROBLEM HAS BEEN CORRECTED PRESS THE RETURN KEY'
        READ(*,'(A1)') A
        GO TO 64
      END IF
C    OPEN FILE ON SUBJECT'S DISK
      OPEN(7,FILE='B:'//SR//SUBNUM//'.'//STANUM,STATUS='NEW',ERR=80)
C    WRITE INFORMATION ON SUBJECT'S DISK
      WRITE(7,'(I6,2X,A18,I3,2X,A127)') NSUB, MNAME, IS, EXP
      DO 66 I = 1, 3
   66 WRITE(7,'(30F7.1)') ( DATA(J,I), J = 1, NVS )
      WRITE(7,'(30F5.1)') (ER(J),J=1,NVS)
      REWIND 7
C   REWIND FILE 7 TO READ IT
      READ(7,'(I6,2X,A18,I3/30(I5,2X))') NN, MN, ISEX, (IDAT(J),J=1,NVS)
      CLOSE(7)
      WRITE(6,'(A1)') ( ' ', I = 1, NLD )
C    MOVE PAPER DOWN NLD LINES TO POSITION THE PRINTER FOR PRINTING
   68 WRITE(6,'(1X,4A1,I5,A18,I3)') CHAR(27), 'x', '0', CHAR(15),
     + NN, MN, ISEX
C    PRINT INFORMATION READ FROM SUBJECT'S DISK ONTO SUBJECT'S PRINT OUT
      WRITE(6,'(5(1X,A18,I5))') ( NAME(J), IDAT(J), J = 1, 3 )
      WRITE(6,'(1X,2A1)') CHAR(27), '8'
C    CLEAR THE PRINTER
      WRITE(6,'(1H1)')
   72 WRITE(*,'(A\)') ' DO YOU WANT TO PRINT THE DATA AGAIN?  (Y/N) '
      READ(*,'(A1)') A
      IF(A.EQ.'Y'.OR.A.EQ.'y'.OR.A.EQ.'+') THEN
C    THERE WAS A PROBLEM PRINTING TO THE SUBJECT'S PRINT OUT
C       PRINT IT AGAIN
        WRITE(6,'(1X,2A1)') CHAR(27), '9'
        WRITE(*,'(A/A/A)') ' PUT A CLEAN SHEET OF PAPER IN THE PRINTER',
     +  ' CHECK THE ON-LINE LIGHT', ' PRESS THE ENTER KEY WHEN READY'
        READ(*,'(A1)') A
        GO TO 68
```

```
      ELSE
        IF(A.NE.'N'.AND.A.NE.'n'.AND.A.NE.'-') GO TO 72
      END IF
C     ZERO DATA ARRAY
   76 DO 78 I = 1, 3
      DO 78 J = 1, 3
   78 DATA(I,J) = 0.0
C     * * * * * *
C     THE CURRENT SUBJECT IS FINISHED
C     * * * * * *
      WRITE(*,'(3X,A\)') 'DO YOU HAVE MORE FEET TO MEASURE?  (Y/N)  '
      READ(*,'(A1)') ANS
C     GO BACK TO STATEMENT # 10 AND START A NEW SUBJECT
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 10
C     RETURN TO MAIN MENU
      IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') RETURN
      WRITE(*,'(1X,A1)') BEEP
      GO TO 76
   80 IF(SR.EQ.SN) THEN
C     FILE HAS BEEN REMEASURED, CHANGE FILE NAME PREFIX
        SR = RE
        GO TO 65
      ELSE
C     FILE HAS BEEN MEASURED THREE OR MORE TIMES
        WRITE(*,'(A/A/A/A)') ' THIS SUBJECT HAS BEEN MEASURED TWICE AT T
     +HIS STATION',' TYPE A NEW TWO LETTER PREFIX',
     +  ' IF THE PROGRAM COMES BACK HERE TRY A DIFFERENT PREFIX',
     +  ' IF YOU DO NOT WANT TO SAVE THIS DATA TYPE IN 13'
        READ(*,'(A2)') SR
        IF(SR.EQ.'13') GO TO 76
        GO TO 65
      END IF
      END
      SUBROUTINE MM(I,NAME)
C     THIS SUBROUTINE CHECKS TO SEE IF A DATA POINT IS WITHIN RANGE
      CHARACTER*18 NAME, A*1, BEEP*1
      COMMON/ M / MINMAX(3,2,2), DATA(3,3), IS
      REAL MINMAX
      BEEP = CHAR(7)
C     IF WITHIN RANGE RETURN
    2 IF(DATA(I,1).GE.MINMAX(I,1,IS).AND.DATA(I,1).LE.MINMAX(I,2,IS))
     + RETURN
C     WRITE(*,'('' MIN ='',F8.1,'' MAX ='',F8.1)') MINMAX(I,1,IS),
C     + MINMAX(I,2,IS)
C     REVERSE THE SCREEN
      WRITE(*,'(1X,A1,''[0;7m'')') CHAR(27)
    4 WRITE(*,'('' THIS VALUE IS NOT IN RANGE!'',A1/
     +  '' PLEASE CHECK THE INPUTTED VALUE FOR '',A18,A1/
     +  '' IF YOU WANT TO CHANGE THE VALUE TYPE IN C OR *'',A1/
     +  '' IF THE SUBJECT IS SMALL TYPE IN S OR -'',A1/
     +  '' IF THE SUBJECT IS LARGE TYPE IN L OR + '',A1,\)')
     + BEEP,NAME,BEEP,BEEP,BEEP,BEEP
      READ(*,'(A1)') A
C     RESET THE SCREEN
```

```
      WRITE(*,'(1X,A1,''[0;1m'')') CHAR(27)
C     SMALL DATA VALUE
        IF((A.EQ.'S'.OR.A.EQ.'s'.OR.A.EQ.'-').AND.
     + DATA(I,1).LT.MINMAX(I,2,IS)) RETURN
C     LARGE DATA VALUE
        IF((A.EQ.'L'.OR.A.EQ.'l'.OR.A.EQ.'+').AND.
     + DATA(I,1).GT.MINMAX(I,2,IS)) RETURN
C     READ NEW DATA VALUE
        IF(A.EQ.'C'.OR.A.EQ.'c'.OR.A.EQ.'*') THEN
         WRITE(*,'(1X,A18,'' = '',\)') NAME
        READ(*,*) DATA(I,1)
        GO TO 2
        END IF
         GO TO 4
       END
       SUBROUTINE REGEQ
C     THIS SUBROUTINE ESTIMATES EACH DATA POINT AND COMPARES
C       THE ESTIMATED AND MEASURED VALUES
       COMMON/ M / MINMAX(3,2,2), DATA(3,3), IS
       REAL MINMAX
       COMMON/ REG / EST(3), IVN(3,2,2), EQC(3,3,2), SE(3,2),
     + ER(0:3), KLM(0:3), NVS
       DO 2 I = 1, NVS
C     REGRESSION ESTIMATE
       EST(I) = DATA(IVN(I,1,IS),1) * EQC(I,1,IS) + DATA(IVN(I,2,IS),1) *
     + EQC(I,2,IS) + EQC(I,3,IS)
C     DIFFERENCE BETWEEN ESTIMATED AND MEASURED DATA DIVIDED BY THE
C       STANDARD ERROR OF ESTIMATE
       ER(I) = ( DATA(I,1) - EST(I) ) / SE(I,IS)
C    CHECK FOR ZERO VALUES
C    SET ER(I) TO 0.0 IF ANY OF THE VARIABLES IN THE EQUATION ARE ZERO
       IF(DATA(I,1).EQ.0.0) ER(I) = 0.0
       IF(DATA(IVN(I,1,IS),1).EQ.0.0) ER(I) = 0.0
       IF(DATA(IVN(I,2,IS),1).EQ.0.0) ER(I) = 0.0
     2 IF(ER(I).LT.0.0) ER(I) = -ER(I)
       ER(0) = 0.0
       KLM(0) = 0
       KLM(1) = 1
C     ORDER BY SIZE OF ER(L), SMALLEST ER(1), LARGEST ER(NVS)
       DO 8 L = 2, NVS
       LL = L - 1
       GO TO 6
     4 KLM(LL+1) = KLM(LL)
       LL = LL - 1
     6 IF(ER(L).LT.ER(KLM(LL))) GO TO 4
     8 KLM(LL+1) = L
       WRITE(*,'(5F16.2)') ( ER(I), I = 1, NVS )
       RETURN
       END
       SUBROUTINE WSEX(IS)
C    SUBROUTINE WSEX ASKS WHICH SEX IS TO BE MEASURED
C       THEN CHECKS FOR A CORRECT RESPONSE
       CHARACTER*1 SEX
     2 WRITE(*,'('' WHAT SEX IS TO BE MEASURED? (M/F)  '',\)')
```

```
      READ(*,'(A1)') SEX
      IF(SEX.EQ.'M'.OR.SEX.EQ.'m'.OR.SEX.EQ.'1') THEN
        WRITE(*,'('' WE NOW MEASURE MEN'')')
        IS = 1
      ELSE
        IF(SEX.EQ.'F'.OR.SEX.EQ.'f'.OR.SEX.EQ.'2') THEN
          WRITE(*,'('' WE NOW MEASURE WOMEN'')')
          IS = 2
        ELSE
          WRITE(*,'('' ONE MORE TIME'')')
          GO TO 2
        END IF
      END IF
      RETURN
      END
      SUBROUTINE QESIN
C     SUBROUTINE QESIN ENTER QUESTIONNAIRE DATA AND WRITE ONTO THE
C      SUBJECT'S DISK
      CHARACTER*20 TP, ANS*1, DP(2)*4, MPCB*10, RANK*3, GRADE*3, WH*1
      CHARACTER*10 UNIT(5), UNITD(5)*18, ESC*3, UPLINE*14,MOS(2)*5, WE*1
      CHARACTER*9 POST(15), C20(9)*20, C1(3)*1, Q*3, AA*5, BEEP*1
      DIMENSION INDATA(15)
      COMMON/STUFF/ DATA(15), IDATA(7), NSUBR, C20, C1
      DATA POST / 'McCLELLAN','CAMPBELL','BRAGG','STEWART','ORD',
     + 'LEWIS','HOOD','JACKSON','DIX','RUCKER','GORDON',4*' ' /
      DATA UNIT, UNITD / 'COMPANY', 'BATTALION', 'REGIMENT', 'BRIGADE',
     + 'DIVISION', 'COMPANY OR BATTERY', 'BATTALION', 'REGIMENT OR GROUP
     +', 'BRIGADE', 'DIVISION'/
      DATA MONTH, MDAY, MYEAR / 3*0 /, TP, DP / ' ', 'DATE', 'POST' /
C     ESCAPE CHARACTER FOR SCREEN CONTROL
      ESC = ' '//CHAR(27)//'['
      BEEP = CHAR(7)
      IZ = 0
      NOISE = 0
C     MOVE THE CURSOR UP ONE LINE AT THE START OF THE LINE
      UPLINE = ESC//'1A'//ESC//'3D'//ESC//'K'
      WRITE(*,'(A3,A)') ESC, '2J'
C     SET POST NAME
    1 WRITE(*,'(11(5X,I3,3X,A9/),A\)') ( I, POST(I), I = 1, 11 ),
     +  '   ENTER THE NUMBER OF TODAY''S POST  '
      READ(*,*) ITP
C     INCORRECT RESPONSE TRY AGAIN
      IF(ITP.LT.1.OR.ITP.GT.15) GO TO 1
      IF(ITP.GT.11.AND.ITP.LT.16) THEN
C     THE NAME OF TODAY'S POST IS NOT ON THE LIST SO ENTER IT
        WRITE(*,'(3X,A\)') 'TYPE IN NAME OF TODAY''S POST  '
        READ(*,'(A9)') POST(ITP)
      END IF
C     WRITE TODAY'S POST ON THE SCREEN AND CHECK IF OK
        WRITE(*,'(3X,A,A9/A\)') 'THE NAME OF TODAY''S POST IS ',
     + POST(ITP), '  DO YOU WANT TO CHANGE TODAY''S POST?  (Y/N)  '
       READ(*,'(A1)') ANS
       IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') GO TO 1
C     GET TODAY'S DATE FROM SYSTEM
```

```
      CALL GETDAT(MYEAR,MONTH,MDAY)
C     START A NEW SUBJECT
    2 WRITE(*,'(3X,A)') 'PUT SUBJECT''S DISK IN DRIVE. THEN PRESS ENTER'
      READ(*,'(A1)') ANS
      OPEN(12,FILE='B:SUBJECT.NUM',STATUS='OLD',ERR=2)
C     GET SUBJECT'S NUMBER FROM SUBJECT.NUM FILE
      READ(12,*) NSUBCK
      CLOSE(12)
C     READ SUBJECT'S NUMBER FROM KEYBOARD
    3 WRITE(*,'(3X,A\)') 'TYPE IN THE SUBJECT NUMBER PLEASE: '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
C     NOISE IS SET NOT EQUAL TO ZERO IF THERE IS AN IO ERROR READING AN
C      ENTRY FROM THE KEYBOARD
C     THE ERR IN THE READ STATEMENT SENDS THE PROGRAM BACK TO WRITE THE
C      QUESTION ON THE SCREEN
C     THEN THE COMPUTER BEEPS BECAUSE NOISE IS NOT EQUAL TO ZERO
C     NOISE IS SET TO ZERO IF THE NEXT READ IS OK
C     SUBROUTINES QESIN, PTWO, PTHREE, AND PFOUR DO THIS WHEN A REAL OR
C      INTEGER VALUE IS ENTERED
      READ(*,*,ERR=3,IOSTAT=NOISE) NSUB
      IF(NSUB.NE.NSUBCK) THEN
C     SUBJECT NUMBERS DO NOT MATCH, FIX IT
         WRITE(*,'(3X,A1,A,I7,2X,A,I7,/3X,A)') BEEP,
     + 'THE DISK IS FOR SUBJECT NUMBER ', NSUBCK, 'NOT SUBJECT NUMBER ',
     + NSUB, 'PLEASE PUT THE CORRECT DISK IN OR TYPE IN THE CORRECT SUBJ
     +ECT NUMBER!'
         GO TO 2
      END IF
C     CLEAR SCREEN AND WRITE A HEADING
    6 WRITE(*,'(A3,A\)') ESC, '2J'
      WRITE(*,'(A)') '  BIOGRAPHICAL DATA:  MILITARY HISTORY'
      WRITE(*,'(A,I3,''/'',I3,''/'',I4,A,A)') '  TODAY''S DATE:',
     + MONTH, MDAY, MYEAR, '    TODAY''S POST: FT. ', POST(ITP)
C     DO YOU WANT TO CHANGE THE DATE?
C     THAT IS YOU ARE ENTERING QUESTIONNAIRES NOT FILLED OUT TODAY
    4 WRITE(*,'(A,A4,A\)') '  DO YOU WANT TO CHANGE TODAY''S ', DP(1),
     + ' ?  (Y/N)  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'y'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'+') THEN
C     CHANGE DATE
      WRITE(*,'(1X,A1,A,A1,A\)') CHAR(27),'[1A',CHAR(27), '[K'
      WRITE(*,'(A,A4,A\)') '  TODAY''S ', DP(1), ' = '
      READ(*,*,ERR=4,IOSTAT=NOISE) MONTH, MDAY, MYEAR
        GO TO 6
      ELSE
        IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') THEN
             WRITE(*,'(A9,A1\)') UPLINE, BEEP
             GO TO 4
        END IF
      END IF
      WRITE(*,'(1X,A1,A,A1,A\)') CHAR(27), '[1A', CHAR(27), '[K'
      WRITE(*,'(2X,A,I7)') '1 NSUB =', NSUB
    8 WRITE(*,'(A)') '   2 UNIT TO WHICH YOU ARE ASSIGNED AT THIS POST:'
```

```
C     WRITE THE PREVIOUS SUBJECT'S UNIT IF THE CURRENT SUBJECT IS IN THE
C      SAME UNIT DO NOT CHANGE
   10 DO 12 I = 1, 5
   12 WRITE(*,'(5X,A18,T26,A10)') UNITD(I), UNIT(I)
   13 WRITE(*,'(5X,A\)') 'DO YOU WANT TO CHANGE THE UNIT FOR THIS SUBJEC
     +T?  (Y/N)  '
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
C     READ IN THE UNIT FOR THIS SUBJECT
      DO 16 I = 1, 5
        WRITE(*,'(2X,A18,A\)') UNITD(I), ' = '
   16   READ(*,'(A10)') UNIT(I)
        DO 18 I = 1, 11
   18   WRITE(*,'(A14\)') UPLINE
        WRITE(*,'(A3,A )') ESC, '1A'
        GO TO 10
      ELSE
        IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') THEN
C     INCORRECT ENTRY TRY AGAIN
           WRITE(*,'(A14,A1\)') UPLINE, BEEP
           GO TO 13
        END IF
C     GO ON TO NEXT QUESTION
        WRITE(*,'(A14\)') UPLINE
        WRITE(*,'(A3,A)') ESC, '1A'
      END IF
C     MILITARY COMPONENT
   20 WRITE(*,'(3X,A/5X,A/5X,A\)') '3 MILITARY COMPONENT:', '1 - REGULAR
     + ARMY   2 - ARMY RESERVE   3 - NATIONAL GUARD   4 - MISSING',
     + 'TYPE IN THE APPROPRIATE NUMBER  '
      READ(*,'(A1)') ANS
C     IF MISSING IS ENTERED AS M CHANGE TO 4
      IF(ANS.EQ.'M'.OR.ANS.EQ.'m') ANS = '4'
      IF(ANS.LT.'1'.OR.ANS.GT.'4') THEN
C     INCORRECT ENTRY TRY AGAIN
         WRITE(*,'(3A14,A1\)') UPLINE, UPLINE, UPLINE, BEEP
         WRITE(*,'(A3,A)') ESC, '1A'
         GO TO 20
      END IF
C     CONVERT FROM CHARACTER TO INTEGER
      MC = ICHAR(ANS) - 48
C     MILITARY PERSONNEL CLASS
   22 WRITE(*,'(3X,A/5X,A/5X,A\)') '4 MILITARY PERSONNEL CLASS:',
     + 'E - ENLISTED   W - WARRANT OFFICER   C - COMMISSIONED OFFICER
     +M - MISSING', 'TYPE IN THE APPROPRIATE LETTER  '
      READ(*,'(A1)') ANS
      MPC = - 1
      IF(ANS.EQ.'E'.OR.ANS.EQ.'e') MPC = 1
      IF(ANS.EQ.'W'.OR.ANS.EQ.'w') MPC = 2
      IF(ANS.EQ.'C'.OR.ANS.EQ.'c') MPC = 3
      IF(ANS.EQ.'M'.OR.ANS.EQ.'m') MPC = 0
      IF(MPC.LT.0) THEN
C     INCORRECT ENTRY TRY AGAIN
         WRITE(*,'(3A14,A1\)') UPLINE, UPLINE, UPLINE, BEEP
```

```
          WRITE(*,'(A3,A\)') ESC, '4D'
          GO TO 22
        END IF
        IF(MPC.GT.1.AND.MPC.LT.4) THEN
C     GET BRANCH FOR WARRANT OFFICERS AND COMMISSIONED OFFICERS
          WRITE(*,'(5X,A\)') 'SPECIFY BRANCH:  '
          READ(*,'(A10)') MPCB
        END IF
C     RANK AND GRADE ON SAME LINE
   24   WRITE(*,'(3X,A\)') '5 RANK/GRADE RANK: '
        READ(*,'(A3)') RANK
        WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '28C', 'GRADE: '
        READ(*,'(A3)') GRADE
C     TIME IN SERVICE, YEARS AND MONTHS ON SAME LINE
   26   WRITE(*,'(3X,A\)') '6 TIME IN SERVICE YEARS: '
        IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
        READ(*,*,ERR=26,IOSTAT=NOISE) IYTIS
        WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '30C', 'MONTHS: '
        READ(*,*,ERR=26,IOSTAT=NOISE) IMTIS
C     MOS  PRIMARY AND SECONDARY ON SAME LINE
   28   WRITE(*,'(3X,A\)') '7 MOS PRIMARY: '
        READ(*,'(A5)') MOS(1)
        WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '26C', 'SECONDARY: '
        READ(*,'(A5)') MOS(2)
C     WHICH HAND?
   30   WRITE(*,'(10X,A/3X,A/5X,A\)') 'R - RIGHT   L - LEFT   E - EITHER
       + M - MISSING', '8 WITH WHICH HAND DO YOU USUALLY FIRE A WEAPON?',
       + 'ENTER THE APPROPRIATE LETTER: '
        READ(*,'(A1)') WH
C     CALL SUBROUTINE HAND TO CHANGE THE CHARACTER VALUE INTO AN INTEGER
C      VALUE
        CALL HAND ( WH, IWH )
        IF(IWH.LT.0) THEN
C     INCORRECT ENTRY TRY AGAIN
          WRITE(*,'(3A14,A1,A3,A2)') UPLINE, UPLINE, UPLINE,BEEP, ESC,'1A'
          GO TO 30
        END IF
C     WHICH EYE?
   32   WRITE(*,'(3X,A/5X,A\)') '9 WITH WHICH EYE DO YOU USUALLY SIGHT YOU
       +R WEAPON?', 'ENTER THE APPROPRIATE LETTER: '
        READ(*,'(A1)') WE
C     CALL SUBROUTINE HAND TO CHANGE THE CHARACTER VALUE INTO AN INTEGER
C      VALUE
        CALL HAND ( WE, IWE )
        IF(IWE.LT.0) THEN
C     INCORRECT ENTRY TRY AGAIN
          WRITE(*,'(2A14,A1,A3,A2)') UPLINE, UPLINE, BEEP, ESC, '1A'
          GO TO 32
        END IF
C     DOES THE OPERATOR WANT TO CHANGE ANYTHING?
   34   WRITE(*,'(3X,A\)') 'DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMA
       +TION  (Y/N)  '
        READ(*,'(A1)') ANS
        IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
```

```
C     IF Y THEN TYPE IN FIRST VARIABLE NUMBER TO BE CHANGED
   36     WRITE(*,'(3X,A\)') 'TYPE IN THE NUMBER OF THE FIRST QUESTION TO
      + BE CHANGED  '
         READ(*,'(I1)',ERR=36) I
C     GO TO THE LINE THAT WRITES THE QUESTION ON THE SCREEN FOR THE
C      VARIABLE TO BE CHANGED
         GO TO ( 2, 8, 20, 22, 24, 26, 28, 30, 32 ), I
         GO TO 34
      ELSE
         IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') THEN
C     INCORRECT ENTRY TRY AGAIN
           WRITE(*,'(1X,A1)') BEEP
           GO TO 34
         END IF
C     FINISHED WITH PAGE ONE GO ON TO NEXT PAGE
      END IF
C     STORE ONLY THE LAST TWO DIGITS OF MYEAR
      IF(MYEAR.GT.99) MYEAR = MOD(MYEAR,100)
C     CLEAR SCREEN AND CALL PTWO TO ENTER SECOND PAGE
      WRITE(*,'(A3,A2\)') ESC, '2J'
      CALL PTWO( IDATA(1), IDATA(2), IDATA(3), IDATA(4), IDATA(5),
     + C1(1), C20(1), DATA(1), DATA(2), DATA(3), IDATA(6), IDATA(7) )
C     CLEAR SCREEN AND CALL PTHREE TO ENTER THIRD PAGE
      WRITE(*,'(A3,A2\)') ESC, '2J'
      CALL PTHREE( DATA(4), DATA(5), DATA(6), DATA(7), DATA(8),
     + DATA(9), DATA(10), DATA(11), DATA(12), DATA(13) )
C     CLEAR SCREEN AND CALL PFOUR TO ENTER FOURTH PAGE
      WRITE(*,'(A3,A2\)') ESC, '2J'
      CALL PFOUR( C20(2), C20(3), C20(4), C1(2), C20(5), C1(3),
     +  C20(6), C20(7), C20(8), C20(9), DATA(14), DATA(15) )
C     WRITE NSUB INTO CHARACTER VARIABLE AA
      WRITE(AA,'(I5.5)') NSUB
C     OPEN FILE       .QES ON SUBJECT'S DISK
      OPEN(10,FILE='B:SN'//AA//'.QES',STATUS='NEW')
C     WRITE THE QUESTIONNAIRE INPUT TO THE      .QES FILE
      WRITE(10,'(I6,4I2,5A10,2I1,A10,2A3,2I2,2A5,2I1)') NSUB, MONTH,
     + MDAY, MYEAR, ITP, UNIT, MC, MPC, MPCB, RANK, GRADE, IYTIS, IMTIS,
     + MOS, IWH, IWE
      WRITE(10,'(5I2,A1,A20,2F4.1,F5.1,2I1)') (IDATA(I),I=1,5),
     + C1(1), C20(1), ( DATA(I), I = 1, 3 ), IDATA(6), IDATA(7)
      WRITE(10,'(10F5.1)') ( DATA(I), I = 4, 13 )
      WRITE(10,'(3A20,A1,A20,A1,4A20,2F6.1)') ( C20(I), I = 2, 4 ),
     + C1(2), C20(5), C1(3), ( C20(I), I = 6, 9 ), DATA(14),DATA(15)
      CLOSE(10)
C     THIS SUBJECT IS FINISHED ARE THERE MORE SUBJECTS TO BE ENTERED?
   38 WRITE(*,'(3X,A\)') 'DO YOU HAVE MORE DATA TO ENTER  (Y/N)  '
      READ(*,'(A1)') ANS
C     GO TO LINE #2 AND START A NEW SUBJECT
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 2
C     RETURN TO THE MAIN MENU
      IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') RETURN
C     INCORRECT ENTRY TRY AGAIN
      WRITE(*,'(1X,A1)') BEEP
      GO TO 38
```

```
      STOP
      END
      SUBROUTINE HAND( C, I )
C     SUBROUTINE HAND CONVERTS FROM A CHARACTER CODE TO AN INTEGER CODE
      CHARACTER*1 C, CODE(0:3), NAME(0:3)*7
      DATA CODE, NAME / 'M', 'R', 'L', 'E', 'MISSING', 'RIGHT', 'LEFT',
     +  'EITHER' /
C     CHANGE A LOWER CASE LETTER TO AN UPPER CASE LETTER
      IF( ICHAR(C).GT.90 ) C = CHAR( ICHAR(C) - 32 )
      DO 2 I = 0, 3
         IF(C.EQ.CODE(I) ) THEN
C     WHEN CODE ENTERED MATCHES ONE OF THE CODES IN THE ARRAY CODE
C     WRITE THE NAME OF THE CODE ON THE SCREEN AND RETURN
C     I IS THE INTEGER CODE
           WRITE(*,'(1X,A1,A3,A1,A4,A)') CHAR(27), '[1A', CHAR(27),
     +       '[40C', NAME(I)
           RETURN
         END IF
    2 CONTINUE
C     THE CODE ENTERED WAS NOT VALID
      I = -1
      RETURN
      END
      SUBROUTINE PTWO( IMON, IDAY, IYR, IAGE, IS, R, RACE, FEET,
     + INCH, POUND, GL, HD )
C     SUBROUTINE PTWO FOR ENTERING THE SECOND PAGE OF THE QUESTIONNAIRE
      CHARACTER*20 RACE, R*1, ANS*1, WH*1, BEEP*1, ESC*3
      INTEGER GL, HD
      REAL INCH
      ESC = ' '//CHAR(27)//'['
      BEEP = CHAR(7)
      IZ = 0
      NOISE = IZ
      WRITE(*,'(A)') ' BIOGRAPHICAL DATA: PERSONAL HISTORY'
C     SUBJECT'S BIRTHDAY MONTH, DAY, YEAR ON ONE LINE WITH A SPACE IN
C      BETWEEN
   10 WRITE(*,'(A/A\)') '  1 - TYPE IN SUBJECT''S BIRTHDATE PLEASE'  ,
     +  '  MONTH, DAY, YEAR  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,*,ERR=10,IOSTAT=NOISE) IMON, IDAY, IYR
C     ONLY KEEP THE LAST TWO DIGITS OF THE YEAR ( IYR )
      IF(IYR.GT.99) IYR = MOD(IYR,100)
C     SUBJECT'S AGE IN WHOLE YEARS
   20 WRITE(*,'(A\)') '  2 - TYPE IN SUBJECT''S AGE PLEASE  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,*,ERR=20,IOSTAT=NOISE) IAGE
C     SUBJECT'S SEX M OR F
   30 WRITE(*,'(A\)')'  3 - TYPE IN M FOR MALE OR F FOR FEMALE PLEASE  '
      IS = 0
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'M'.OR.ANS.EQ.'m') IS = 1
      IF(ANS.EQ.'F'.OR.ANS.EQ.'f') IS = 2
      IF(ANS.NE.' '.AND.IS.EQ.0) THEN
C     INCORRECT ENTRY TRY AGAIN
```

```
        WRITE(*,'(1X,A1\)') BEEP
        GO TO 30
       END IF
C     SUBJECT'S RACE
   40 WRITE(*,'(A/A/A/A/A/A/A/A/A\)')'  4 - TYPE IN A LETTER FOR THE SUBJE
      +CT''S RACE PLEASE',
      +  '  W  FOR White, not of Hispanic origin',
      +  '  B  FOR Black, not of Hispanic origin',
      +  '  H  FOR Hispanic',
      +  '  A  FOR Asian/Pacific Islander',
      +  '  I  FOR American Indian/Alaskan Native',
      +  '  M  FOR Mixed',
      +  '  O  FOR Other    '
       READ(*,'(A1)') R
       RACE = '    '
       IF(R.EQ.'M'.OR.R.EQ.'m'.OR.R.EQ.'O'.OR.R.EQ.'o') THEN
C     IF RACE IS MIXED OR OTHER THE PROGRAM ASKS FOR A RACE TO BE TYPED
C      IN 20 CHARACTERS OR LESS
        WRITE(*,'(A\)') ' PLEASE SPECIFY: '
        READ(*,'(A20)') RACE
       END IF
C     HOW TALL? FEET AND INCHES ON THE SAME LINE
   50 WRITE(*,'(A/A\)') '  5 - HOW TALL IS THE SUBJECT IN BARE FEET?',
      +  '  FEET: '
       IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
       READ(*,*,ERR=50,IOSTAT=NOISE) FEET
       WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '10C', 'INCHES: '
       READ(*,*,ERR=50,IOSTAT=NOISE) INCH
C     SUBJECT'S WEIGHT IN POUNDS
   60 WRITE(*,'(A\)') '  6 - HOW MUCH DOES THE SUBJECT WEIGH WITHOUT CL
      +OTHES, IN POUNDS  '
       IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
       READ(*,*,ERR=60,IOSTAT=NOISE) POUND
C     GLASSES OR CONTACTS?
   70 WRITE(*,'(A/2A/3A/,A\)') '  7 - DOES THE SUBJECT WEAR:?',
      +  '  G FOR PRESCRIPTION GLASSES',
      +  '  C FOR PRESCRIPTION CONTACT LENS',
      +  '  B FOR BOTH', '  N FOR NEITHER', '  M FOR MISSING',
      +  '    TYPE IN A LETTER  '
       READ(*,'(A1)') ANS
       GL = -1
       IF(ANS.EQ.'N'.OR.ANS.EQ.'n') GL = 1
       IF(ANS.EQ.'G'.OR.ANS.EQ.'g') GL = 2
       IF(ANS.EQ.'C'.OR.ANS.EQ.'c') GL = 3
       IF(ANS.EQ.'B'.OR.ANS.EQ.'b') GL = 4
       IF(ANS.EQ.'M'.OR.ANS.EQ.'m') GL = 0
       IF(GL.LT.0) THEN
C     INCORRECT ENTRY TRY AGAIN
        WRITE(*,'(1X,A1\)') BEEP
        GO TO 70
       END IF
C     WHICH HAND DOES THE SUBJECT WRITE WITH?
   80 WRITE(*,'(10X,A/3X,A/5X,A\)') 'R - RIGHT   L - LEFT   E - EITHER
      + M - MISSING', '8 WITH WHICH HAND DO YOU USUALLY WRITE?',
```

```
     + 'ENTER THE APPROPRIATE LETTER: '
      READ(*,'(A1)') WH
C     CALL SUBROUTINE HAND TO CONVERT FROM A CHARACTER CODE TO AN
C      INTEGER CODE
      CALL HAND ( WH, HD )
      IF(HD.LT.0) THEN
C     INCORRECT RESPONSE TRY AGAIN
        WRITE(*,'(3A14,A1,A3,A2)') UPLINE, UPLINE, UPLINE,BEEP,ESC, '1A'
        GO TO 80
      END IF
C     WRITE THE INFORMATION ENTERED FROM PAGE TWO ONTO THE SCREEN
      WRITE(*,'(A,2X,I4,I4,I6)') '  1 - SUBJECT''S BIRTHDATE  -', IMON,
     + IDAY, IYR
      WRITE(*,'(A,2X,I2)') '  2 - SUBJECT''S AGE IS ', IAGE
      WRITE(*,'(A,2X,I1)') '  3 - SUBJECT''S SEX IS ', IS
      WRITE(*,'(A,2X,A1,4X,A20)') '  4 - SUBJECT''S RACE IS ', R, RACE
      WRITE(*,'(A,F5.1,A,F6.1,A)') '  5 - SUBJECT''S HEIGHT ', FEET,
     +  ' FEET ', INCH, ' INCHES'
      WRITE(*,'(A,F6.0,A)') '  6 - SUBJECT''S WEIGHT', POUND, ' POUNDS'
      WRITE(*,'(A,I2)') '  7 - DOES THE SUBJECT WEAR CONTACTS OR GLASSES
     +', GL
      WRITE(*,'(A,I2)') '  8 - WITH WHICH HAND DOES THE SUBJECT USUALLY
     +WRITE  ', HD
C     DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMATION?
   86 WRITE(*,'(3X,A\)') 'DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMA
     +TION  (Y/N)  '
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
C     ENTER THE NUMBER OF THE FIRST QUESTION YOU WANT TO CHANGE
   88    WRITE(*,'(3X,A\)') 'TYPE IN THE NUMBER OF THE FIRST QUESTION TO
     + BE CHANGED  '
         READ(*,'(I1)',ERR=88) I
C     GO TO THE LINE THAT WRITES THE QUESTION ON THE SCREEN FOR THE
C      VARIABLE TO BE CHANGED
         GO TO ( 10, 20, 30, 40, 50, 60, 70, 80 ), I
         WRITE(*,'(1X,A1)') BEEP
         GO TO 86
      ELSE
         IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') THEN
C     INCORRECT RESPONSE TRY AGAIN
           WRITE(*,'(1X,A1)') BEEP
           GO TO 86
         END IF
C     ALL INFORMATION IS OK GO TO NEXT PAGE
        RETURN
      END IF
      END
      SUBROUTINE PTHREE( WYEARS, WMONTHS, UBDPW, LBDPW, UBHPD, LBHPD,
     + RYEARS, RMONTHS, DPW, RMILE )
C     SUBROUTINE PTHREE ENTER DATA FROM THIRD PAGE OF QUESTIONNAIRE
      REAL LBDPW, LBHPD
      CHARACTER*1 ANS, BEEP, ESC*3
      ESC = ' '//CHAR(27)//'['
      BEEP = CHAR(7)
```

```
      IZ = 0
      NOISE = IZ
C     DOES THE SUBJECT PARTICIPATE IN WEIGHT TRAINING?
   90 WRITE(*,'(A,6X,A\)') '  9 - DOES THE SUBJECT CURRENTLY PARTICIPATE
     + IN RESISTANCE OR FREE-WEIGHT','TRAINING AT LEAST ONCE A WEEK?  (Y/
     +N)  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,'(A1)') ANS
C     SUBJECT PARTICIPATES IN WEIGHT TRAINING
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
C     HOW LONG? YEARS AND MONTHS ON SAME LINE
         WRITE(*,'(A/A\)') '  A - HOW LONG HAS THE SUBJECT BEEN INVOLVED
     +IN THIS TRAINING?  ', '  YEARS: '
         READ(*,*,ERR=90,IOSTAT=NOISE) WYEARS
         WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '12C', 'MONTHS: '
         READ(*,*,ERR=90,IOSTAT=NOISE) WMONTHS
C     DAYS PER WEEK
         WRITE(*,'(A/A\)') '  B - HOW MANY DAYS PER WEEK DOES THE SUBJECT
     + NOW TRAIN?  ', '  UPPER BODY: DAYS PER WEEK  '
         READ(*,*,ERR=90,IOSTAT=NOISE) UBDPW
         WRITE(*,'(A\)') '  LOWER BODY: DAYS PER WEEK  '
         READ(*,*,ERR=90,IOSTAT=NOISE) LBDPW
C     HOURS PER DAY
         WRITE(*,'(A/A\)') '  C - ON THE DAYS THAT THE SUBJECT TRAINS HOW
     + MANY HOURS DOES THE SUBJECT TRAIN?'
     +   , '  UPPER BODY: HOURS PER DAY  '
         READ(*,*,ERR=90,IOSTAT=NOISE) UBHPD
         WRITE(*,'(A\)') '  LOWER BODY: HOURS PER DAY  '
         READ(*,*,ERR=90,IOSTAT=NOISE) LBHPD
      ELSE
C     INCORRECT ENTRY TRY AGAIN
         IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') GO TO 90
C     SUBJECT DOES NOT WEIGHT TRAINING SET ALL DATA VALUES TO ZERO
         WYEARS = 0.0
         WMONTHS = 0.0
         UBDPW = 0.0
         LBDPW = 0.0
         UBHPD = 0.0
         LBHPD = 0.0
      END IF
C     DOES SUBJECT RUN?
  100 WRITE(*,'(A\)') '  10 - DOES THE SUBJECT RUN ON A REGULAR BASIS?
     +(Y/N)  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,'(A1)') ANS
C     SUBJECT RUNS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
C     HOW LONG?  YEARS AND MONTHS ON SAME LINE
         WRITE(*,'(A/A\)') '  A - HOW LONG HAS THE SUBJECT BEEN RUNNING?
     + '   , '  YEARS: '
         READ(*,*,ERR=100,IOSTAT=NOISE) RYEARS
         WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '10C', 'MONTHS: '
         READ(*,*,ERR=100,IOSTAT=NOISE) RMONTHS
C     DAYS PER WEEK
```

```
      WRITE(*,'(A/A\)') '  B - HOW MANY DAYS PER WEEK DOES THE SUBJECT
     + NOW RUN?', '  DAYS PER WEEK  '
      READ(*,*,ERR=100,IOSTAT=NOISE) DPW
      WRITE(*,'(A/A\)')
C     MILES PER DAY
     +  '  C - ON THE DAYS THE SUBJECT RUNS, HOW MANY MILES DOES THE SUB
     +JECT USUALLY COVER', '  MILES  '
      READ(*,*,ERR=100,IOSTAT=NOISE) RMILE
     ELSE
C     INCORRECT ENTRY TRY AGAIN
      IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') GO TO 100
C     SUBJECT DOES NOT RUN SET ALL DATA VALUES TO ZERO
      RYEARS = 0.0
      RMONTHS = 0.0
      DPW = 0.0
      RMILE = 0.0
     END IF
     IF(WYEARS+WMONTHS.GT.0.0) THEN
C     WRITE WEIGHT TRAINING DATA TO SCREEN
      WRITE(*,'(A,F5.1,A,F5.1,A)') ' THE SUBJECT HAS BEEN WEIGHT TRAIN
     +ING FOR ', WYEARS, ' YEARS, ', WMONTHS, ' MONTHS'
      WRITE(*,'(A,F5.1,A,F5.1,A)') ' THE SUBJECT TRAINS FOR ', UBDPW,
     +  ' DAYS PER WEEK ON THE UPPER BODY AND ', LBDPW,
     +  ' ON THE LOWER BODY'
      WRITE(*,'(A,F5.1,A,F5.1,A)') ' THE SUBJECT TRAINS FOR ', UBHPD,
     +  'HOURS PER DAY ON THE UPPER BODY AND ', LBHPD,
     +  ' ON THE LOWER BODY'
     ELSE
C     WRITE SUBJECT DOES NOT PARTICIPATE TO SCREEN
      WRITE(*,'(A)') ' THE SUBJECT DOES NOT PARTICIPATE IN WEIGHT TRAI
     +NING'
     END IF
     IF(RYEARS+RMONTHS.GT.0) THEN
C     WRITE RUNNING DATA TO SCREEN
      WRITE(*,'(A,F5.1,A,F5.1,A)') ' THE SUBJECT HAS BEEN RUNNING FOR
     +', RYEARS, ' YEARS, ', RMONTHS, ' MONTHS'
      WRITE(*,'(A,F5.1,A)') ' THE SUBJECT RUNS ', DPW, ' A WEEK'
      WRITE(*,'(A,F5.1,A)') ' THE SUBJECT RUNS FOR ', RMILE, 'MILES'
     ELSE
C     WRITE SUBJECT DOES NOT RUN TO SCREEN
      WRITE(*,'(A)') ' THE SUBJECT DOES NOT RUN'
     END IF
C     DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMATION?
  102 WRITE(*,'(3X,A\)') 'DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMA
     +TION  (Y/N)  '
     READ(*,'(A1)') ANS
     IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') THEN
C     OK GO TO THE NEXT PAGE
      RETURN
     ELSE
C     GO TO THE TOP OF THE PAGE AND ENTER THE INFORMATION AGAIN
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 90
C     INCORRECT ENTRY TRY AGAIN
      WRITE(*,'(1X,A1\)') BEEP
```

```
        GO TO 102
       END IF
       END
       SUBROUTINE PFOUR( SBIRTH, MBIRTH, FBIRTH, MR, MRACE, FR, FRACE,
      + SENE, MENE, FENE, AH, AW )
C     SUBROUTINE PFOUR ENTERS DATA FOR THE FOURTH PAGE OF THE
C      QUESTIONNAIRE
       CHARACTER*20 SENE, MENE, FENE, SBIRTH, MBIRTH, FBIRTH, FRACE
       CHARACTER*20 MRACE, FR*1, MR*1, BEEP*1, ESC*3
       ESC = ' '//CHAR(27)//'['
       BEEP = CHAR(7)
       IZ = 0
       NOISE = IZ
C     TYPE IN BIRTHPLACES  UP TO 20 CHARACTERS
  110  WRITE(*,'(A\)') ' 11 - SUBJECT''S BIRTHPLACE:  '
       READ(*,'(A20)') SBIRTH
  120  WRITE(*,'(A\)') '  12 - MOTHER''S BIRTHPLACE:   '
       READ(*,'(A20)') MBIRTH
  130  WRITE(*,'(A\)') '  13 - FATHER''S BIRTHPLACE:   '
       READ(*,'(A20)') FBIRTH
C     MOTHER'S RACE
  140  WRITE(*,'(A/A/A/A/A/A/A/A\)')'  14 - TYPE IN A LETTER FOR THE SUBJ
      +ECT''S MOTHER''S RACE PLEASE',
      +   '  W  FOR White, not of Hispanic origin',
      +   '  B  FOR Black, not of Hispanic origin',
      +   '  H  FOR Hispanic',
      +   '  A  FOR Asian/Pacific Islander',
      +   '  I  FOR American Indian/Alaskan Native',
      +   '  M  FOR Mixed',
      +   '  O  FOR Other    '
       READ(*,'(A1)') MR
       MRACE = '    '
       IF(MR.EQ.'M'.OR.MR.EQ.'m'.OR.MR.EQ.'O'.OR.MR.EQ.'o') THEN
C     IF RACE IS MIXED OR OTHER SPECIFY
         WRITE(*,'(A)') ' PLEASE SPECIFY: '
         READ(*,'(A20)') MRACE
       END IF
C     FATHER'S RACE
  150  WRITE(*,'(A/A/A/A/A/A/A/A\)')'  15 - TYPE IN A LETTER FOR THE SUBJ
      +ECT''S FATHER''S RACE PLEASE',
      +   '  W  FOR White, not of Hispanic origin',
      +   '  B  FOR Black, not of Hispanic origin',
      +   '  H  FOR Hispanic',
      +   '  A  FOR Asian/Pacific Islander',
      +   '  I  FOR American Indian/Alaskan Native',
      +   '  M  FOR Mixed',
      +   '  O  FOR Other    '
       READ(*,'(A1)') FR
       FRACE = '    '
       IF(FR.EQ.'M'.OR.FR.EQ.'m'.OR.FR.EQ.'O'.OR.FR.EQ.'o') THEN
C     IF RACE IS MIXED OR OTHER SPECIFY
         WRITE(*,'(A)') ' PLEASE SPECIFY: '
         READ(*,'(A20)') FRACE
       END IF
```

```
C     TYPE IN ETHNICITY/NATIONAL EXTRACTION  UP TO 20 CHARACTERS
  160 WRITE(*,'(A/A\)') '  16 - ETHNICITY/NATIONAL EXTRACTION:',
     +  '  SUBJECT:  '
      READ(*,'(A20)') SENE
      WRITE(*,'(A\)') '  MOTHER:   '
      READ(*,'(A20)') MENE
      WRITE(*,'(A\)') '  FATHER:   '
      READ(*,'(A20)') FENE
C     ACTUAL HEIGHT AND WEIGHT IN MM  AND KG ON ONE LINE
  170 WRITE(*,'(A/A\)') '  17 - BODY DIMENSIONS', '  ACTUAL HEIGHT  '
      IF(NOISE.NE.IZ) WRITE(*,'(1X,A1\)') BEEP
      READ(*,*,ERR=170,IOSTAT=NOISE) AH
      WRITE(*,'(A3,A,A3,A,A\)') ESC, '1A', ESC, '24C', 'ACTUAL WEIGHT  '
      READ(*,*,ERR=170,IOSTAT=NOISE) AW
C     WRITE THE ABOVE INFORMATION TO THE SCREEN
      WRITE(*,'(A,A20)') '  11 - SUBJECT''S BIRTHPLACE ', SBIRTH
      WRITE(*,'(A,A20)') '  12 - MOTHER''S BIRTHPLACE ', MBIRTH
      WRITE(*,'(A,A20)') '  13 - FATHER''S BIRTHPLACE ', FBIRTH
      WRITE(*,'(A,A1,4X,A20)') '  14 - MOTHER''S RACE ', MR, MRACE
      WRITE(*,'(A,A1,4X,A20)') '  15 - FATHER''S RACE ', FR, FRACE
      WRITE(*,'(A/A,A20/A,A20/A,A20)') '  16 - ETHNICITY/NATIONAL EXTRAC
     +TION:', ' SUBJECT: ', SENE, ' MOTHER: ', MENE, ' FATHER: ', FENE
      WRITE(*,'(A,F7.1/A,F7.1)') '  17 - ACTUAL HEIGHT ', AH, ' ACTUAL W
     +EIGHT',  AW
C     DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMATION?
  172 WRITE(*,'(3X,A\)') 'DO YOU WANT TO CHANGE ANY OF THE ABOVE INFORMA
     +TION  (Y/N)  '
      READ(*,'(A1)') ANS
C     YES CHANGE SOME INFORMATION
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
  174    WRITE(*,'(3X,A\)') 'TYPE IN THE NUMBER OF THE FIRST QUESTION TO
     + BE CHANGED  '
C     TYPE IN THE NUMBER OF THE FIRST VARIABLE YOU WANT TO CHANGE
         READ(*,'(I2)',ERR=174) I
         I = I - 10
C     GO TO THE LINE THAT WRITES THE QUESTION ON THE SCREEN FOR THE
C      VARIABLE TO BE CHANGED
         GO TO ( 110, 120, 130, 140, 150, 160, 170 ), I
         WRITE(*,'(1X,A1)') BEEP
         GO TO 172
      ELSE
         IF(ANS.NE.'N'.AND.ANS.NE.'n'.AND.ANS.NE.'-') THEN
C     INCORRECT RESPONSE  TRY AGAIN
           WRITE(*,'(1X,A1)') BEEP
           GO TO 172
         END IF
C     FINISHED ENTERING THIS SUBJECT'S QUESTIONNAIRE
C     RETURN TO QESIN AND WRITE QUESTIONNAIRE DATA TO SUBJECT'S DISK
         RETURN
      END IF
      END
      SUBROUTINE DISKIN
C     THIS SUBROUTINE READS IN ALL THE DATA FILES ON THE SUBJECT'S
C      DISK
```

```
C     IF A DATA FILE THAT HAS BEEN PRINTED ON THE SUBJECT'S PRINTOUT
C      IS MISSING OR INCOMPLETE THE MISSING DATA CAN BE TYPED IN
C     THE SUBJECT'S DATA IS WRITTEN IN BINARY ON THE DATA DISK
      CHARACTER*18 NAME(30,7), MNAME(7,2), EXP(7,2)*127, A*5, B*5
      CHARACTER*2 NS(7), SN, RE, HNAME(28)*18, ANS*1, TWO*2, FILE*8
      CHARACTER*1 BEEP
      DIMENSION NVS(7), REDATA(30,3,7), IDATA(30,7)
      DIMENSION DIFF(30,7), IDIFF(30,7), REER(30,7)
      COMMON/ZEAD/ ISS, NSUB, IDATA, HNAME
      COMMON/SML/ DATA(30,3,7),ER(30,7),IXYZ(3,28),ISUB(7,2),IS(7,2)
      COMMON/SMLNA/ MNAME, EXP
      LOGICAL THERE, WHY
      INTEGER ZERO
      INTEGER*2 IXYZ, IHR, IMIN, ISEC, ITH, IYR, IMON, IDAY
      DATA NS,SN,RE/'.1','.2','.3','.4','.5','.6','.8','SN','RE'/
      ZERO = 0
      BEEP = CHAR(7)
      GO TO 2
      ENTRY START3
C     AT THE START OF THE MAIN PROGRAM THIS SECTION OF CODE IS RUN TO
C      READ IN THE NAMES OF THE VARIABLES
      OPEN(9,FILE='NAMES',STATUS='OLD')
      DO 1 I = 1, 7
    1 READ(9,*) NVS(I), ( NAME(J,I), J = 1, NVS(I) )
      READ(9,*) NVSH, ( HNAME(J), J = 1, NVSH )
      CLOSE(9)
      RETURN
    2 INQUIRE(4,OPENED=THERE)
C     THE FIRST TIME THAT SUBROUTINE DISKIN IS ENTERED DURING THE
C      RUNNING OF PROGRAM INOUT8 A FILE IS OPENED FOR REMEASURE DATA
      IF(.NOT.THERE) THEN
        DO 3 I = 1, 99
C     LOOP #3 GOES UNTIL A FILE NAME THAT HAS NOT BEEN USED IS FOUND
C     THAT FILE IS OPENED AS FILE #4
        IF(I.LT.10) THEN
          WRITE(ANS,'(I1)') I
          FILE = 'A:REM.'//ANS
        ELSE
          WRITE(TWO,'(I2)') I
          FILE = 'A:REM.'//TWO
        END IF
        INQUIRE(FILE=FILE,EXIST=THERE)
        IF(.NOT.THERE) THEN
C     A FILE NAMED REM.X IS NOT ON THE DATA DISK
C     OPEN A FILE NAMED REM.X
          OPEN(4,FILE=FILE,STATUS='NEW')
          GO TO 33
        END IF
    3 CONTINUE
C     ALL THE POSSIBLE EXTENSIONS HAVE BEEN USED
C     PUT IN A NEW DATA DISK
      WRITE(*,'(4(3X,A/))') CHAR(7), '99 FILES ON THE DATA DISK',
     + 'PUT A NEW DATA DISK IN DRIVE A PLEASE',
     + 'PRESS THE ENTER KEY WHEN READY TO CONTINUE'
```

```
      READ(*,'(A1)') ANS
      GO TO 2
   33 CONTINUE
      END IF
C     RESET INITIAL VALUES IN THE ARRAYS AND VARIABLES BEFORE
C      STARTING A NEW SUBJECT
    4 DO 5 I = 1, 7
      DO 51 JJ = 1, 2
        ISUB(I,JJ) = 0
        MNAME(I,JJ) = 'BLANK'
        IS(I,JJ) = 0
        EXP(I,JJ) = 'BLANK'
   51 CONTINUE
      DO 5 L = 1, 30
      IDATA(L,I) = 0
      DIFF(L,I) = -999.0
      IDIFF(L,I) = -999
      REER(L,I) = 0.0
      ER(L,I) = 0.0
      DO 5 J = 1, 3
      REDATA(L,J,I) = 0.0
    5 DATA(L,J,I) = 0.0
      IHB = 0
      ISS = 0
C     START A NEW SUBJECT
   55 WRITE(*,'(///A/A\)')' PLEASE PUT THE NEXT SUBJECT''S DISK IN DRIVE
     + B', ' PLEASE TYPE IN THE SUBJECT NUMBER  '
      READ(*,*,ERR=4) NSUB
      IF(NSUB.LT.0) RETURN
      OPEN(7,FILE='B:SUBJECT.NUM',STATUS='OLD',ERR=55)
      READ(7,*,ERR=55,END=55) NNSUB
      READ(7,'(A1)',ERR=55,END=55) ANS
      IF(ANS.EQ.'m'.OR.ANS.EQ.'M') ISS = 1
      IF(ANS.EQ.'f'.OR.ANS.EQ.'F') ISS = 2
      CLOSE(7)
      IF(NSUB.NE.NNSUB) THEN
C     THE SUBJECT NUMBER ON THE DISK DOES NOT AGREE WITH THE SUBJECT
C      NUMBER TYPED IN
        WRITE(*,'(A,I5/A,I5)')
     +  ' YOU TYPED IN A SUBJECT NUMBER OF ', NSUB,
     +  ' FILE SUBJECT.NUM HAS A SUBJECT NUMBER OF ', NNSUB
        GO TO 55
      END IF
      IF(NSUB.EQ.LNSUB) THEN
C     CHECK TO SEE IF PREVIOUS SUBJECT HAS BEEN ENTERED AGAIN
        WRITE(*,'(A,I6)') ' THIS IS THE LAST SUBJECT ENTERED  NSUB =',
     +   NSUB
        GO TO 55
      END IF
      WRITE(A,'(I5.5)') NSUB
C     HAS A DATA FILE FOR THIS SUBJECT BEEN WRITTEN ON THE DATA DISK?
      INQUIRE(FILE='A:STATION\'//SN//A//'.CON',EXIST=WHY)
      IF(WHY) THEN
        INQUIRE(FILE='B:HAND',EXIST=THERE)
```

```
        IF(THERE) THEN
C     THERE IS A FILE HAND ON THE SUBJECT'S DISK
C     WRITE THE TIME THAT THIS DISK WAS OUT PROCESSED TO THE SCREEN
C     RETURN TO LINE #4 AND START THE NEXT SUBJECT
          WRITE(*,'(1X,A1)') BEEP
          OPEN(7,FILE='B:HAND',STATUS='OLD')
          READ(7,'(I7//4I3/I5,2I3)') NNSUB, IHR, IMIN, ISEC, ITH,
     +      IYR, IMON, IDAY
          WRITE(*,'(3X,A,I5,A,I3,A1,I2,A,2I3,I5)') 'THIS SUBJECT #',
     +      NNSUB, ' WAS OUT-PROCESSED AT', IHR, ':', IMIN, ' ON',
     +      IMON, IDAY, IYR
          CLOSE(7)
          WRITE(*,'(1X,A1)') BEEP
          GO TO 4
        ELSE
C     SOMETHING IS WRONG MAY BE THE SAME SUBJECT NUMBER WAS GIVEN TO
C      TWO PEOPLE?
C     RENAME THE FILE ON THE DATA DISK
C     RETURN TO LINE #4 AND START OUT-PROCESSING THIS SUBJECT AGAIN
          WRITE(*,'(2X,A1,A)') BEEP,
     +      'THERE IS A FILE WITH THIS SUBJECT''S NUMBER ON THE DATA DIS
     +K
          WRITE(B,'(I5.5)') MOD( NSUB, 10000 ) + 90000
          WRITE(*,'(3X,A//10X,A,A5,A,A5,A/)')
     +     'TYPE IN THE FOLLOWING DOS COMMAND', 'REN \STATION\SN',
     +     A, '.CON, SN', B, '.CON'
          PAUSE '   TYPE IN THE ABOVE DOS COMMAND PLEASE'
          WRITE(*,'(3X,A1,A/3X,A/3X,A)') BEEP, 'IF THERE IS NO ERROR MES
     +SAGE CONTINUE OUT-PROCESSING THIS DISK', 'IF THERE IS AN ERROR MES
     +SAGE STOP OUT-PROCESSING THIS DISK', 'AND PUT A NOTE ON THE DISK''
     +S ENVELOP'
          GO TO 4
        END IF
       END IF
C     CLEAR SCREEN AND WRITE SUBJECT NUMBER ON IT
       WRITE(*,'(1X,A1,''[2J'',A,A5//)') CHAR(27), '  SUBJECT ', A
       DO 12 I = 1, 7
C     LOOP #12 THE PROGRAM GOES THRU THIS LOOP FOR STATION 1 - 6 & 8
       NX = 0
       K = 10 + I
       INQUIRE(FILE='B:'//SN//A//NS(I),EXIST=THERE)
C     IS THE FILE FOR THIS STATION ON THE SUBJECT'S DISK?
       IF(THERE) THEN
       OPEN(K,FILE='B:'//SN//A//NS(I),STATUS='OLD',IOSTAT=IOS)
C     IF THE FILE IS THERE OPEN IT TO BE READ
       IF(IOS.EQ.ZERO) THEN
C     FILE WAS OPENED WITHOUT ERROR READ THE FILE
          J = 0
          READ(K,'(I6,2X,A18,I3,2X,A127)',END=8,ERR=8) ISUB(I,1),
     +    MNAME(I,1), IS(I,1), EXP(I,1)
          DO 6 J = 1, 3
     6    READ(K,'(30F7.1)',END=8,ERR=8) ( DATA(L,J,I), L = 1, NVS(I) )
          J = 4
          READ(K,'(30F5.1)',END=8,ERR=8) (ER(L,I), L = 1, NVS(I) )
```

```
      WRITE(*,'(A,A2,A)') ' STATION', NS(I), ' HAS BEEN READ IN'
C     IF THE PROGRAM GETS HERE THE FILE HAS BEEN READ WITHOUT PROBLEM
        CLOSE(K)
        IF(IHB.EQ.1.AND.I.NE.4) THEN
C     IF THE OPERATOR TOLD THE PROGRAM THAT ONLY THE HEADBOARD WAS
C      MEASURED AND A STATION OTHER THAN #4 WAS MEASURED RESET
C      IHB TO 0 SO THAT OTHER STATIONS WILL SHOW ON THE SCREEN
          WRITE(*,'(A,A2,A/A)') ' WARNING STATION NUMBER ', NS(I),
     +     ' HAS BEEN READ', ' YOU TOLD THE COMPUTER THAT THIS SUBJECT WA
     +S ONLY MEASURED AT THE HEADBOARD!!!'
          IHB = 0
        END IF
      ELSE
C     THERE WAS A PROBLEM OPENING THE FILE
        WRITE(*,'(A,A2,A)') ' THE FILE FOR STATION', NS(I),
     +    ' IS MISSING OR DAMAGED'
        NX = 1
      END IF
      ELSE
C     THE FILE IS NOT ON THE SUBJECT'S DISK
C     IF ONLY THE HEADBOARD ( AND MAYBE STATION #4 ) WAS MEASURED
C      THIS IS NOT A PROBLEM AND THE PROGRAM WILL GO TO THE NEXT
C      STATION
        IF(IHB.EQ.1) GO TO 12
        WRITE(*,'(A,A2,A)') ' THE FILE FOR STATION', NS(I),
     +    ' IS MISSING OR DAMAGED'
        NX = 1
      END IF
      J = 10
    8 IF(J.LT.10) THEN
C     IF J IS LESS THAN 10 SOME INFORMATION IS MISSING
        IF(J.EQ.0) WRITE(*,'(A,A2,A)') ' THE FILE FOR STATION',NS(I),
     +  ' IS BLANK'
        IF(J.EQ.1) WRITE(*,'(A,A2,A)') ' THE FILE FOR STATION',NS(I),
     +  ' HAS NO DATA'
C     IF J EQUALS 0 OR 1 NO DATA HAS BEEN READ
C     IF J EQUALS 2 ONLY THE FIRST LINE OF DATA WAS READ
C     IF J IS GREATER THAN 2 ONLY REMEASURED OR ERROR DATA IS MISSING
        IF(J.GT.1.AND.J.LT.4) WRITE(*,'(A,A2,A,I2,A)')
     +  ' THE FILE FOR STATION', NS(I), ' HAS', J-1, ' LINES OF DATA'
        IF(J.EQ.4) WRITE(*,'(A,A2,A)') ' THE FILE FOR STATION',NS(I),
     +  ' HAS NO ERROR DATA'
        IF(J.LT.2) NX = 1
      IF(J.EQ.2) THEN
C     CHECK EACH DATA VALUE
        DO 99 L = 1, NVS(I)
   99   IF(DATA(L,1,I).EQ.0) NX = 1
        END IF
      END IF
C     IF NX EQUALS 0 THEN ALL THE DATA ON THE FIRST MEASUREMENT HAS
C      BEEN READ IN
      IF(NX.EQ.1) THEN
C     NX EQUALS 1 THERE IS A PROBLEM
C     THE PROGRAM WRITES A MENU ON THE SCREEN
```

```
    9   WRITE(*,'(A/A/A/A/A/A/A\)')
     +  '  THERE IS A PROBLEM, PLEASE CHECK THE SUBJECT''S PRINT OUT',
     +  '  M - THIS STATION IS MISSING FROM THE SUBJECT''S PRINT OUT',
     +  '  H - THIS SUBJECT WAS ONLY MEASURED AT THE HEADBOARD',
     +  '  E - THIS STATION WAS PRINTED ON THE SUBJECT''S PRINT OUT. ENT
     +ER THE DATA VALUES',
     +  '  P - PAUSE TO USE DOS SYSTEM COMMANDS',
     +  '  C - CANCEL THIS SUBJECT',
     +  '  PLEASE TYPE IN ONE OF THE ABOVE LETTERS  '
        READ(*,'(A1)') ANS
C     CANCEL THIS SUBJECT GO TO LINE #4 AND START A NEW SUBJECT
C      THIS SUBJECT IS A MESS
        IF(ANS.EQ.'C'.OR.ANS.EQ.'c') GO TO 4
C     THIS STATION IS MISSING FROM THE SUBJECT'S PRINT OUT
C     GO TO NEXT STATION
        IF(ANS.EQ.'M'.OR.ANS.EQ.'m') GO TO 12
C     USE THE DOS SYSTEM TO CHECK THE SUBJECT'S DISK OR ?
C     THEN RETURN TO THE MENU
        IF(ANS.EQ.'P'.OR.ANS.EQ.'p') THEN
          PAUSE
          GO TO 9
        END IF
       IF(ANS.EQ.'E'.OR.ANS.EQ.'e') THEN
C     ENTER DATA FROM SUBJECT'S PRINT OUT
        IF(IS(I,1).LT.1.OR.IS(I,1).GT.2) THEN
C     IF IS ( SEX CODE ) IS NOT 1 OR 2 ENTER SUBJECT NUMBER, MEASURER'S
C      NAME, SEX CODE, & AN EXPLANATION
          WRITE(*,'(A\)') ' TYPE IN THE SUBJECT NUMBER  '
          READ(*,*) ISUB(I,1)
          WRITE(*,'(A\)') ' TYPE IN THE MEASURER''S NAME  '
          READ(*,'(A18)') MNAME(I,1)
          WRITE(*,'(A\)') ' TYPE IN SUBJECT''S SEX (M=1, F=2)  '
          READ(*,*) IS(I,1)
          WRITE(*,'(A\)') ' TYPE IN THE EXPLANATION  '
          READ(*,'(A127)') EXP(I,1)
        END IF
        DO 11 L = 1, NVS(I)
C     GO THRU LOOP #11 FOR EACH VARIABLE
        IF(DATA(L,1,I).LT.1.0) THEN
C     IF DATA VALUE IS MISSING WRITE VARIABLE NAME ON THE SCREEN
C     THEN READ DATA VALUE FROM THE KEYBOARD
          WRITE(*,'(1X,A18,'' =  '',\)') NAME(L,I)
          READ(*,*) DATA(L,1,I)
        ELSE
C     IF DATA VALUE IS THERE WRITE IT TO THE SCREEN AND ASK THE
C      OPERATOR TO CHECK THE VALUE AND REPLACE THE VALUE IF INCORRECT
          WRITE(*,'(1X,A18,'' =  '',F7.1,A)') NAME(L,I), DATA(L,1,I),
     +     ' IS THIS CORRECT? IF SO PRESS THE RETURN KEY, IF NOT TYPE IN
     +THE CORRECT VALUE'
          READ(*,'(F7.1)') X
          IF(X.GT.0) DATA(L,1,I) = X
        END IF
   11   CONTINUE
          INQUIRE(K,OPENED=THERE)
```

```
          IF(THERE)THEN
C     IF THE SUBJECT'S DATA FILE FOR THIS STATION HAS BEEN OPENED
C      REWIND IT
            REWIND(K)
          ELSE
C     IF NOT OPEN  OPEN IT AS A NEW FILE TO WRITE ON
            OPEN(K,FILE='B:'//SN//A//NS(I),STATUS='NEW',IOSTAT=IOS)
          END IF
C     WRITE THE DATA FOR THIS SUBJECT ONTO THE SUBJECT'S DATA DISK
C     THE REMEASURED DATA AND THE ERROR DATA WILL BE 0
          WRITE(K,'(I6,2X,A18,I3,2X,A127)') ISUB(I,1), MNAME(I,1),
     +      IS(I,1), EXP(I,1)
          DO 14 J = 1, 3
   14       WRITE(K,'(30F7.1)') ( DATA(L,J,I), L = 1, NVS(I) )
          WRITE(K,'(30F5.1)') ( ER(L,I), L = 1, NVS(I) )
          CLOSE(K)
         GO TO 12
         END IF
         IF(ANS.EQ.'H'.OR.ANS.EQ.'h') THEN
C     THIS SUBJECT WAS ONLY MEASURED AT THE HEADBOARD AND POSSIBLY AT
C      STATION #4
C     THE PROGRAM WILL NOT PUT ANY MESSAGE ON THE SCREEN ABOUT
C      MISSING STATION DATA FILES
           IHB = 1
           GO TO 12
         END IF
C     IF THE PROGRAM GETS HERE AN INCORRECT RESPONSE WAS TYPED IN
C     THE PROGRAM RETURNS TO THE MENU
         GO TO 9
       END IF
   12 CONTINUE
       IF(IS(1,1).EQ.0) IS(1,1) = ISS
C     IF STATION #1 WAS NOT READ USE VALUES FROM SUBJECT.NUM FILE
       IF(ISUB(1,1).EQ.0) ISUB(1,1) = NSUB
C     FOR CHECKING IF A SUBJECT IS REPEATED
       LNSUB = NSUB
C     SUBROUTINE HEAD READS THE FILE WITH THE HEADBOARD DATA
       CALL HEAD(A)
C     SUBROUTINE SMALL WRITES THE SUBJECT'S INFORMATION ON THE DATA
C      DISK IN BINARY
       CALL SMALL(A)
       IRE = 0
       DO 46 I = 1, 7
       K = 10 + I
C     WERE ANY OF THE STATIONS REMEASURED?
       INQUIRE(FILE='B:'//RE//A//NS(I),EXIST=THERE)
       IF(THERE) THEN
       OPEN(K,FILE='B:'//RE//A//NS(I),STATUS='OLD',ERR=46)
C     READ IN THE DATA FOR THE REMEASURED STATION
       READ(K,'(I6,2X,A18,I3,2X,A127)',END=46,ERR=46) ISUB(I,2),
     + MNAME(I,2), IS(I,2), EXP(I,2)
       DO 42 J = 1, 3
   42  READ(K,'(30F7.1)',END=44,ERR=44) ( REDATA(L,J,I), L = 1, NVS(I) )
       READ(K,'(30F5.1)',END=44,ERR=44) ( REER(L,I), L=1,NVS(I) )
```

```
C     ADD 1 TO IRE IF REMEASURED
   44 IF(REDATA(1,1,I).GT.0.0)IRE = IRE + 1
      CLOSE(K)
      END IF
   46 CONTINUE
      IF(IRE.LT.1) WRITE(*,'(A)') ' THIS SUBJECT WAS NOT REMEASURED'
C     IF NO STATIONS WERE REMEASURED SKIP TO LINE #66
      IF(IRE.LT.1) GO TO 66
   50 DO 54 I = 1, 7
      IF(ISUB(I,2).GT.0) THEN
C     WRITE MESSAGE TO SCREEN
        IF(REDATA(1,1,I).GT.0.0) WRITE(*,'(A,I2,A)') ' STATION', I,
     +     ' WAS REMEASURED'
        DO 52 L = 1, NVS(I)
        IF(DATA(L,1,I).GT.0.0.AND.REDATA(L,1,I).GT.0.0) THEN
C     CALCULATE DIFFERENCE AND SET TO POSITIVE INTEGER
          DIFF(L,I) = DATA(L,1,I) - REDATA(L,1,I)
          IF(DIFF(L,I).LT.0.0) DIFF(L,I) = -DIFF(L,I)
          IDIFF(L,I) = DIFF(L,I) + 0.5
        END IF
   52 CONTINUE
      END IF
   54 CONTINUE
C     WRITE REMEASURE DIFFERENCES TO FILE REM.X ON DATA DISK
      WRITE(4,'(I6,I2,31I4/(33I4))') ISUB(1,1), IS(1,1),
     + ( ( IDIFF(L,I), L = 1, NVS(I) ), I = 1, 7 )
C     WRITE FILE HAND ON THE SUBJECT'S DISK
   66 OPEN(7,FILE='B:HAND',STATUS='NEW',ERR=76)
      CALL GETTIM(IHR,IMIN,ISEC,ITH)
C     GET TIME AND DATE
      CALL GETDAT(IYR,IMON,IDAY)
      WRITE(*,'(3X,A//)') 'INFORMATION FROM THE HAND PHOTO BOX'
C     ASK OPERATOR FOR ROLL NUMBER AND FRAME NUMBERS FROM HAND PHOTO BOX
   68 WRITE(*,'(3X,A\)') 'PLEASE TYPE IN THE ROLL NUMBER  '
      READ(*,*,ERR=68) NROLL
   70 WRITE(*,'(3X,A/3X,A\)') 'PLEASE TYPE IN THE FRAME NUMBERS',
     + 'WITH A SPACE IN BETWEEN  '
      READ(*,*,ERR=70) NF1, NF2
   72 WRITE(*,'(3X,A,I5,2X,A,I3,2X,A,I3,2X,A\)') 'ROLL NUMBER =',
     + NROLL, 'FRAME #1 =', NF1, 'FRAME #2 =', NF2,
     + 'ARE THESE CORRECT? (Y/N) '
C     WRITE ROLL AND FRAME NUMBERS ON SCREEN TO CHECK
C     IF OK THEN WRITE ON TO SUBJECT'S DISK
      READ(*,'(A1)') ANS
      IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') GO TO 68
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 74
      WRITE(*,'(1X,A1)') BEEP
      GO TO 72
C     WRITE HAND PHOTO BOX INFO AND TIME & DATE ON TO SUBJECT'S DISK
   74 WRITE(7,'(I7/3I5/4I3/I5,2I3)') NSUB, NROLL, NF1, NF2, IHR,
     +  IMIN, ISEC, ITH, IYR, IMON, IDAY
      CLOSE(7)
   76 WRITE(*,'(3X,A\)') 'DO YOU HAVE MORE DISKS TO ENTER?  (Y/N)  '
      READ(*,'(A1)') ANS
```

```
C     THIS SUBJECT HAS BEEN FINISHED START NEW SUBJECT OR GO BACK TO
C      MAIN INOUT8 MENU
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') GO TO 4
       IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') RETURN
       WRITE(*,'(1X,A1)') BEEP
       GO TO 76
       END
       SUBROUTINE HEAD(A)
C     THIS SUBROUTINE READS HEADBOARD DATA FROM THE SUBJECT'S DISK
C     IF THE HEADBOARD DATA CAN NOT BE READ FROM THE SUBJECT'S DATA
C      DISK THE OPERATOR CAN ENTER IT FROM THE SUBJECT'S PRINT OUT
       COMMON/ZEAD/ ISS, NSUB, IDATA, HNAME
       COMMON/SML/ DATA(30,3,7),ER(30,7),IXYZ(3,28),ISUB(7,2),IS(7,2)
       CHARACTER*5 A, ANS*1, FNAME*8, HNAME(28)*18
       DIMENSION  IDATA(30,7), HDATA(16), XYZ(3,28)
       INTEGER HDATA
       INTEGER*2 IXYZ
       LOGICAL THERE
C     INITIALIZE
       BIG = 3276.7
       DO 1 J = 1, 3
       DO 1 K = 1, 28
       IXYZ(J,K) = 0
     1 XYZ(J,K) = 0.0
C     IS THE HEADBOARD FILE ON THE SUBJECT'S DISK UNDER THE CORRECT
C      NAME?
       INQUIRE(FILE='B:SN'//A//'.YS',EXIST=THERE)
       IF(.NOT.THERE) GO TO 10
C     YES THE FILE IS THERE OPEN AND READ THE FILE
       OPEN(3,FILE='B:SN'//A//'.YS',STATUS='OLD',ERR=10,BLOCKSIZE=1536)
     2 READ(3,'(3F12.1)',ERR=8,END=8) XYZ
       CLOSE(3)
     4 WRITE(*,'(A)') ' THE HEAD X-Y-Z COORDINATES HAVE BEEN READ IN'
       DO 6 L = 1, 28
       IZYX = 1
       IF(XYZ(1,L).EQ.0.0.OR.XYZ(2,L).LT.-111.1.OR.XYZ(3,L).EQ.0.0)THEN
C     SET ZERO POINTS OR POINTS TAKEN AT THE HOME POSITION TO -999
       IXYZ(1,L) = -999
       IXYZ(2,L) = -999
       IXYZ(3,L) = -999
       IZYX = -1
       END IF
       DO 6 J = 1, 3
       IF(IZYX.GT.0) THEN
         IF(XYZ(J,L).GT.BIG) XYZ(J,L) = BIG
C     SET ANY OUT OF RANGE VALUES TO FIT IN INTEGER*2
         IF(XYZ(J,L).LT.-BIG) XYZ(J,L) = -BIG
         IF(XYZ(J,L).GE.0.0) THEN
C     CONVERT TO INTEGER*2 INCLUDING THE FIRST DECIMAL PLACE
           IXYZ(J,L) = XYZ(J,L) * 10.0 + 0.1
         ELSE
           IXYZ(J,L) = XYZ(J,L) * 10.0 + 0.1
         END IF
       END IF
```

```
    6 CONTINUE
C     HEADBOARD DATA HAS BEEN READ IN RETURN TO DISKIN
      RETURN
    8 NFILES = 0
      GO TO 12
C     PROBLEM READING HEADBOARD DATA
   10 WRITE(*,'(A/A\)') ' CHECK THE BACK OF THE PRINTED FORM FOR HEAD X-
     +Y-Z COORDINATES',
     +' IF THE HEAD X-Y-Z COORDINATES ARE PRINTED ENTER + IF NOT ENTER
     +-  '
C     WAS THE HEADBOARD DATA PRINTED ON THE SUBJECT'S DATA FORM?
      READ(*,'(A1)') ANS
C     SUBJECT WAS NOT MEASURED AT THE HEADBOARD RETURN TO DISKIN
      IF(ANS.EQ.'-') RETURN
      IF(ANS.NE.'+') GO TO 10
      WRITE(*,'(A)') ' THE PROGRAM IS NOT ABLE TO FIND HEAD X-Y-Z COORD
     +INATE FILE'
C     OPERATOR TYPES IN DOS DIR COMMAND TO SEE IF THE HEADBOARD DATA
C      FILE WAS MISNAMED
      PAUSE ' PLEASE TYPE IN   DIR B:*.YS '
   11 WRITE(*,'(A\)') ' PLEASE TYPE IN THE NUMBER OF FILES DISPLAYED ON
     +THE SCREEN  '
      READ(*,*,ERR=11) NFILES
      IF(NFILES.LT.0) GO TO 11
   12 IF(NFILES.EQ.0)THEN
C TYPE IN THE STUFF FROM THE PRINT OUT
      WRITE(*,'(A/A)') ' PLEASE TYPE IN THE X-Y-Z COORDINATES',
     +  ' TYPE IN THE THREE COORDINATES WITH A SPACE IN BETWEEN'
      DO 16 I = 1, 28
C     THE PROGRAM GOES THRU LOOP #16 FOR EACH HEADBOARD LANDMARK
      IF(XYZ(1,I)+XYZ(2,I)+XYZ(3,I).LT.1.0) THEN
C     IF COORDINATES ARE MISSING THE PROGRAM ASKS THE OPERATOR TO
C      ENTER THEM
   13   WRITE(*,'(2X,A18,'' = '',\)') HNAME(I)
        READ(*,*,ERR=13) ( XYZ(J,I), J = 1, 3 )
      ELSE
C     IF COORDINATES ARE THERE THE PROGRAM ASKS THE OPERATOR TO CHECK
C      THEM AND REPLACE THEM IF INCORRECT
   14   WRITE(*,'(2X,A18,'' =  '',3F9.1,A,\)') HNAME(I),
     +     ( XYZ(J,I),J=1,3), ' ARE THESE CORRECT? (Y/N)  '
        READ(*,'(A1)') ANS
        IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-')THEN
   15     WRITE(*,'(2X,A18,'' = '',\)') HNAME(I)
          READ(*,*,ERR=15) ( XYZ(J,I), J = 1, 3 )
        ELSE
          IF(ANS.NE.'Y'.AND.ANS.NE.'y'.AND.ANS.NE.'+') GO TO 14
        END IF
      END IF
   16 CONTINUE
        INQUIRE(FILE='B:SN'//A//'.YS',OPENED=THERE)
        IF(THERE)THEN
C     IF THE HEADBOARD DATA FILE HAS BEEN OPENED CORRECTLY REWIND IT
          REWIND(3)
        ELSE
```

```
C     IF THE HEADBOARD DATA DATA FILE WAS NOT OPENED CORRECTLY CLOSE
C      IT THEN OPEN IT CORRECTLY
          CLOSE(3)
          OPEN(3,FILE='B:SN'//A//'.YS')
        END IF
C     WRITE THE HEADBOARD DATA TO THE HEADBOARD DATA FILE
        WRITE(3,'(F12.1)') XYZ
        CLOSE(3)
       GO TO 4
       END IF
       IF(NFILES.EQ.1) THEN
C     IF ONE HEADBOARD FILE WITH AN INCORRECT NAME WAS SHOWN IN THE
C      DIRECTORY USE THAT NAME
        WRITE(*,'(A\)') ' TYPE IN THE FILE NAME PLEASE  '
        READ(*,'(A8)') FNAME
        OPEN(3,FILE='B:'//FNAME//'.YS',STATUS='OLD',ERR=10)
C     GO BACK TO LINE #2 AND TRY TO READ THAT FILE
        GO TO 2
       END IF
       IF(NFILES.GT.1) THEN
C     IF MORE THAN ONE HEADBOARD FILE WITH AN INCORRECT NAME WAS
C      SHOWN IN THE DIRECTORY SELECT ONE OF THE FILES AND USE THAT
C      NAME
        WRITE(*,'(A,I4,A/A\)') ' THERE ARE', NFILES,
     +    ' FILES ON THIS DISK'
     + , ' TYPE IN THE FILE NAME OF THE FILE TO BE READ  '
        READ(*,'(A8)') FNAME
        OPEN(3,FILE='B:'//FNAME//'.YS',STATUS='OLD',ERR=10)
C     GO BACK TO LINE #2 AND TRY TO READ THAT FILE
        GO TO 2
       END IF
       RETURN
       END
       SUBROUTINE SMALL(A)
C -- THIS SUBROUTINE WRITES THE DATA FILES USING BINARY WRITES
       INTEGER*2 IDATA(30,3,7), IER(30,7), IXYZ
       INTEGER*2 ISEX, NC(7), NCX(7), NVS(7), NCNAME, NCEXP, KT(7)
       INTEGER*4 NSUB
       CHARACTER*18 NAME(7,2), EXP(7,2)*127, A*5, FILE*22, BEEP*1
       CHARACTER*1 NUM(7), CNAME(126), CEXP(889), JUNK*127, ESC*3
       COMMON/SML/ DATA(30,3,7),ER(30,7),IXYZ(3,28),ISUB(7,2),IS(7,2)
       COMMON/SMLNA/ NAME, EXP
       LOGICAL THERE
C     INITIALIZE
       DATA NUM / '1','2','3','4','5','6','8' /
       DATA NVS / 26, 12, 20, 23, 18, 30, 3 /
       BEEP = CHAR(7)
       BIG = 3276.7
       Z = 0.0
       NXYZ = 1
       ESC = ' '//CHAR(27)//'['
       FILE = 'A:\STATION\SN'//A//'.CON'
C     OPEN FILE ON DATA DISK TO WRITE SUBJECT'S DATA ON
       OPEN(10,FILE=FILE,FORM='BINARY',STATUS='NEW',IOSTAT=IOS)
```

```
      IF(IOS.NE.0) THEN
C     THE PROGRAM SHOULD NOT GET HERE
C     IF FOR SOME REASON IT DOES I WANT A PRINTER RECORD
C     I WILL BELIEVE IT WHEN I SEE IT
         WRITE(*,'(2X,A1,A/3X,2A/3X,A/3X,A/3X,A,A1)') BEEP,
     +    'THE FILE FOR THIS SUBJECT HAS BEEN WRITTEN ON THE DATA DISK',
     +    'FILE =', FILE, 'PLEASE PUT PAPER IN THE PRINTER',
     +    'THEN USE THE SHIFT AND PRINT SCREEN KEYS TO COPY THE SCREEN TO
     + THE PRINTER', 'GIVE THE COPY TO PHIL', BEEP
         RETURN
      END IF
      ISEX = 0
      IEXP = 1
      NSUB = ISUB(1,1)
      ISEX = IS(1,1)
      NCNAME = 0
      NCEXP = 0
      DO 14 I = 1, 7
C     GO THRU LOOP #14 ONCE FOR EACH STATION
         IF(ISEX.EQ.0.AND.IS(I,1).GT.0.AND.IS(I,1).LT.3) ISEX = IS(I,1)
C     CHECK FOR AN EXPLANATION
      IF(EXP(I,1).NE.'OK'.AND.EXP(I,1).NE.'BLANK'.AND.EXP(I,1).NE.' ')
     +    IEXP = -1
      JUNK = NAME(I,1)
C     FIND LENGTH OF MEASURER'S NAME
      CALL COMP(JUNK,18,NC(I) )
C     SUM LENGTH OF MEASURER'S NAMES
      NCNAME = NCNAME + NC(I)
   14 CONTINUE
      IST = 0
      DO 16 I = 1, 7
      IST = IST + 1
      CNAME(IST) = NUM(I)
      DO 16 K = 1, NC(I)
      IST = IST + 1
C     MOVE EACH CHARACTER OF EACH MEASURER'S NAME THAT IS USED INTO
C      CNAME
   16 CNAME( IST ) = NAME(I,1)(K:K)
      NCNAME = NCNAME + 7
      IF(IEXP.LT.0) THEN
C     THERE IS ONE OR MORE EXPLANATION OTHER THAN 'OK', 'BLANK', '  '
         DO 18 I = 1, 7
C     CALL COMP TO FIND THE ACTUAL LENGTH OF EACH EXPLANATION
          CALL COMP(EXP(I,1),127,NCX(I) )
C     SUM THE LENGTH OF THE EXPLANATIONS
   18     NCEXP = NCEXP + NCX(I)
      IST = 1
      DO 20 I = 1, 7
      IST = IST + 1
      CEXP(IST) = NUM(I)
      DO 20 K = 1, NCX(I)
        IST = IST + 1
C     MOVE EACH CHARACTER OF EACH EXPLANATION THAT IS USED INTO CEXP
   20 CEXP( IST ) = EXP(I,1)(K:K)
```

```
      NCEXP = NCEXP + 7
      END IF
      DO 24 I = 1, 7
      KT(1) = 1
      DO 24 L = 1, 30
        DO 22 K = 1, 3
C     SET ANY OUT OF RANGE VALUES TO FIT IN INTEGER*2
        IF(DATA(L,K,I).GT.BIG) DATA(L,K,I) = BIG
C     SET KT(I) TO 2 OR 3 IF ANY VARIABLES WERE REMEASURED
        IF(DATA(L,K,I).GT.Z.AND.KT(I).LT.K) KT(I) = K
C     CONVERT TO INTEGER*2 INCLUDING FIRST DECIMAL PLACE
   22   IDATA(L,K,I) = DATA(L,K,I) * 10 + 0.1
      IF(ER(L,I).GT.BIG) ER(L,I) = BIG
   24 IER(L,I) = ER(L,I) * 10 + 0.1
      IF(NCEXP.LT.1) NCEXP = 1
C     WRITE ALL INFORMATION TO DATA DISK IN BINARY
      WRITE(10) NSUB, ISEX, ( KT(L), L = 1, 7),
     + (((IDATA(I,K,L),I = 1, NVS(L) ),K = 1, KT(L) ), L = 1, 7 ),
     + ((IER(I,L),I = 1, NVS(L) ),L = 1, 7 ), NCNAME,
     + ( CNAME(I), I = 1, NCNAME ), NCEXP, ( CEXP(I), I = 1, NCEXP),
     + NXYZ, IXYZ
      CLOSE(10)
C     CLOSE FILE AND RETURN TO DISKIN
   99 RETURN
      END
      SUBROUTINE COMP(CHS,NCH,NACH)
C     SUBROUTINE COMP READS BACKWARDS THRU A CHARACTER ARRAY UNTIL A
C      NON BLANK CHARACTER IS FOUND
C     THEN RETURNS THE LENGTH OF THE NAME OR EXPLANATION
      INTEGER*2 NACH
      CHARACTER*127 CHS
      BLANK = ' '
      DO 2 I = NCH, 1, -1
      IF(CHS(I:I).NE.BLANK) THEN
        NACH = I
C     RETURN TO SMALL WHEN NON BLANK CHARACTER IS FOUND
        RETURN
      END IF
    2 CONTINUE
C     BLANK NAME OR EXPLANATION
      NACH = 0
      RETURN
      END
      SUBROUTINE READSML
C     SUBROUTINE READSML READS THE BINARY FILES WRITTEN BY SUBROUTINE
C      SMALL
      INTEGER*2 ISEX,IDATA(30,3,7),IER(30,7),NCNAME,NCEXP
      INTEGER*2 IXYZ(3,28), KT(7)
      INTEGER*4 NSUB
      LOGICAL THERE
      CHARACTER*1 CNAME(126), CEXP(889), CH, ANS, A*5, FILE*22
      CHARACTER*18 NAME(7), EXP(7)*127, BEEP*1, CLEAR*5
      DIMENSION DATA(30,3,7), ER(30,7), XYZ(3,28), NVS(7)
      DATA NVS / 26, 12, 20, 23, 18, 30, 3 /
```

```
      BEEP = CHAR(7)
      CLEAR = ' '//CHAR(27)//'[2J'
C     CLEAR THE SCREEN
      WRITE(*,'(A5)') CLEAR
    1 WRITE(*,'(3X,A\)') 'TYPE IN SUBJECT NUMBER YOU WANT TO SEE  '
      READ(*,*,ERR=1) J
C     READ IN THE SUBJECT NUMBER FOR THE FILE YOU WANT TO SEE
C     THEN WRITE THE SUBJECT NUMBER INTO CHARACTER VARIABLE A
      IF(J.LT.0) RETURN
      WRITE(A,'(I5.5)') J
      FILE = 'A:\STATION\SN'//A//'.CON'
C     SEE IF THE FILE IS ON THE DATA DISK
      INQUIRE(FILE=FILE,EXIST=THERE)
      IF(THERE) THEN
C     IF THE FILE IS THERE OPEN AND READ WITH BINARY READ
      OPEN(9,FILE=FILE,FORM='BINARY',STATUS='OLD')
      READ(9,END=2)NSUB,ISEX,( KT(L), L = 1, 7 ),
     + (((IDATA(I,K,L),I=1,NVS(L) ),K = 1, KT(L) ), L = 1, 7 ),
     + ((IER(I,L),I = 1, NVS(L) ), L = 1, 7 ),
     + NCNAME, ( CNAME(I),I = 1, NCNAME ), NCEXP,
     + ( CEXP(I),I = 1, NCEXP ), NXYZ, IXYZ
C     CONVERT DATA, ERROR DATA, AND HEADBOARD DATA FROM INTEGER TO
C      REAL WITH ONE DECIMAL PLACE
    2 DO 4 I = 1, 30
      DO 4 L = 1, 7
      DO 3 K = 1, 3
    3 DATA(I,K,L) = IDATA(I,K,L) / 10.0
    4 ER(I,L) = IER(I,L) / 10.0
      DO 6 I = 1,28
      DO 6 K = 1, 3
    6 XYZ(K,I) = IXYZ(K,I) / 10.0
      IST = 1
      DO 10 I = 2, 7
      IFR = IST + 1
      CH = CHAR( I + 48 )
      IF(I.EQ.7) CH = '8'
        DO 8 K = 1, 18
          IST = IST + 1
          IF(CNAME(IST).EQ.CH) GO TO 10
C     SEPARATE EACH STATION'S MEASURER'S NAME
    8   CONTINUE
   10 WRITE(NAME( I - 1 ),'(18A1)') ( CNAME(L), L = IFR, IST - 1 )
      IF(NCNAME - IST.GT.19) NCNAME = IST + 19
      WRITE(NAME(7),'(18A1)') ( CNAME(L), L = IST + 1, NCNAME )
      IF(NCEXP.GT.1) THEN
C     IF NCEXP IS GREATER THAN ONE THEN THERE ARE SOME EXPLANATIONS
C      OTHER THAN 'OK', 'BLANK', OR '  '
        IST = 1
        DO 14 I = 2, 7
        IFR = IST + 1
        IF(IFR.GE.NCEXP) WRITE(*,'(3X,A)') 'GARBAGE!!!!!'
        CH = CHAR( I + 48 )
        IF(I.EQ.7) CH = '8'
        DO 12 K = 1, 127
```

```
        IST = IST + 1
C     SEPARATE EACH STATION'S EXPLANATION
          IF(CEXP(IST).EQ.CH) GO TO 14
   12   CONTINUE
   14   WRITE(EXP( I - 1 ),'(127A1)') ( CEXP(L), L = IFR, IST - 1 )
        IF(NCEXP - IST.GT.127) NCEXP = IST + 127
        WRITE(EXP(7),'(127A1)') ( CEXP(L), L = IST + 1, NCEXP )
       ELSE
C     IF NCEXP EQUALS ONE SET EACH STATION'S EXPLANATION TO 'OK'
        DO 16 I = 1, 7
        EXP(I) = 'OK'
C     EXCEPT IF THE STATION'S DATA IS BLANK THEN SET THE EXPLANATION
C      TO 'BLANK'
   16   IF(DATA(1,1,I).EQ.0.0) EXP(I) = 'BLANK'
       END IF
       DO 18 I = 1, 7
       II = I
       IF(II.EQ.7) II = 8
       WRITE(*,'(3X,A,I2,A)') 'STATION', II, '    PRESS THE ENTER KEY TO
      +VIEW THE NEXT STATION '
C     WRITE DATA FOR ONE STATION TO THE SCREEN
       WRITE(*,'( I7,2X,A18,I3/1X,A79)') NSUB, NAME(I), ISEX, EXP(I)
       WRITE(*,'(10F8.1)') ( DATA(K,1,I), K = 1, NVS(I) )
       WRITE(*,'(10F8.1)') ( DATA(K,2,I), K = 1, NVS(I) )
       WRITE(*,'(10F8.1)') ( DATA(K,3,I), K = 1, NVS(I) )
       WRITE(*,'(10F8.1)') ( ER(K,I), K = 1, NVS(I) )
       READ(*,'(A1)') ANS
C     WAIT FOR A RESPONSE FROM THE KEYBOARD BEFORE WRITING THE NEXT
C      STATION TO THE SCREEN
   18  WRITE(*,'(A5)') CLEAR
       WRITE(*,'(3X,A)') 'HEADBOARD DATA'
       WRITE(*,'(10(3(2X,3F8.1)/))') XYZ
       CLOSE(9)
       ELSE
C     FILE WAS NOT FOUND TRY AGAIN OR ENTER -1 TO RETURN TO THE MAIN
C      MENU
         WRITE(*,'(2X,A1,A,A22,A)') BEEP, 'FILE ', FILE, ' NOT FOUND'
         GO TO 1
       END IF
   20  WRITE(*,'(3X,A)') 'READ ANOTHER SUBJECT''S FILE?  (Y/N)  '
       READ(*,'(A1)') ANS
       IF(ANS.EQ.'N'.OR.ANS.EQ.'n'.OR.ANS.EQ.'-') RETURN
C     Y TO READ ANOTHER SUBJECT
C     N TO RETURN TO MAIN MENU
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y'.OR.ANS.EQ.'+') THEN
         WRITE(*,'(A5)') CLEAR
         GO TO 1
       END IF
       WRITE(*,'(1X,A1)') BEEP
       GO TO 20
       END
```

```
1  '        Subject select program 9/28/87 (C) PFA
2  PRINT@85,"Survey program V3.8 9/28/87"' size stored numbers
10 ' INITIALIZE
12 CLEAR 500,MAXRAM :MAXFILES=1 :DEFINT A-Z
20 DIM X1%(7,4)' Input table
21 DIM X2%(7,4)' counter table
22 DIM X3%(7,4)' Measured table
23 DIM X4$(4)'   age names
24 DIM X5$(7)'   category names
25 DIM X6%(7,4)' Screened table
26 DIM X7%(7,4)' All screened table
27 DIM X8%(7,4)' Total measured table
28 E1$=CHR$(27) :E2$=E1$+"p" :E3$=E1$+"q"' highlight
30 P1$=" Enter a first letter: " :P2$=" [A]ccept   [R]etry   [S]kip "
:P3$=" Female " :P4$=" Male " :P6$="\      \" :P7$="  Enter [N]ext
screen or [R]eturn"
40 FOR I=0 TO 3 :READ A$ :X4$(I)=A$ :NEXT I' load ages
42 FOR I=0 TO 6 :READ A$ :X5$(I)=A$ :NEXT I' load categories
44 DATA " < 21"," 21-24"," 25-30","  > 30"
46 DATA "White","Black","Hispanic","Pacific","Indian","Mixed","Other"
49 'get last values for All screened
50 ON ERROR GOTO 58
51 OPEN "ALLS.DO" FOR INPUT AS 1
52 LINEINPUT#1,A$:LINEINPUT#1,A$:FOR I=0 TO 6:INPUT#1,A$: FOR J=0 TO 3
53  INPUT#1,A:   X7%(I,J)=A
54  NEXT J: NEXT I
56 CLOSE 1:GOTO60
58 CLOSE:IF ERR>50 THEN RESUME 60 ELSE GOTO 9000
59 'get last values for Total measured
60 ON ERROR GOTO 68
61 OPEN "TOTALM.DO" FOR INPUT AS 1
62 LINEINPUT#1,A$:LINEINPUT#1,A$:FOR I=0 TO 6: INPUT#1,A$: FOR J=0 TO 3
63  INPUT#1,A:   X8%(I,J)=A
64  NEXT J: NEXT I
66 CLOSE 1:GOTO80
68 CLOSE:IF ERR>50 THEN RESUME 80 ELSE GOTO 9000
69 '
79 '.................MAIN MENU.............................
80 ON ERROR GOTO 9000
82 CLS :PRINT :PRINT " [L]oad Parameters" :PRINT " [S]elect Sample"
:PRINT " [D]isplay values " :PRINT " [Q]uit and save Tables":PRINT "
[H]elp Screens"
84 PRINT P1$;
86 ON INSTR(" LlSsDdQqHh",INPUT$(1))\2 GOSUB 100,1000,2000,3000,4000
88 GOTO 82
89 '
99 '.................LOAD SUBROUTINE.......................
100 CLS :PRINT :PRINT" "; P3$,P4$ :PRINT P1$; :P5=INSTR("
FfMm",INPUT$(1))\2
112 IF P5=1 THEN P5$=P3$ ELSE IF P5=2 THEN P5$=P4$ ELSE BEEP:GOTO 100
115 PRINT@121," Maximum number to be Measured";:INPUT"";MN:IF MN<2 THEN
BEEP:GOTO 115
```

```
120 FOR I=0 TO 6  :FOR J=0 TO 3
130 CLS :PRINT :PRINT " For ";E2$;" "X5$(I);E3$; P5$;"aged
";E2$;X4$(J);E3$;" years "
131 IF X1%(I,J)=0 THEN150' no value assigned, treat as accept all
132 IF X1%(I,J)=-1 THEN PRINT"   Reject all Screened":GOTO 138
133 IF X1%(I,J)= 1 THEN PRINT"    Accept all Screened":GOTO 138
134 IF X1%(I,J)<0 THEN N=-X1%(I,J):PRINT"    Reject 1 in";N:GOTO 138
135 PRINT"    Accept 1 in";X1%(I,J)
138 PRINT " Change? [Y]es or [N]o";
139 ON INSTR(" YyNn",INPUT$(1))\2 GOTO 150,830:BEEP:GOTO 139
149 'accept new values
150 PRINT@121,"Select [L]ess, or [G]reater than
half,":PRINT@168,"[N]one, or [A]ll ?":ON INSTR(" LlGgNnAa",INPUT$(1))\2
GOTO200,300,400,500:BEEP:GOTO150
199 ' select less than half
200 INPUT " Accept 1 in  ";A$:A=VAL(A$)
210 IF A<0 THEN 900 ' check for exit (for troubleshooting only)
220 IF A=>MN THEN 400'reject all
290 GOTO800' load table with A
299 'select more than half
300 INPUT " Reject 1 in  ";A$:A=VAL(A$)
310 IF  A<0 THEN 900 ' check for exit
320  IF A>MN THEN A=1:GOTO340' accept all
330 A=-A ' store reject as negative number
340 GOTO800' load table with -A
399 'reject all
400 A=-1
410 GOTO 800
499 'accept all
500 A=1
510 GOTO800
799 'load in array
800 X1%(I,J)=A'
830 NEXT J :NEXT I
899 '
900 RETURN' done with selection
998 '
999 '.................SELECT SAMPLE SUBROUTINE.............
1000 IF NN<0 THEN RETURN' catch quit trap
1010 CLS :PRINT @280," Use negative number to return to menu"; :PRINT
@41,"Enter Age:";:LINE INPUT""; A$:A=VAL(A$)
1011 IF A<0 THEN 1190' check for exit
1012 IF A>70 OR A<15 THEN 1010' sort age into range
1014 IF A>30 THEN A=3 :GOTO 1020
1016 IF A>24 THEN A=2 :GOTO 1020
1017 IF A>20 THEN A=1 :GOTO 1020
1018 A=0
1020 FOR I= 0 TO 3 : PRINT "  ";X5$(I); :NEXT :PRINT
1022 FOR I= 4 TO 6 : PRINT "  ";X5$(I); :NEXT :PRINT
1030 PRINT P1$;
1032 B=INSTR(" WwBb4hPpIiMmOo", INPUT$(1))\2-1 :IF B<0 THEN BEEP: GOTO
1032
1100 PRINT :PRINT " category= ";E2$;" "X5$(B);E3$;", age range=
";E2$;X4$(A);E3$
```

```
1110 PRINT P2$;
1120 ON INSTR(" AaRrSs",INPUT$(1))\2 GOTO 1200,1000,1600 :GOTO1120
1190 RETURN
1199 ' check for selection
1200 IF X1%(B,A)=-1 THEN GOSUB 1400 :GOTO 1000' reject all
1203 IF X1%(B,A)=1 OR X1%(B,A)=0 THEN GOSUB 1300:GOTO 1000' accept all
1204 IF X1%(B,A)<0 THEN GOTO 1250' count skips
1210 X2%(B,A)=X2%(B,A)+1' inc. cell in skipper table
1220 IF X2%(B,A)=1 THEN GOSUB 1300 ELSE GOSUB 1400' accept or reject
1230 IF X2%(B,A)=>X1%(B,A) THEN X2%(B,A)=0' reset skipper cell
1240 GOTO 1000' selection complete, get next
1249 ' negative values=greater than half
1250 X2%(B,A)=X2%(B,A)+1' inc. cell in skipper table
1270 IF X2%(B,A)=>-(X1%(B,A)) THEN X2%(B,A)=0: GOSUB 1400 ELSE  GOSUB
1300' reject and reset skipper cell or accept
1290 GOTO 1000' selection complete, get next
1299 ' selected
1300 PRINT@240, E2$;"  *** SELECTED ***  ";E3$;"                    "'
message
1320 FOR K=1 TO 3 : SOUND 2348,10 : SOUND 23,10 :NEXT K' sound a
"selected" tone
1330 X3%(B,A)=X3%(B,A)+1' add to measured table
1335 GOSUB 1700' add to screened table
1338 GOSUB 1800' add to All screened
1339 GOSUB 1900' add to Total measured
1340 GOSUB 1500' prompt
1349 'check for max subjects
1350 NN=NN+1 :IF NN<MN THEN RETURN
1351 CLS:PRINT@85, MN;" subjects measured":PRINT" [Q]uit or [C]ontinue"
1352 ON INSTR(" QqCc",INPUT$(1))\2 GOTO 1354,1356:BEEP:GOTO1352
1354 NN=-1:RETURN' set quit trap
1356 NN=0:RETURN' allow more subjects
1399 ' rejected'
1400 PRINT@240, "               ";E2$; "  *** REJECTED ***  ";E3$'
message
1420 SOUND 9394,16' sound a "rejected" tone
1425 GOSUB 1700' add to screened tally
1428 GOSUB 1800' add to All Screened
1430 GOSUB 1500' prompt
1490 RETURN
1499 ' prompt
1500 PRINT@280,STRING$(39," ");:PRINT@280," Hit any key for next
subject"; :A$=INPUT$(1)
1510 RETURN
1599 'handle Skipped subjects
1600 PRINT@240, "               ";E2$; "  *** SKIPPED ***  ";E3$'
message
1620 SOUND 9394,16' sound a "rejected" tone
1625 GOSUB 1700' add to screened tally
1628 GOSUB 1800' add to All screened
1630 GOSUB 1500' prompt
1690 GOTO1000' get next subject
1699 ' update screened tally
1700 X6%(B,A)=X6%(B,A)+1
```

```
1720 RETURN
1799 'update x7
1800 X7%(B,A)=X7%(B,A)+1
1810 RETURN
1899 'update x8
1900 X8%(B,A)=X8%(B,A)+1
1910 RETURN
1998 '
1999 '.....DISPLAY SUBROUTINE.....
2000 CLS: PRINT@41,"[I]nput":PRINT" [S]creened    [A]ll screened":PRINT"
[M]easured    [T]otal measured":PRINT" [R]eturn":PRINT P1$;
2020 ON  INSTR(" IiSsMmAaTtRr",INPUT$(1))\2 GOSUB
2100,2200,2300,2400,2500,2040
2040 RETURN
2099 ' display inputs
2100 CLS :PRINT " Inputs    "; :FOR I=0 TO 3 : PRINT X4$(I); :NEXT I
:PRINT' header line
2120 FOR J=0 TO 6 : PRINT USING P6$; X5$(J);' print categories
2130   FOR K=0 TO 3 :  PRINT USING "  ####"; X1%(J,K); : NEXT K' print
input cells
2140 IF J<6 THEN PRINT
2142 NEXT J
2150 A$=INPUT$(1)' hold display until key
2190 RETURN
2199 'display subjects screened
2200 CLS :PRINT " Screened "; :FOR I=0 TO 3 : PRINT X4$(I); :NEXT I
:PRINT' header line
2220 FOR J=0 TO 6 : PRINT USING P6$; X5$(J);' print categories
2230   FOR K=0 TO 3 :  PRINT USING "  ####"; X6%(J,K); : NEXT K' print
screened cells
2240 IF J<6 THEN PRINT
2242 NEXT J
2250 A$=INPUT$(1)
2290 RETURN
2299 'display subjects measured
2300 CLS :PRINT " Measured "; :FOR I=0 TO 3 : PRINT X4$(I); :NEXT I
:PRINT' header line
2320 FOR J=0 TO 6 : PRINT USING P6$; X5$(J);' print categories
2330   FOR K=0 TO 3 :  PRINT USING "  ####"; X3%(J,K); : NEXT K' print
measured cells
2340 IF J<6 THEN PRINT
2342 NEXT J
2350 A$=INPUT$(1)
2390 RETURN
2399 'display all subjects screened
2400 CLS :PRINT " All     "; :FOR I=0 TO 3 : PRINT X4$(I); :NEXT I
:PRINT' header line
2420 FOR J=0 TO 6 : PRINT USING P6$; X5$(J);' print categories
2430   FOR K=0 TO 3 :  PRINT USING "  ####"; X7%(J,K); : NEXT K' print
input cells
2440 IF J<6 THEN PRINT
2442 NEXT J
2450 A$=INPUT$(1)
2490 RETURN
```

```
2499 'display total subjects measured
2500 CLS :PRINT " Total     "; :FOR I=0 TO 3 : PRINT X4$(I); :NEXT I
:PRINT' header line
2520 FOR J=0 TO 6 : PRINT USING P6$; X5$(J);' print categories
2530   FOR K=0 TO 3 :  PRINT USING "  ####"; X8%(J,K); : NEXT K' print
input cells
2540 IF J<6 THEN PRINT
2542 NEXT J
2550 A$=INPUT$(1)
2590 RETURN
2998 '
2999 '.....QUIT AND SAVE ROUTINE .....
3000 CLS:PRINT@45,"Wait, Saving all Files":PRINT@85,"Saving LOG.DO"
3010 CLOSE:OPEN "LOG.DO" FOR APPEND AS 1
3019 ' write input table
3020 PRINT#1,DATE$;"  ";TIME$;"  ";P5$
3030 PRINT#1, USING P6$;"input"; :PRINT#1, " "; :FOR I=0 TO 4 : PRINT
#1,X4$(I); :NEXT I :PRINT#1," "
3040 FOR J=0 TO 6 : PRINT#1, USING P6$; X5$(J);
3050   FOR K=0 TO 3 :  PRINT#1, USING "  ####"; X1%(J,K); : NEXT K
3060  PRINT#1," " :NEXT J :PRINT#1," "
3119 'write screened table
3120 PRINT#1, USING P6$;"screened"; :PRINT#1, " "; :FOR I=0 TO 4 : PRINT
#1,X4$(I); :NEXT I :PRINT#1," "
3140 FOR J=0 TO 6 : PRINT#1, USING P6$; X5$(J);
3150   FOR K=0 TO 3 :  PRINT#1, USING "  ####"; X6%(J,K); : NEXT K
3160  PRINT#1," " :NEXT J :PRINT#1," " :PRINT#1," "
3219 'write measured table
3220 PRINT#1, USING P6$;"measured"; :PRINT#1, " "; :FOR I=0 TO 4 : PRINT
#1,X4$(I); :NEXT I :PRINT#1," "
3240 FOR J=0 TO 6 : PRINT#1, USING P6$; X5$(J);
3250   FOR K=0 TO 3 :  PRINT#1, USING "  ####"; X3%(J,K); : NEXT K
3260  PRINT#1," " :NEXT J :PRINT#1," " :PRINT#1," "
3290 CLOSE 1:PRINT@125,"Saving ALLS.DO"
3299 ' save ALLS.DO
3300 OPEN "ALLS.DO" FOR OUTPUT AS 1
3312 PRINT#1," All Screened by "; DATE$
3320 PRINT#1, USING P6$;""; :PRINT#1, " "; :FOR I=0 TO 4 : PRINT
#1,X4$(I); :NEXT I :PRINT#1," "
3340 FOR J=0 TO 6 : PRINT#1, USING P6$; X5$(J);
3350   FOR K=0 TO 3 :  PRINT#1, USING ", ####"; X7%(J,K); : NEXT K
3360  PRINT#1," " :NEXT J :PRINT#1," "
3390 CLOSE 1:PRINT@165,"Saving TOTALM.DO"
3399 ' save TOTALM.DO
3400 OPEN "TOTALM.DO" FOR OUTPUT AS 1
3412 PRINT#1," Total Measured by "; DATE$
3420 PRINT#1, USING P6$;""; :PRINT#1, " "; :FOR I=0 TO 4 : PRINT
#1,X4$(I); :NEXT I :PRINT#1," "
3440 FOR J=0 TO 6 : PRINT#1, USING P6$; X5$(J);
3450   FOR K=0 TO 3 :  PRINT#1, USING ", ####"; X8%(J,K); : NEXT K
3460  PRINT#1," " :NEXT J :PRINT#1," "
3490 CLOSE 1
3499 '
3900 MENU
```

```
3999 '.....................HELP SCREENS.............................
4000 CLS :PRINT" page 1 ":PRINT" ":PRINT" ":PRINT" ":PRINT" ":PRINT"
":PRINT" ":PRINT P7$;:A$=INPUT$(1) :IFA$<>"N" AND A$<>"n" THEN 4500
4020 CLS :PRINT" page 2 ":PRINT" ":PRINT" ":PRINT" ":PRINT" ":PRINT"
":PRINT" ":PRINT P7$;:A$=INPUT$(1) :IFA$<>"N" AND A$<>"n" THEN 4500
4030 CLS :PRINT" page 3 ":PRINT" ":PRINT" ":PRINT" ":PRINT" ":PRINT"
":PRINT" ":PRINT P7$;:A$=INPUT$(1) :IFA$<>"N" AND A$<>"n" THEN 4500
4040 'more help screens
4500 RETURN
9000 PRINT"Error "ERR" on Line "ERL :STOP
```

```
10 'x(j) are the actual dimensions read in for each station from the
20 'subject diskette
30 'y(j) are the dimensions read in from the second file
40 'x(j) and y(j) are used to compute the deltas, which are d(j)
50 'when deltas are read back in to compute means, it is done through
60 ' a nested do-loop, in which each dimension is run through, for
70 'each station.  Here, the deltas are dd(k,i)
80 's(k) is the running sum, when calculating the means
90 'mean(k) is the mean for each dimension
100 'dim$(k,i) is the dimension name; ae(k,i) is the allowable error
110 'a(j),b(j),c(j) are dummy variables to get through the reading of
the file.
120 'nv is the number of dimensions at a station when reading data in
130 'nv(i) is the number of dimensions when the stations are handled
140 'as a group in a loop
150 DIM X(30),A(30),B(30),C(30),D(30),Y(30),AE(30,7)
160 DIM DD(30,30),S(30),MEAN(30),DIMS$(30,7)
170 '
180 '*****************************Read in dimension names,************
190 '*****************************numbers of variables,***************
200 '*****************************station numbers, allowable errors***
210 '
220 FOR I=1 TO 7
230   READ STA$(I)
240 NEXT I
250 FOR I=1 TO 7
260   READ NV(I)
270 NEXT I
280 FOR I=1 TO 7
290   FOR K=1 TO NV(I)
300     READ DIMS$(K,I)
310   NEXT K
320 NEXT I
330 FOR I=1 TO 7
340   FOR K=1 TO NV(I)
350     READ AE(K,I)
360   NEXT K
370 NEXT I
380 '
390 '*****************************Menu*********************************
400 '
410 CLS
420 FOR N=1 TO 5
430 PRINT
440 NEXT N
450 PRINT TAB(10) "This program calculates delta values between the
first and "
460 PRINT TAB(10) "second measures.  It stores the values and, at a
later time, "
470 PRINT TAB(10) "calculates and prints the mean value."
480 PRINT
490 PRINT
```

```
500 PRINT TAB(15) "1. Read values from subject diskettes"
510 PRINT
520 PRINT TAB(15) "2. Calculate and print means"
530 PRINT
540 PRINT TAB(15) "3. Delete files before starting a new week"
550 PRINT
560 PRINT TAB(15) "4. End"
570 PRINT
580 INPUT "What is your selection";Q
590 ON Q GOTO 650,1380,2440,2780
600 '
610 '*****************************Input of subject number*************
620 '*****************************and station number of***************
630 '*****************************remeasured station******************
640 '
650 CLS
660 INPUT "What is the subject number (-1 to end)";SN$
670 IF VAL(SN$)=-1 THEN GOTO 410
680 INPUT "What is the station number which has been remeasured";STA$
690 IF VAL(STA$)=1 THEN NV=26
700 IF VAL(STA$)=2 THEN NV=12
710 IF VAL(STA$)=3 THEN NV=20
720 IF VAL(STA$)=4 THEN NV=23
730 IF VAL(STA$)=5 THEN NV=18
740 IF VAL(STA$)=6 THEN NV=30
750 IF VAL(STA$)=8 THEN NV=3
760 '
770 '*****************************Create file names and***************
780 '*****************************open files for reading data*********
790 '
800 B$="b:re0"
810 IF VAL(SN$)>=10000 THEN B$="b:re"
820 C$="."
830 A$=B$+SN$+C$+STA$
840 BB$="b:sn0"
850 IF VAL(SN$)>=10000 THEN BB$="b:sn"
860 AA$=BB$+SN$+C$+STA$
870 ON ERROR GOTO 2600
880 ON ERROR GOTO 2600
890 OPEN "i",1,A$:ON ERR GOTO 2600
900 INPUT #1,SN,S1$
910 FOR K=1 TO NV
920 INPUT #1,X(K)
930 NEXT K
940 FOR K=1 TO NV
950 INPUT #1,A(K)
960 NEXT K
970 FOR K=1 TO NV
980 INPUT #1,B(K)
990 NEXT K
1000 FOR K=1 TO NV
1010 INPUT #1,C(K)
1020 NEXT K
1030 CLOSE #1
```

```
1040 OPEN "i",1,AA$
1050 INPUT #1,SN,S1$
1060 FOR K=1 TO NV
1070 INPUT #1,Y(K)
1080 NEXT K
1090 FOR K=1 TO NV
1100 INPUT #1,A(K)
1110 NEXT K
1120 FOR K=1 TO NV
1130 INPUT #1,B(K)
1140 NEXT K
1150 FOR K=1 TO NV
1160 INPUT #1,C(K)
1170 NEXT K
1180 CLOSE #1
1190 '
1200 '*****************************Calculate deltas*********************
1210 '
1220 FOR K=1 TO NV
1230 D(K)=ABS(X(K)-Y(K))
1240 NEXT K
1250 '
1260 '*****************************Create file name and*****************
1270 '*****************************open file for writing data***********
1280 '
1290 CC$="a:delta."
1300 DD$=CC$+STA$
1310 OPEN "a",1,DD$
1320 FOR K=1 TO NV
1330 PRINT #1,D(K)
1340 NEXT K
1350 CLOSE #1
1360 GOTO 660
1370 '
1380 '*****************************Opens files containing***************
1390 '*****************************deltas and reads them in*************
1400 '
1410 CC$="a:delta."
1420 FOR I=1 TO 7
1430   EE$=CC$+STA$(I)
1440 OPEN "i",#1,EE$
1450 FOR N=1 TO 30
1460 IF EOF(1) THEN 1510
1470 FOR K=1 TO NV(I)
1480 INPUT #1,DD(K,N)
1490 NEXT K
1500 NEXT N
1510 CLOSE #1
1520 '
1530 '*****************************Calculate means**********************
1540 '
1550 T=N-1
1560 FOR K=1 TO NV(I)
1570 S(K)=0
```

```
1580 FOR N=1 TO T
1590 S(K)=S(K) + DD(K,N)
1600 NEXT N
1610 MEAN(K)=S(K)/T
1620 NEXT K
1630 '
1640 '*****************************Print tables************************
1650 '
1660 LPRINT "Station " STA$(I) " Deltas";
1670 LPRINT TAB(70); DATE$
1680 LPRINT
1690 LPRINT
1700 LPRINT
1710 LPRINT "Dimension"; TAB(23);
1720 LPRINT "D e l t a s"; TAB(70); "N"; TAB(73);
1730 LPRINT "Mean"; TAB(79); "AE"
1740 FOR K=1 TO NV(I)
1750 LPRINT DIMS$(K,I); TAB(20);
1760 FOR N=1 TO T
1770 LPRINT USING "###";DD(K,N);
1780 NEXT N
1790 LPRINT TAB(69);
1800 LPRINT USING "##"; T;
1810 LPRINT TAB(71);
1820 LPRINT USING "###.##";MEAN(K);
1830 LPRINT TAB(79);
1840 LPRINT USING "##"; AE(K,I)
1850 NEXT K
1860 LPRINT CHR$(12)
1870 NEXT I
1880 GOTO 410
1890 '
1900 '*****************************Data statements*********************
1910 '
1920 DATA 1,2,3,4,5,6,8
1930 DATA 26,12,20,23,18,30,3
1940 DATA "Stature","Cervicale Height","Neck Ht Lateral","Acromial
Height"
1950 DATA "Axilla Height","Suprasternale Ht","Chest Height","Tenth Rib
Height"
1960 DATA "Waist Height NI", "Iliocristale Ht", "Waist Height OMPH"
1970 DATA "Waist Breadth", "Waist Depth", "Hip Breadth", "Chest Depth"
1980 DATA "Chest Breadth", "Bstpt/T-Bstpt/T Br", "Acrom-Radiale Lgth"
1990 DATA "Radiale-Stylion L", "Shoulder-Elbow Lgth", "Forearm-Hand
Lgth"
2000 DATA "Axillary Arm Circ","Elbow Circ","Wrist Circ","Biceps Circ
Flexed"
2010 DATA "Forearm Circ Flex","Thumbtip Reach","Wrist-Wall Length"
2020 DATA "Wrist-Wall Lgth Ext","Ovhd Fgtp Rch","Ovhd Fgtp Rch
Ext","Span"
2030 DATA "Ovhd Fgtp Rch Sit","Lat Fem Epicond Ht","Knee Ht Midpatella"
2040 DATA "Calf Height","Lat Malleolus Ht","Wrist Ht Sitting","Functnl
Leg Lgth"
2050 DATA "Crotch Length OMPH", "Crotch Length NI","VTC (ASCC)", "VTC
```

```
(USA)"
2060 DATA "Waist NI-Waist OMP","Waist-Hip Length","Buttock Circ","Thigh
Circ"
2070 DATA "Gluteal Furrow Ht","Buttock Height","Trochanteric Ht","Wrist
Height"
2080 DATA "Crotch Height","Buttock Depth","Bispinous Breadth"
2090 DATA "Bimalleolar Brdth","Heel Breadth","Crotch L Post OMPH"
2100 DATA "Crotch L Post NI","Head Circumference","Bitrag Coronal Arc"
2110 DATA "Bitrag Crinion Arc","Bitrag Frontal Arc","Bitrag Subnasl Arc"
2120 DATA "Bitragion Chin Arc","Bitrag Submand Arc","Bizygomatic Brdth"
2130 DATA "Head Length","Head Breadth","Menton-Sellion L","Ear Length"
2140 DATA "Ear Lgth abve Trag","Ear Breadth","Ear Protrusion"
2150 DATA "Interpupil Breadth","Thumb Breadth","Wrist-Thmbtip Lgth"
2160 DATA "Wrst-Ctr of Grip L","Hand Length","Wrist-Indx Fingr L"
2170 DATA "Hand Breadth","Hand Circumference","Sitting Height"
2180 DATA "Cervicale Ht Sit","Midshoulder Ht Sit","Acromial Ht Sit"
2190 DATA "Eye Height Sitting","Elbow Rest Height","Waist Ht Sit NI"
2200 DATA "Waist Ht Sit OMPH","Thigh Clearance","Knee Ht Sitting"
2210 DATA "Popliteal Height","Buttock-Knee Lgth","Buttock-Pop Lgth"
2220 DATA "Hip Brdth Sitting","Biacromial Breadth","Bideltoid Breadth"
2230 DATA "Forearm-Forearm Br","Abdom Ext Dpth Sit","Weight"
2240 DATA "Neck Circumference","Neck Circ Base", "Shoulder Circ"
2250 DATA "Chest Circ at Scye","Chest Circ","Chest C below Brst"
2260 DATA "Waist Circ NI","Waist Circ OMPH","Waist Front L NI"
2270 DATA "Waist Front L OMPH","Strap Length","Neck-Bstpt/T Lgth"
2280 DATA "Shoulder Length","Interscye II","Interscye I","Scye Depth"
2290 DATA "Waist Back Lgth NI","Waist Back L OMPH","Scye Circumference"
2300 DATA "Sleeve Outseam","Sleeve L Spne-Scye","Sleeve L Spne-Elbw"
2310 DATA "Sleeve L Spne-Wrst","Lower Thigh Circ","Knee Circumference"
2320 DATA "Calf Circumference","Ankle Circ","Heel Ankle Circ"
2330 DATA "Ball of Foot Circ","Foot Length","Ball of Foot Lgth"
2340 DATA "Foot Br Horizontal"
2350 DATA 10,7,7,7,10,5,11,5,4,5,7,6,8,7,4,8,10,4,6,6,4,8,4,5,6,5
2360 DATA 20,20,20,20,20,10,20,3,6,3,3,10
2370 DATA 17,18,16,22,24,3,6,12,6,6,7,7,11,10,8,3,2,2,11,11
2380 DATA 5,7,5,5,6,8,6,2,2,2,3,2,2,3,3,2,2,3,4,3,4,2,4
2390 DATA 6,10,9,9,8,10,6,8,3,2,7,6,7,6,8,8,17,10
2400 DATA .3,6,11,22,15,15,16,11,12,5,5,12,8,3,13,10,4,5,5,13,6,11,10,9
2410 DATA 4,4,5,4,6,4
2420 DATA 3,6,2
2430 '
2440 '**************************Delete all files**********************
2450 '
2460 CLS
2470 FOR N=1 TO 5
2480 PRINT
2490 NEXT N
2500 PRINT TAB(10) "Are you sure (Y,N)?"
2510 INPUT " ",Q$
2520 IF (Q$= "Y" OR Q$= "y") THEN GOTO 2530 ELSE 410
2530 CC$="a:delta."
2540 FOR I=1 TO 7
2550 EE$=CC$+STA$(I)
2560 KILL EE$
```

```
2570 NEXT I
2580 GOTO 410
2590 '
2600 '*****************************Error Routine*************************
2610 '
2620 FOR N=1 TO 5
2630 PRINT
2640 NEXT N
2650 PRINT TAB(10) "There is a difficulty in locating one or more files"
2660 PRINT TAB(10) "on the diskette.  This could be a result of
mistyping"
2670 PRINT TAB(10) "the subject number or the station number.  Please "
2680 PRINT TAB(10) "recheck the numbers and try it again."
2690 PRINT
2700 PRINT TAB(10) "If this is already a second try, abandon this
subject"
2710 PRINT TAB(10) "diskette and go to the next one."
2720 PRINT
2730 PRINT TAB(10) "When you are ready to proceed, hit the ENTER key."
2740 INPUT " ",Q2$
2750 RESUME 650
2760 '
2770 '*****************************End************************************
2780 END
```

Hardware Specifications Required to Support Survey Software

1. Program INED runs on a Compaq® portable computer, with 256K RAM, 2 floppy disk drives and the MS-DOS® operating system. It may run on other computers which support MS-DOS®, but has not been tested on other machines. The Compaq® is paired, in the field, with a Brother® M-1109 printer, but several other dot matrix printers have performed well in brief tests.

2. Program INOUT8 runs on Compaq II® portable computer, with 640K RAM, 2 floppy drives and the MS-DOS operating system. It may run on the Compaq® with 256K, and it may run on other computers which support MS-DOS®, but has not been tested on these machines. As with INED, the program works well with a Brother® M-1109 printer and some others, but may work with all dot matrix printers.

3. Program DELTAS has the same hardware requirements as INOUT8, but requires additionally the GW-BASIC® interpreter. Other requirements are identical to INOUT8.

4. SURVEY38 runs on a Radio Shack® TRS-80 with 32K RAM. It requires a BASIC interpreter, which is standard equipment on the TRS-80. This software has not been tested on other hardware, and is unlikely to work well on other units, because the BASIC in the TRS-80 is a special version for that computer.

APPENDIX C.

Allowable Observer Error

APPENDIX C.

Allowable Observer Error

Control of observer error in anthropometric surveys is a critical factor in the reliability of the resulting data. Nevertheless, data on observer error have been only infrequently collected during such surveys because to do so is costly, time-consuming, and often a source of irritation for measurer and subject alike. The problem is particularly acute in military surveys, since subject time is always in lieu of duty time. Some studies of interobserver error have been conducted, but usually only after the survey has been completed. These data are useful in assessing results, but cannot affect them. For the 1987-1988 survey, the U.S. Army set limits for observer error in advance, and tracked measurer performance throughout the data collection phase of the survey.

The acceptable observer errors listed in this appendix had four uses. First they were used in training to indicate when measurers had learned their tasks. Second, they were used for identifying measurer drift which might have occurred during the year-long survey. Third, the allowable errors were used as a measure against which daily interobserver error data were checked throughout the course of the data collection.

The final use of allowable errors was in the software installed on the computer at each measuring station. There, the allowable errors were used as part of the algorithm to check whether a remeasured value was different in a significant way from the originally measured value. The allowable errors for each dimension are listed in Table C-1.

TABLE C-1. Allowable Observer Error.

Dimension	Allowable Error (in mm)
Abdominal Extension Depth, Sitting	10
Acromial Height	7
Acromial Height, Sitting	9
Acromion-Radiale Length	4
Ankle Circumference	4
Axilla Height	10
Axillary Arm Circumference	8
Ball of Foot Circumference	4
Ball of Foot Length	6
Biacromial Breadth	8
Biceps Circumference, Flexed	6
Bideltoid Breadth	8
Bimalleolar Breadth	2
Bispinous Breadth	3
Bitragion Chin Arc	8
Bitragion Coronal Arc	7
Bitragion Crinion Arc	5
Bitragion Frontal Arc	5
Bitragion Submandibular Arc	6
Bitragion Subnasale Arc	6
Bizygomatic Breadth	2
Bustpoint/Thelion-Bustpoint/Thelion Breadth	10
Buttock Circumference	12
Buttock Depth	8
Buttock Height	7
Buttock-Knee Length	6
Buttock-Popliteal Length	7
Calf Circumference	5
Calf Height	3
Cervicale Height	7
Cervicale Height, Sitting	10
Chest Breadth	8
Chest Circumference	15
Chest Circumference at Scye	15
Chest Circumference below Breast	16
Chest Depth	4
Chest Height	11
Crotch Height	10
Crotch Length (Natural Indentation)	16
Crotch Length (Omphalion)	18
Crotch Length, Posterior (Natural Indentation)	11
Crotch Length, Posterior (Omphalion)	11

TABLE C-1. (cont'd)

Dimension	Allowable Error (in mm)
Ear Breadth	3
Ear Length	2
Ear Length above Tragion	2
Ear Protrusion	3
Elbow Circumference	4
Elbow Rest Height	10
Eye Height, Sitting	8
Foot Breadth, Horizontal	2
Foot Length	3
Forearm Circumference, Flexed	5
Forearm-Forearm Breadth	17
Forearm-Hand Length	4
Functional Leg Length	17
Gluteal Furrow Height	6
Hand Breadth	2
Hand Circumference	4
Hand Length	3
Head Breadth	2
Head Circumference	5
Head Length	2
Heel Ankle Circumference	6
Heel Breadth	2
Hip Breadth	7
Hip Breadth, Sitting	6
Iliocristale Height	5
Interpupillary Breadth	2
Interscye I	10
Interscye II	13
Knee Circumference	4
Knee Height, Midpatella	6
Knee Height, Sitting	2
Lateral Femoral Epicondyle Height	3
Lateral Malleolus Height	3
Lower Thigh Circumference	4
Menton-Sellion Length	3
Midshoulder Height, Sitting	9
Neck-Bustpoint/Thelion Length	8
Neck Circumference	6
Neck Circumference, Base	11
Neck Height, Lateral	7
Overhead Fingertip Reach	20
Overhead Fingertip Reach, Extended	20

TABLE C-1. (cont'd)

Dimension	Allowable Error (in mm)
Overhead Fingertip Reach, Sitting	20
Popliteal Height	7
Radiale-Stylion Length	6
Scye Circumference	13
Scye Depth	4
Shoulder Circumference	22
Shoulder-Elbow Length	6
Shoulder Length	3
Sitting Height	6
Sleeve Length: Spine-Elbow	10
Sleeve Length: Spine-Scye	11
Sleeve Length: Spine-Wrist	9
Sleeve Outseam	6
Span	10
Stature	10
Strap Length	12
Suprasternale Height	5
Tenth Rib Height	5
Thigh Circumference	6
Thigh Clearance	3
Thumb Breadth	2
Thumbtip Reach	20
Trochanteric Height	7
Vertical Trunk Circumference (ASCC)	22
Vertical Trunk Circumference (USA)	24
Waist Back Length (Natural Indentation)	5
Waist Back Length (Omphalion)	5
Waist Breadth	6
Waist Circumference (Natural Indentation)	11
Waist Circumference (Omphalion)	12
Waist Depth	8
Waist Front Length (Natural Indentation)	5
Waist Front Length (Omphalion)	5
Waist Height (Natural Indentation)	4
Waist Height (Omphalion)	7
Waist Height, Sitting (Natural Indentation)	6
Waist Height, Sitting (Omphalion)	8
Waist-Hip Length	6
Waist (Natural Indentation) - Waist (Omphalion)	3
Weight	0.3 kg
Wrist-Center of Grip Length	4
Wrist Circumference	5

TABLE C-1. (cont'd)

Dimension	Allowable Error (in mm)
Wrist Height	11
Wrist Height, Sitting	10
Wrist-Index Finger Length	4
Wrist-Thumbtip Length	3
Wrist-Wall Length	20
Wrist-Wall Length, Extended	20

APPENDIX D.

The Biographical Data Questionnaire

APPENDIX D.

US ARMY ANTHROPOMETRIC SURVEY (ANSUR)

BIOGRAPHICAL DATA: MILITARY HISTORY

TODAY'S DATE: ...../...../..... TODAY'S POST: .....
Month Day Year

1. Name: .....
(Last) (First) (Middle)

2. Unit to which you are assigned at this post:

...../...../...../.....
(Company) (Battalion/Battery/Group) (Brigade/Regiment) (Division)

3. Military Component: /...../ Regular Army
/...../ Army Reserve
/...../ National Guard

4. Military Personnel Class:
/...../ Enlisted
/...../ Warrant Officer (Specify Branch: .....)
/...../ Commissioned Officer (Specify Branch: .....)

5. Rank/Grade: ..... / ..... (e.g., LTC / 05)

6. Time in Service: ..... Years, ... Months (e.g., 2 Years, 4 Months)

7. MOS: ..... (Primary) ..... (Secondary)

8. With which hand do you usually fire a weapon?
/...../ Right /...../ Left /...../ Either Hand

9. With which eye do you usually sight your weapon?
/...../ Right /...../ Left /...../ Either Eye

NATICK Form 1 (TEMP)
1 Aug 87

US ARMY ANTHROPOMETRIC SURVEY (ANSUR)

BIOGRAPHICAL DATA: PERSONAL HISTORY

1. Your Birthdate: ...../...../.....
(Month) (Day) (Year)

2. Age: ..... Years

3. Sex: /...../ Male
/...../ Female

4. Race: /...../ White, not of Hispanic origin
/...../ Black, not of Hispanic origin
/...../ Hispanic
/...../ Asian/Pacific Islander
/...../ American Indian/Alaskan Native
/...../ Mixed (Specify: .....)
/...../ Other (Specify: .....)

5. How tall are you in bare feet? .....' ....." (e.g., 5' 8")
Feet Inches

6. How much do you weigh without clothes? ..... Pounds

7. Do you wear: /...../ Prescription Glasses?
/...../ Prescription Contact Lenses?
/...../ Both?
/...../ Neither?

8. With which hand do you usually write?
/...../ Right /...../ Left /...../ Either Hand

9. *Do you currently participate in resistance or free-weight training at least once a week?*

/...../ Yes

/...../ No

If you answered "No", go to question 10.

If you answered "Yes", complete questions 9a, 9b, and 9c.

a. How long have you been involved in this training?

..... Years, ... Months (Example: 2 Years, 7 Months)

b. How many days per week do you now train?

Upper body: ..... Days per week

Lower body: ..... Days per week

c. On the days that you train, how many hours per day do you train?

Upper body: ..... Hours per day

Lower body: ..... Hours per day

10. Do you currently run on a regular basis?

/...../ Yes

/...../ No

If you answered "No", go to question 11.

If you answered "Yes", complete questions 10a, 10b, and 10c.

a. How long have you been running?

..... Years, ... Months (Example: 3 Years, 9 Months)

b. How many days per week do you now run?

..... Days per week

c. On the days that you run, how many miles do you usually cover?

..... Miles

11. Your Birthplace: .....

12. Mother's Birthplace: .....

13. Father's Birthplace: .....

14. Mother's Race: /...../ White, not of Hispanic origin
/...../ Black, not of Hispanic origin
/...../ Hispanic
/...../ Asian/Pacific Islander
/...../ American Indian/Alaskan Native
/...../ Mixed (Specify: .....)
/...../ Other (Specify: .....)

15. Father's Race: /...../ White, not of Hispanic origin
/...../ Black, not of Hispanic origin
/...../ Hispanic
/...../ Asian/Pacific Islander
/...../ American Indian/Alaskan Native
/...../ Mixed (Specify: .....)
/...../ Other (Specify: .....)

DO NOT WRITE BELOW THIS LINE

---

16. Ethnicity/National Extraction:

Subject: .....

Mother: .....

Father: .....

17. Body Dimensions

Actual Height .....

Actual Weight .....