UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

A Computer Program Designed to Produce Tables From Alphanumeric Data

By

Jennie Ridgley and Robert W. Schnabel

Open-file Report 78- 875

1978

Contents

Abstract

This program is designed to produce tables from alphanumeric data. Each line of data that appears in the table is entered into a data file as a single line of data. Where necessary, a predetermined delimiter is added to break up the data into column data. The program can process the following types of data: (1) title, (2) headnote, (3) footnote, (4) two levels of column headers, (5) solid lines, (6) blank lines, (7) most types of numeric data, and (8) all types of alphanumeric data. In addition, the program can produce a series of continuation tables from large data sets.

Fitting of all data to the final table format is performed by the program, although provisions have been made for user modification of the final format. The width of the table is adjustable, but may not exceed 158 characters per line.

The program is useful in that it permits alteration of original data or table format without having to physically retype all or portions of the table. The final results may be obtained quickly using interactive terminals, and execution of the program requires only minimal knowledge of computer usage. Tables produced may be of publishable quality, especially when reduced. Complete user documentation and program listing are included.

NOTE: Although this program has been subjected to many tests a warranty on accuracy or proper functioning is neither implied nor expressed.

Introduction

This computer program produces tables from alphanumeric data. The program contains an option for producing a single table from the data or, if there are too many data, for producing a set of continuation tables. The maximum width of any table is 158 characters, but is limited to the width of the terminal being used or to 136 characters if printed on the line printer.

The program is written in PL/1 (Multics PL/1 Reference Manual) and uses several of the Multics standard subroutines (Multic's Programmer's Manual Subroutines). Thus it may not be transportable to other computers unless these subroutines are compensated for.

The program was designed to be interactive and the user does not need to know PL/1 in order to use it. However, if the user wishes to understand the source code, a knowledge of PL/1 is assumed.

The program is useful in that it permits addition of data to a data set, correction of original data entries, and respacing of column widths without the user having to physically retype all or portions of the table. Processing the data through the program will produce a new table within minutes.

The following section contains user documentation that discusses the general format of a table, the types of data that can be manipulated, and the execution of the program. Samples of data input and program execution and table output may be found in Appendixes A and B. Appendix C contains a listing of the

User documentation

General discussion

In determining the format of a table, the program first examines the value entered for the final table width. This value should be an estimate of the final table width. If this estimated width is too small, the program tells the user the minimum width needed to run the program. If the width chosen exceeds that of the output device, the individual lines of data will be continued at the beginning of the next line. The result is a staggered output and the table is not usable.

In determining the minimum width of the table, the program finds the longest word in each column of data, including column headers, and adds two characters to the word size. The program then computes a sum of the new word lengths. This sum is the minimum table width. Next the program computes the excess number of characters between the chosen table width and the computed minimum table width. It then prorates, in a predetermined manner, these excess characters over all the columns. There is a provision for adjusting the column widths if the final computed column widths are not satisfactory. This provision is discussed in the section on program execution. If the user believes that the final table is too wide for publication, satisfactory results have been achieved by photographically reducing the table or using a reducing photocopying machine.

As mentioned previously, the program is able to produce a set of continuation tables if the data set is too long. The user

program.

specifies, at execution, the number of lines per output page. This value includes the number of lines allocated to title and column-header information. The program will then repeat all title and column-header information at the top of each subsequent page. In addition, the character string "--Continued" is appended to the title. If the page option is chosen, the first line of data on each page represents a new line of data, because continuation lines of data are not broken across pages. If the first column of the first data line on each subsequent page is blank, the program prints, in that position, the appropriate line identifier. If any column on the first data line on each subsequent page contains data of the form, do or Do, the program will automatically repeat the original data in that column. When subtitle data are included in continuation tables, the value entered for the number of lines per output page should be such that a new subtitle and bounding dashed or solid lines will not be the last data on the page.

In general the program is able to process title data, headnote and footnote data, two levels of column-header data, alphanumeric data, and most types of numeric data. In addition, the program will permit blank and solid lines to be printed.

Title and headnote data lines are centered over the table. Continuation lines of these data types are also centered over the table. Subtitle data lines cannot be longer than the width of the table minus eleven. Column headers are centered within their column widths. Most alphanumeric data, which are not column

headers, begin at the left margin of the column. If the data exceeds the column width, the remainder is continued on subsequent lines. Each subsequent line is indented two spaces. The exception to this is the use of a one- or two-character alphanumeric string, which begins with a character that is not a number or letter. This character string is centered within the column.

The program permits the inclusion, as data items, the following special alpha strings:

NA, ND, do, Do.

The alpha strings NA and ND are generally used in tables to refer to the unavailability of data or to the lack of data. Reference to their meaning is usually found in the headnote. The alpha strings do and Do. stand for ditto. The program treats these alpha strings in the following manner: the strings NA and ND are centered within the column; the strings do and Do. are centered within the column if the column width is less than nine spaces; otherwise they are indented three spaces.

Numeric data are centered within the column. All integer data are right-justified, all decimal data are centered about the decimal point, and all mixed integer and decimal data within a column are aligned with respect to the decimal point.

Numeric data that cannot be processed by the program include (1) ratio data of the form X:Y or X/Y and (2) range data of the form X-Y. The handling of these various types of data is discussed more fully in the section on data set-up.

The main limitation in using this program is the width of the table. If the final table is too wide, the user should divide his data into two separate tables or into two portions and run each portion separately. The result of the latter choice is a table that spans two pages. It should be noted that the two halves of this table may have different numbers of continuation lines on each page. In order to make data lines match across the center of the table, the user must enter an editing system and add blank lines in the appropriate places. In addition, the title will have to be split and centered across the entire table or repeated on the second page with the string, "--Continued", appended. The final results may then be reduced to page size. This technique should be used only with data sets that will fit on a table not longer than one page.

Data set-up

The data to be processed are placed in a disk file, which may have any name. There are 12 different ways in which lines of data may be interpreted. In order to correctly identify the type of data on each line, one of the following capital letters, T, X, S, H, L, W, B, U, D, Q, A, or I must be the first letter on each line. The order of data lines in the data file will determine the final table format. Discussed below is the use of each of these letters. The discussion covers the type of data processed, the way in which the data are entered into the data file, and the way in which the program presents the data. It is important to remember that the data to be processed must be entered in stream format, not record format. When entering data from a terminal, each line of data must be terminated with a line feed character. To separate the input data into columns, a delimeter is used between adjacent input values. This delimeter is the two-character alpha string " /". In the discussion below the reference to the use of a delimeter will refer to this character string. Examples of data input and output are found in Appendixes A and B.

T -- title data

-- The complete title immediately follows T. It is typed in as a single line regardless of length.

-- On output the title is centered over the table. If the title length exceeds the table width, the remainder of the title is continued on subsequent lines and is

centered on each line. If the page option is chosen, the title is repeated on all subsequent pages with the character string, "--Continued", appended.

X -- headnote data

-- Complete headnote data follows X. It is typed in as a single line regardless of length. When necessary, enclosing square brackets, [ ], should be included in the data.

-- On output the headnote is centered over the table. Any continuation lines are also centered.

S -- primary column headers, used only when two levels of column headers are needed.

-- Enter all column headers as a single line of data immediately after S. The character string for each column header should end with a delimeter. If there is no primary column header for one or more secondary column headers, enter a delimeter where needed.

-- On output, primary column headers will be spaced over the table and centered over the appropriate secondary column headers. The delimeter entered for blank column headers permits output of blank spaces over the appropriate secondary column headers. Lengthy column headers are continued on subsequent lines; each continuation line is centered over the appropriate columns.

H -- primary column headers, if only one set of headers is

used; secondary headers, if two levels of column headers are used.

-- Enter all column headers on a single line immediately following the H. Each column header should end in a delimeter. There should be no blank column headers.

-- On output, column headers will be spaced across the table. Each column header will be centered within the appropriate column. Lengthy column headers are continued on subsequent lines; each continuation line is centered within the appropriate column.

L -- raw numeric or alphanumeric data

-- Each line of data is entered immediately after L. For each line of data, data for each column end with a delimeter. There should be the same number of delimiters per line of data as there were column headers entered after H. If there are no data for a particular column, enter that data item as a delimeter. Padding of raw-data items with leading and trailing spaces is not necessary and, if used, will create problems in output.

-- On output, all numeric data are centered within the column. All integer data are right-justified, all decimal data are centered about the decimal point, and mixed integer and decimal data within a column are justified with respect to the decimal position. Data in column 1 and most alphanumeric data begin at the

leftmost margin of the column. Any continuation lines of these data types are indented two spaces. An exception to this provision is the use of data in the form of one or more characters in an alphanumeric string, where the first character is not a number or letter. This type of data is centered within the column.

The program also contains an option for adding leaders, in the form of trailing periods or underscores, after the last character in the data string. Generally leaders are added to the data in any column, except the last one, that is not a numeric or symbol data column. However the user has the option of specifying which columns are to have leaders added. One or more leaders are added, to within three spaces of the right-column margin. (Note: On some terminals the underscore is printed as a solid line, instead of a dashed line. If a dashed line is desired, the table should be printed on the line printer. Otherwise the use of trailing periods, as leaders, is recommended.) If the alphanumeric string which is to have leaders appended exceeds the width of the column, a single period is added after the last word.

F -- footnote data

-- Complete footnote data follows. It is typed in as a single line regardless of length.

-- On output, the footnote is indented two spaces on the first line. Continuation lines are started at the left

margin of the table.

W -- blank-line data

-- Type a line feed following W.

-- On output, the program prints a blank line. This feature is used for breaking up lines of data into smaller, more easily readable groups.

B -- blank-line data, inserted after title

-- Enter a line feed after B.

-- The program prints two blank lines after the title. B differs from W in that if the page option is selected, the coding for B permits storage of the blank lines so that they may be printed on subsequent pages.

U -- underscore line between primary and secondary column headers

-- Enter at least two underscore characters.

-- On output the program prints a solid line (dashed if printed on the line printer) between the primary and secondary column headers. The length of each line segment is the width of the primary column minus one. A single space is allowed between line segments to help delineate columns. If there is no primary column header for one or more secondary column headers, the program omits the line segment.

D -- solid underscore line

-- Enter at least two dash characters.

-- On output, a solid (dashed on the line printer)

underscore line is printed before all column headers, after all column headers, or at the end of the data set.

Q -- provides counter data, used only in conjunction with the first internal subtitle data

-- Enter a line feed after Q. Note: A D line of data must precede it and a T (title) line of data must follow.

-- There is no visual output; it is used as a counter when subtitle information is included.

A -- used in conjunction with subsequent changes in internal subtitle information

-- Enter a line feed after A. A D line of data must precede it and a T (title) line of data must follow.

-- There is no visual output; it is used as a counter in changing the output of subsequent subtitle information.

I -- used in conjunction with subsequent changes in internal subtitle information

-- Enter a line feed after I. This line of data must precede any D line of data that precedes an A or Q line of data (see table 3).

-- There is no visual output; it is used to cahnge output of subsequent subtitle information where a change in the subtitle occurs at the top of a new page.

Execution

In order to execute the program, the user must create a link to the object segment of the program or copy the source program and create an object segment using the PL/1 compiler. An object segment and source code of this program are stored in the author's area and will be available to any user. A link to it may be created with the following command:

```
link >udd>Chama>JRidgley>PL1>table
```

Once this is done, in order to run the program the object segment must be invoked. Upon execution the program prompts the user with a series of questions. The user types in the requested data followed by a line feed or carriage return. Each question and the necessary user response, as well as program execution, are discussed.

1. Enter name of output file

   Enter the name of the output file.

2. Enter width of table

   Enter total number of characters per line, not to exceed 158.

3. Enter name of input segment

   Enter the name of your data file.

4. Enter number of columns

   Enter number of primary-level columns, or of second-level columns if two levels are used.

5. Enter number of header columns

   Enter number of columns of primary-level columns when

two levels are used; otherwise enter 0(zero).

6. Enter number of lines per page

   This is the page option. Enter number of lines per page; value should account for all title and column-header information. If pagination is desired this value should not exceed 50, as the resultant tables can be more easily reproduced or reduced. If pagination is not desired, make this value sufficiently large to permit all data to be printed as one page.

7. Enter total number of columns to get leaders

   Enter the value of the total number of columns to get leaders. Generally this value will be at least 1, but may be greater or may be 0 if no leaders are desired.

8. Enter column number to get leaders

   Enter the value of the column, followed by a carriage return or line feed. This question is repeated the number of times specified by the value entered in response to question 6.

The next question is asked only if the value entered in response to question 8 is not zero.

9. What type of leader do you want? Enter 0 for period. Enter 1 for underscore.

   Enter a zero for period leaders or 1 for underscore leaders.

The next question is asked only if two levels of column headers are to be processed:

10. Enter beginning and ending column numbers that encompass header n

(n is a column number of one of the primary columns). Enter the range of secondary column numbers to be contained under this header. If only one column is to be contained, the beginning and ending column value are the same. Enter as two values, separated by a space, for example, 1 2; terminate each line with a line feed or carriage return.

The program continues execution and prints out the following data:

word (1) is value (1)

•

•

word (n) is value (n),

where n is the total number of columns entered in response to question 3 and value (column number) is the length of the longest word in that column plus two (see "General Discussion").

The program then prints the comment:

wide is value,

where value is the sum of values, including the two added spaces, of the longest word in each column.

The program then compares this total value of wide to the table width entered. If the value of wide exceeds the value of the table width entered, the following message is printed:

minimum width for this table is value,

where value is the computed minimum table width. The program then allows the user to increase the table width. The program prompts the user as follows:

enter new value for table width.

If the value provided by the user is still less than the computed table width the message,

minimum width for this table is wide,

is printed and execution terminates.

If the table-width value entered exceeds the computed table-width value, the excess characters are prorated, in a predetermined manner, over all the columns. After this, the program prints the following message:

column 1 = value (1)

•

•

column n = value (n),

where n is the column number and value (column number) is the final width for that column.

The program then asks:

Are those O.K.?

The user should answer yes or no. If the user answers yes the program completes execution. If the user answers no, the program asks the user to enter new column widths for each column. Limitations on the new values entered are as follows: (1) the new value cannot be less than the original value computed for each column; (2) a value must be supplied for each column, even if

there is to be no change; and (3) the sum of all the new column values should equal the table width. If the sum of the new values entered is less than or greater than the input table width, the program prints a message to the user and asks the user to try again. Upon receiving the correct sum of new column widths, the program completes execution.

There are two other instances in which the program prints a message to the user. When processing lines of column-header data or raw data, the program counts the number of delimiters per line. If it detects the wrong number of delimiters on any line, the following message is printed:

line number has value delimiters,

where number is the number of the line containing the error and value is the number of delimiters detected on that line. After printing the message the program is terminated.

As the pogram reads in each line of data, it checks to make sure that the first letter of the line is a valid code letter. If it is not, the following message is printed:

line starts with the wrong character -- data,

where data is the first 10 letters of the line. The program is then terminated.

Output from the program is placed in a file with the name specified by the user. To print tables on the line printer, issue the command

dp file name,

where file name is the name of the file containing output from

the program. No additional commands are necessary when printing continuation tables on the line printer, as the program inserts all page controls into the output file. If the user possesses a wide-carriage terminal with the option of choosing 12 characters per inch, lines that are 158 characters long may be printed. When printing on these terminals, it is necessary to set the pitch to 12 and to issue the multics command

```
ll 158
```

If the file is to be printed on a wide-carriage terminal and it contains continuation tables, issue the command:

```
stty -modes vertsp
```

The user must then position his printer at the top of the page and enter the control characters that perform pagination for his particular terminal. For example, if the user has an Anderson-Jacobson 832, he should key in, in local mode, CTRL L or ESC F. Every time a page-control character is encountered in the output file, output will be printed on a new page.

References

Ahlbrandt, T. S., in press, Textural parameters of eolian deposits, in McKee, E. D., ed., A study of global sand seas: U. S. Geological Survey Professional Paper 1052.

Lewis, R. Q., and Campbell, R. H., 1965, Geology and uranium deposits of Elk Ridge and vicinity San Juan County, Utah: U. S. Geological Survey Professional Paper 474-B, 69 p.

Multics Programmers' Manual Subroutines, Series 60 (level 68), AG 93.

Multics PL/1 Reference Manual, Series 60 (level 68), AM83.

Appendix A: Samples of input data

Several examples of data sets are included in tables 1 to 5. Program execution and table output using these data sets are found in Appendix B. Copies of the data sets have been reduced, using a reducing photocopier, to page size. Data in each data set was organized so that the table produced follows standard U. S. Geological Survey format. An exception to this is data sets four and five. Data for these data sets were read into the computer from punch cards and have not been altered to conform to standard U. S. Geological Survey style. The only difference in data sets four and five (tables 4 and 5) is in the deletion of the hyphenation of the following words: Sorting, Skewness, and Kurtosis. The difference in output for tables 4 and 5 is shown in tables 17 and 18 in Appendix B.

Table 1.--Data set one

```
TTable 1.--Petrographic data
B
X[--, not computed]
D-
S /Texture / / /
U
HSample No. /Mean grain size(mm) /Mean grain size(phi) /Maximum grain size(mm) /Mean roundness /Standard deviation(mm) /Environment of deposition /Robert Folk
  's scheme /
D
LLBIW-12 /0.32 /1.97 /0.87 /-- /0.174 /Fluvial /Arkose. /
LLBIW-13 /.24 /2.29 /.69 /-- /.011 /Fluvial /Arkose. /
LLBIW-14 /.36 /1.85 /1.06 /-- /.223 /Fluvial /Arkose. /
LLBIW-15 /.52 /1.50 /2.35 /-- /.375 /Fluvial /Arkose. /
LLBIW-16 /.24 /2.41 /1.11 /-- /.184 /Fluvial /Arkose. /
LLBIW-17 /.36 /1.77 /1.02 /-- /.166 /Fluvial /Arkose. /
LLBIW-18 /.31 /2.11 /1.30 /-- /.172 /Fluvial /Arkose. /
W
W
LLV1-1 /.27 /2.22 /.96 /-- /.146 /Fluvial /Arkose. /
LLV1-2 /.44 /1.45 /1.39 /-- /.198 /Fluvial /Arkose. /
LLV1-3 /.16 /2.97 /.83 /.47 /.099 /Fluvial /Arkose. /
LLV1-4 /.74 /.92 /3.17 /-- /.610 /Fluvial /Lithic arkose. /
LLV1-5 /.50 /1.32 /1.48 /-- /.295 /Fluvial /Arkose. /
LLV1-6 /.16 /2.98 /.74 /-- /.987 /Fluvial /Arkose. /
LLV1-7 /.71 /1.16 /8.30 /-- /.790 /Fluvial /Arkose. /
W
W
LLV1-14 /.35 /1.87 /1.11 /-- /.213 /Fluvial /Arkose. /
LLV1-15 /.29 /2.10 /.77 /.47 /.153 /Fluvial /Arkose. /
LLV1-16 /.63 /.96 /3.76 /.51 /.404 /Fluvial /Arkose. /
LLV1-17 /.10 /3.53 /.32 /-- /.048 /Fluvial /Subarkose. /
LLV1-18 /.19 /2.74 /1.09 /-- /.113 /Fluvial /Arkose. /
LLV1-19 /.15 /2.97 /.41 /.66 /.077 /Eolian /Arkose. /
LLV1-20 /.19 /2.73 /.65 /.69 /.113 /Eolian /Arkose. /
W
W
LLV2-1 /.42 /1.84 /1.14 /-- /.261 /Fluvial /Lithic arkose. /
LLV2-2 /.37 /2.03 /1.11 /-- /.247 /Fluvial /Lithic arkose. /
LLV2-3 /.43 /1.67 /1.09 /-- /.296 /Fluvial /Lithic arkose. /
LLV2-4 /.51 /1.25 /1.36 /-- /.237 /Fluvial /Lithic arkose, feldspathic litharenite. /
LLV2-5 /.24 /2.61 /.81 /-- /.187 /Fluvial /Lithic arkose. /
LLV2-6 /.34 /1.83 /3.79 /-- /.201 /Fluvial /Lithic arkose. /
LLV2-7 /.81 /.67 /2.09 /.51 /.522 /Fluvial /Arkose. /
LLV2-8 /.11 /3.48 /.54 /-- /.062 /Marine /Subarkose. /
D
```

Table 2.--Data set two

```
TTable 2.--Petrographic data for limestones
B
X[" , present; --, not present]
D
SSample information /Composition / /
U
HSample No. /Location in section(m above base) /Intraclasts /Oolites /Pellets /Fossils /Micrite /Spar /Chert /Quartz /Feldspar /Rock
   fragments /Robert Folk's scheme /
D
LBIW-11 /130.5 /" /" /-- /" /" /" /-- /" /" /-- /Bio-oosparite. /
LBIW-14 /173. /" /" /-- /" /-- /" /-- /" /" /" /Biosparite. /
LBIW-15 /173.5 /" /-- /-- /" /-- /" /-- /" /" /-- /Biosparite. /
LBIW-16 /173.5 /" /-- /-- /" /-- /" /-- /" /" /-- /Biosparite. /
LBIW-17 /167.5 /" /-- /" /" /" /-- /-- /" /-- /-- /Micrite. /
LBIW-21 /248.5 /" /-- /" /" /" /-- /-- /" /" /-- /Pelmicrite. /
LBIW-23 /372.5 /-- /-- /" /" /" /" /-- /" /-- /-- /Micrite. /
W
W
LLV1-6 /230. /" /-- /" /-- /" /" /" /" /-- /-- /Micrite. /
LLV1-7 /251. /-- /-- /" /" /" /" /-- /" /-- /-- /Biomicrite. /
LLV1-9 /297.5 /" /-- /" /-- /" /" /-- /" /-- /-- /Dismicrite. /
LLV1-11 /316.5 /-- /-- /-- /" /-- /" /" /-- /-- /-- /Chert replacing spar replacing fossils. /
LLV1-12 /317. /-- /-- /-- /-- /-- /" /" /-- /-- /-- /Chert replacing spar. /
LLV1-13 /321.5 /-- /" /" /" /" /" /" /-- /-- /-- /-- /Peldismicrite. /
LLV1-14 /324.5 /-- /-- /" /" /" /" /" /-- /-- /-- /-- /Pelbiomicrite. /
W
W
LLV2-6 /220.5 /" /-- /-- /" /" /" /-- /" /-- /-- /Micrite. /
LLV2-8 /260.5 /" /-- /" /" /" /" /-- /" /" /-- /Intramicrite. /
LLV2-11 /344. /" /-- /" /" /" /" /-- /" /-- /-- /Pelintramicrite. /
D
```

Table 3.--Data set three

```
TTable 2.-Description of fossil localities in the Elk Ridge area
B
X[Numbers refer to collections listed in tables 1 and 3]
D
HColln. No. /Stratigraphic position or lithology /Geographic location /
D
Q
TParadox Member of Hermosa Formation
D
L1 /Uppermost beds, bioclastic limestone /Gypsum Canyon, near mouth. /
L2 /Upper part, fossils in sandy limestone /Do. /
D
A
TUpper Member of Hermosa Formation
D
L3 /Black cherty limestone within 100 ft of top /Mouth of Clearwater Canyon. /
L4 /Light-gray bioclastic limestone, 5-9 ft above colln. 3 /Do. /
L5 /Black chert limestone (above colln. 3) /Do. /
L6 /5-ft bed of dense light-gray limestone, many microfossils /Mouth of Dark Canyon. /
L7 /Sandstone, brown to gray, 20-25 ft thick, above colln. 6 /Mouth of Dark Canyon, about 60 ft above creek level. /
L8 /Cherty limestone with abundant fossils. 200 ft below top /Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W. /
L9 /Limestone within 30 ft of the top of the formation /Intersection of Trail and Dark Canyons. Lat 37°49'45'' N., long 109°55'23'' W. /
L10 /Dark-gray limestone forming top ledge of the formation /Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W. /
L11 /100 ft below colln. 10 /Do. /
L12 /Upper part,/Upper part of Gypsum Canyon. /
D
A
TRico Formation
D
L13 /About 300 ft below top (near base) /Gypsum Canyon. Lat 37°55'56'' N., long 109°57'50'' W. /
L14 /Upper part /Hill on south side of Cross Canyon. Lat 38°00'35'' N., long 109°55'26'' W. /
L15 /Limestone /Cross Canyon, 1¼-2 mi north of intersection with Imperial Valley. /
L16 /do /Do. /
L17 /100-500 ft below top /Cross Canyon. /
L18 /Topmost bed /West end of Ruin Park. /
L19 /Near top /In Peavine Canyon, about midway between its junctions with Kigalia and Dark Canyons. /
L20 /do /Do. /
D
```

Table 4.--Data set four

TTable 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base
B
X[Localities are listed alphabetically by country, and specific area where known. The computer index number and McKee's index number are also listed. ND indic

ates no data.]
D
SLocality / /Mean grain size /Median grain size /Sort- ing /Skew- ness /Kurt- osis /
U
HPrincipal country or area /Specific area /Computer Index No. /McKee's Index No. /phi /mm /phi /mm /rho I./SkI /K'G /
D
LAlgeria /Ouargla /4018 /26 /1.935 /0.2615 /ND /ND /1.560 /-0.471 /0.404 /
LEgypt /Ayn Amur /4130 /767 /-0.683 /1.6055 /ND /ND /0.179 /0.702 /0.668 /
LJordan /Ram Fort /594 /685 /1.403 /0.3789 /1.200 /0.4353 /1.177 /0.318 /0.489 /
LLybia /Fezzan /622 /702 /2.190 /0.2192 /2.200 /0.2176 /0.812 /0.015 /0.507 /
L /do /639 /703 /2.350 /0.1961 /2.490 /0.1780 /0.710 /0.240 /0.492 /
W
L /do /556 /727 /2.370 /0.1934 /2.350 /0.1961 /0.971 /0.006 /0.422 /
L /do /616 /728 /1.570 /0.3368 /1.290 /0.4090 /1.180 /0.316 /0.421 /
L /do /4106 /729 /1.754 /0.2965 /ND /ND /0.913 /0.117 /0.439 /
L /do /4107 /729 /1.578 /0.3349 /ND /ND /0.936 /0.425 /0.462 /
L /do /595 /730 /2.333 /0.1989 /3.000 /0.1250 /1.630 /0.208 /0.371 /
W
L /do /4109 /730 /2.279 /0.2060 /ND /ND /1.569 /-0.758 /0.334 /
L /do /597 /731 /2.213 /0.2161 /2.000 /0.2500 /1.456 /0.169 /0.360 /
L /do /4111 /731 /1.684 /0.3112 /ND /ND /1.392 /0.541 /0.361 /
L /do /561 /732 /2.003 /0.2500 /1.700 /0.3078 /1.420 /0.261 /0.366 /
L /do /573 /733 /1.839 /0.2793 /1.400 /0.3789 /1.253 /0.435 /0.382 /
W
L /do /585 /734 /1.450 /0.3660 /0.700 /0.6156 /1.348 /0.860 /0.441 /
L /do /4115 /734 /1.155 /0.4491 /ND /ND /1.170 /0.725 /0.731 /
L /do /523 /735 /2.886 /0.1350 /3.680 /0.1560 /1.305 /0.641 /0.486 /
L /do /4117 /735 /2.412 /0.1879 /ND /ND /1.408 /-0.565 /0.362 /
L /do /564 /736 /1.563 /0.3392 /1.400 /0.3789 /0.848 /0.087 /0.346 /
W
L /do /524 /737 /2.670 /0.1571 /3.400 /0.0947 /1.386 /0.667 /0.356 /
L /do /598 /738 /1.230 /0.4263 /0.350 /0.7846 /1.616 /0.540 /0.668 /
L /do /4121 /739 /2.468 /0.1807 /ND /ND /1.397 /-0.748 /0.363 /
L /do /4122 /739 /2.044 /0.2425 /ND /ND /1.472 /-0.294 /0.343 /
L /do /4123 /740 /1.677 /0.3127 /ND /ND /1.553 /0.067 /0.377 /
W
L /do /4124 /743 /0.961 /0.5137 /ND /ND /1.464 /0.273 /0.459 /
L /do /599 /744 /1.560 /0.3392 /0.800 /0.5743 /1.671 /0.497 /0.333 /
L /do /519 /745 /2.450 /0.1830 /2.850 /0.1387 /1.390 /0.369 /0.342 /
L Vdo /576 /746 /2.000 /0.2500 /1.900 /0.2679 /1.417 /0.082 /0.351 /
L /Ghat /559 /675 /2.006 /0.2483 /2.620 /0.1627 /1.636 /0.448 /0.368 /
W
L /Sabha /553 /664 /1.553 /0.3415 /1.550 /0.3415 /0.940 /0.146 /0.505 /
L /do /525 /667 /2.300 /0.2031 /2.950 /0.1294 /1.588 /0.515 /0.434 /
L /Sedales /4061 /672 /1.769 /0.2934 /ND /ND /1.431 /-0.080 /0.462 /
L /Three Mountains /596 /680 /2.236 /0.2117 /2.000 /0.2500 /1.164 /0.246 /0.391 /
L /do /526 /681 /2.210 /0.2161 /2.150 /0.2253 /1.113 /0.068 /0.370 /
W
LNew Mexico(U.S.A.) /White Sands National Monument /4137 /828 /2.047 /0.2420 /ND /ND /1.035 /0.126 /0.510 /
L /do /4191 /0 /2.702 /0.1537 /ND /ND /0.820 /-0.113 /0.460 /
LPeru /Pampe Corre Viento, Lomitas /4031 /247 /-0.093 /1.0666 /ND /ND /0.694 /0.701 /0.378 /
L /do /4032 /248 /0.787 /0.5795 /ND /ND /1.411 /0.251 /0.387 /
LTunisia /Bir Sultan /1621 /693 /2.416 /0.1882 /2.350 /0.1961 /0.941 /0.056 /0.405 /
D

Table 5.--Data set five

TTable 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base
B
X[Localities are listed alphabetically by country, and specific area where known. The computer index number and McKee's index number are also listed. ND indic

*cates no data.]
D
SLocality / /Mean grain size /Median grain size /Sorting /Skewness /Kurtosis /
U
HPrincipal country or area /Specific area /Computer Index No. /McKee's Index No. /phi /mm /phi /mm /rho I /SkI /K'G /
D
LAlgeria /Ouargla /4018 /26 /1.935 /0.2615 /ND /ND /1.560 /-0.471 /0.404 /
LEgypt /Ayn Amur /4130 /767 /-0.683 /1.6055 /ND /ND /0.179 /0.702 /0.668 /
LJordan /Ram Fort /594 /685 /1.403 /0.3789 /1.200 /0.4353 /1.177 /0.318 /0.489 /
LLybia /Fezzan /622 /702 /2.190 /0.2192 /2.200 /0.2176 /0.812 /0.015 /0.507 /
L /do /639 /703 /2.350 /0.1961 /2.490 /0.1780 /0.710 /0.240 /0.492 /
W
L /do /556 /727 /2.370 /0.1934 /2.350 /0.1961 /0.971 /0.006 /0.422 /
L /do /616 /728 /1.570 /0.3368 /1.290 /0.4090 /1.180 /0.316 /0.421 /
L /do /4106 /729 /1.754 /0.2965 /ND /ND /0.913 /0.117 /0.439 /
L /do /4107 /729 /1.578 /0.3349 /ND /ND /0.936 /0.425 /0.462 /
L /do /595 /730 /2.333 /0.1989 /3.000 /0.1250 /1.630 /0.208 /0.371 /
W
L /do /4109 /730 /2.279 /0.2060 /ND /ND /1.569 /-0.758 /0.334 /
L /do /597 /731 /2.213 /0.2161 /2.000 /0.2500 /1.456 /0.169 /0.360 /
L /do /4111 /731 /1.684 /0.3112 /ND /ND /1.392 /0.541 /0.361 /
L /do /561 /732 /2.003 /0.2500 /1.700 /0.3078 /1.420 /0.261 /0.366 /
L /do /573 /733 /1.839 /0.2793 /1.400 /0.3789 /1.253 /0.435 /0.382 /
W
L /do /585 /734 /1.450 /0.3660 /0.700 /0.6156 /1.348 /0.860 /0.441 /
L /do /4115 /734 /1.155 /0.4491 /ND /ND /1.170 /0.725 /0.731 /
L /do /523 /735 /2.886 /0.1350 /3.680 /0.1560 /1.305 /0.641 /0.486 /
L /do /4117 /735 /2.412 /0.1879 /ND /ND /1.408 /-0.565 /0.362 /
L /do /564 /736 /1.563 /0.3392 /1.400 /0.3789 /0.848 /0.087 /0.346 /
W
L /do /524 /737 /2.670 /0.1571 /3.400 /0.0947 /1.386 /0.667 /0.356 /
L /do /598 /738 /1.230 /0.4263 /0.350 /0.7846 /1.616 /0.540 /0.668 /
L /do /4121 /739 /2.468 /0.1807 /ND /ND /1.397 /-0.748 /0.363 /
L /do /4122 /739 /2.044 /0.2425 /ND /ND /1.472 /-0.294 /0.343 /
L /do /4123 /740 /1.677 /0.3127 /ND /ND /1.553 /0.067 /0.377 /
W
L /do /4124 /743 /0.961 /0.5137 /ND /ND /1.464 /0.273 /0.459 /
L /do /599 /744 /1.560 /0.3392 /0.800 /0.5743 /1.671 /0.497 /0.333 /
L /do /519 /745 /2.450 /0.1830 /2.850 /0.1387 /1.390 /0.369 /0.342 /
L /do /576 /746 /2.000 /0.2500 /1.900 /0.2679 /1.417 /0.082 /0.351 /
L /Ghat /559 /675 /2.006 /0.2483 /2.620 /0.1627 /1.636 /0.448 /0.368 /
W
L /Sabha /553 /664 /1.553 /0.3415 /1.550 /0.3415 /0.940 /0.146 /0.505 /
L /do /525 /667 /2.300 /0.2031 /2.950 /0.1294 /1.588 /0.515 /0.434 /
L /Sedales /4061 /672 /1.769 /0.2934 /ND /ND /1.431 /-0.080 /0.462 /
L /Three Mountains /596 /680 /2.236 /0.2117 /2.000 /0.2500 /1.164 /0.246 /0.391 /
L /do /526 /681 /2.210 /0.2161 /2.150 /0.2253 /1.113 /0.068 /0.370 /
W
LNew Mexico(U.S.A.) /White Sands National Monument /4137 /828 /2.047 /0.2420 /ND /ND /1.035 /0.126 /0.510 /
L /do /4191 /0 /2.702 /0.1537 /ND /ND /0.820 /-0.113 /0.460 /
LPeru /Pampe Corre Viento, Lomitas /4031 /247 /-0.093 /1.0666 /ND /ND /0.694 /0.701 /0.378 /
L /do /4032 /248 /0.787 /0.5795 /ND /ND /1.411 /0.251 /0.387 /
LTunisia /Bir Sultan /1621 /693 /2.416 /0.1882 /2.350 /0.1961 /0.941 /0.056 /0.405 /
D

Appendix B: Sample runs and resultant tables

This section contains examples of program execution and resultant table output of the data sets found in Appendix A. Program execution and the resultant table for data set one are shown in tables 6 and 7.

During execution of the program using data set two (table 8), the program determined that the table-width value entered was too small. The program then prompted the user for a larger table-width value. The output table for data set two is in table 9.

Data set three was used twice. Execution (table 10) the first time was straightforward, and the output is found in table 12. The user decided that the spacing of the last two columns was too far apart and the program was run again (table 11). This time the user changed the width values for columns 2 and 3 but did not change the table width (table 11). The output is shown in table 13. The reader should note that because there is only one level of column headers the value entered for the answer to the question "Enter number of header columns" is zero. In addition, the reader should examine this data set (table 3) to see how to enter data for multiple internal titles. The data for this data set were taken from Lewis and Campbell (1968, table 2).

Data set four was used twice. Execution (table 14) the first time was straightforward, and the output is found in table 15. The user decided that the spacing between the

first two columns was too close (see third line from the bottom of table 15). The user changed the width values for columns 1 and 2 and the program was run again (table 16). Output is found in table 17.

Execution of data set five is found in table 18 and output in table 19. As previously mentioned the only difference between data sets four and five is in the deletion of hyphenation of certain words. Data for data sets four and five were taken from Alhbrandt (in press).

Table 6.--Execution of program using data set one

```
table
Enter name of output file ptro
Enter width of table 110
Enter name of input segment tabu
Enter number of columns 8
Enter number of header columns 4
Enter number of lines per page 50
Enter total number of columns to get leaders 1
Enter column number to get leaders 7
What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 0
Enter beginning and ending column numbers that encompass header          1 1 1
Enter beginning and ending column numbers that encompass header          2 2 6
Enter beginning and ending column numbers that encompass header          3 7 7
Enter beginning and ending column numbers that encompass header          4 8 8

word (          1       ) is            9
word (          2       ) is           10
word (          3       ) is           11
word (          4       ) is           10
word (          5       ) is           11
word (          6       ) is           15
word (          7       ) is           13
word (          8       ) is           14
wide is        93
col             1       =              11
col             2       =              12
col             3       =              13
col             4       =              12
col             5       =              13
col             6       =              17
col             7       =              17
col             8       =              15
Are those OK? yes
r 1330 1.036 23.942 544 level 2, 9
```

Table 7.--Resultant table using data set one

Table 1.--Petrographic data

[--, not computed]

Sample No.	Texture					Environment of deposition	Robert Folk's scheme
	Mean grain size(mm)	Mean grain size(phi)	Maximum grain size(mm)	Mean roundness	Standard deviation(mm)		
LBIW-12	0.32	1.97	0.87	--	0.174	Fluvial.....	Arkose.
LBIW-13	.24	2.29	.69	--	.011	Fluvial.....	Arkose.
LBIW-14	.36	1.85	1.06	--	.223	Fluvial.....	Arkose.
LBIW-15	.52	1.50	2.35	--	.375	Fluvial.....	Arkose.
LBIW-16	.24	2.41	1.11	--	.184	Fluvial.....	Arkose.
LBIW-17	.36	1.77	1.02	--	.166	Fluvial.....	Arkose.
LBIW-18	.31	2.11	1.30	--	.172	Fluvial.....	Arkose.
LV1-1	.27	2.22	.96	--	.146	Fluvial.....	Arkose.
LV1-2	.44	1.45	1.39	--	.198	Fluvial.....	Arkose.
LV1-3	.16	2.97	.83	.47	.099	Fluvial.....	Arkose.
LV1-4	.74	.92	3.17	--	.610	Fluvial.....	Lithic arkose.
LV1-5	.50	1.32	1.48	--	.295	Fluvial.....	Arkose.
LV1-6	.16	2.98	.74	--	.987	Fluvial.....	Arkose.
LV1-7	.71	1.16	8.30	--	.790	Fluvial.....	Arkose.
LV1-14	.35	1.87	1.11	--	.213	Fluvial.....	Arkose.
LV1-15	.29	2.10	.77	.47	.153	Fluvial.....	Arkose.
LV1-16	.63	.96	3.76	.51	.404	Fluvial.....	Arkose.
LV1-17	.10	3.53	.32	--	.048	Fluvial.....	Subarkose.
LV1-18	.19	2.74	1.09	--	.113	Fluvial.....	Arkose.
LV1-19	.15	2.97	.41	.66	.077	Eolian.....	Arkose.
LV1-20	.19	2.73	.65	.69	.113	Eolian.....	Arkose.
LV2-1	.42	1.84	1.14	--	.261	Fluvial.....	Lithic arkose.
LV2-2	.37	2.03	1.11	--	.247	Fluvial.....	Lithic arkose.
LV2-3	.43	1.67	1.09	--	.296	Fluvial.....	Lithic arkose.
LV2-4	.51	1.25	1.36	--	.237	Fluvial.....	Lithic arkose, feldspathic litharenite.
LV2-5	.24	2.61	.81	--	.187	Fluvial.....	Lithic arkose.
LV2-6	.34	1.83	3.79	--	.201	Fluvial.....	Lithic arkose.
LV2-7	.81	.67	2.09	.51	.522	Fluvial.....	Arkose.
LV2-8	.11	3.48	.54	--	.062	Marine.....	Subarkose.

Table 8.--Execution of program using data set two

```
table

Enter name of output file lime

Enter width of table 120

Enter name of input segment tabx

Enter number of columns 13

Enter number of header columns 3

Enter number of lines per page 45

Enter total number of columns to get leaders 0

Enter beginning and ending column numbers that encompass header        1 1 2

Enter beginning and ending column numbers that encompass header        2 3 12

Enter beginning and ending column numbers that encompass header        3 13 13

word (          1       ) is          8
word (          2       ) is          11
word (          3       ) is          13
word (          4       ) is          9
word (          5       ) is          9
word (          6       ) is          9
word (          7       ) is          9
word (          8       ) is          6
word (          9       ) is          7
word (          10      ) is          8
word (          11      ) is          10
word (          12      ) is          11
word (          13      ) is          18
wide is         128
minimum width for this table is       128
Enter new value for table width 130

col             1       =             9
col             2       =             12
col             3       =             13
col             4       =             9
col             5       =             9
col             6       =             9
col             7       =             9
col             8       =             6
col             9       =             7
col             10      =             8
col             11      =             10
col             12      =             11
col             13      =             18
Are those OK? yes
r 1452 0.973 18.384 421
```

Table 9.--Resultant table using data set two

Table 2.--Petrographic data for limestones

[* , present; --, not present]

Sample information		Composition										
Sample No.	Location in section(m above base)	Intraclasts	Oolites	Pellets	Fossils	Micrite	Spar	Chert	Quartz	Feldspar	Rock fragments	Robert Folk's scheme
BIW-11	130.5	*	*	--	*	*	*	--	*	*	--	Bio-oosparite.
BIW-14	173.	*	*	--	*	--	*	--	*	*	*	Biosparite.
BIW-15	173.5	*	--	--	*	--	*	--	*	*	--	Biosparite.
BIW-16	173.5	*	--	--	*	--	*	--	*	*	--	Biosparite.
BIW-17	167.5	*	--	*	*	*	--	--	*	--	--	Micrite.
BIW-21	248.5	*	--	*	*	*	--	--	*	*	--	Pelmicrite.
BIW-23	372.5	--	--	*	*	*	*	--	*	--	--	Micrite.
LV1-6	230.	*	--	*	--	*	*	*	*	--	--	Micrite.
LV1-7	251.	--	--	*	*	*	*	--	*	--	--	Biomicrite.
LV1-9	297.5	*	--	*	--	*	*	--	*	--	--	Dismicrite.
LV1-11	316.5	--	--	--	*	--	*	*	--	--	--	Chert replacing spar replacing fossils.
LV1-12	317.	--	--	--	--	--	*	*	--	--	--	Chert replacing spar.
LV1-13	321.5	--	*	*	*	*	*	--	--	--	--	Peldismicrite.
LV1-14	324.5	--	--	*	*	*	*	--	--	--	--	Pelbiomicrite.
LV2-6	220.5	*	--	--	*	*	*	--	*	--	--	Micrite.
LV2-8	260.5	*	--	*	*	*	*	--	*	*	--	Intramicrite.
LV2-11	344.	*	--	*	*	*	*	--	*	--	--	Pelintramicrite.

Table 10.--Execution of program using data set three

```
table

Enter name of output file elk1

Enter width of table 100

Enter name of input segment para

Enter number of columns 3 .

Enter number of header columns 0

Enter number of lines per page 45

Enter total number of columns to get leaders 1

Enter column number to get leaders 2

What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 0

word (              1       ) is              8
word (              2       ) is             15
word (              3       ) is             16
wide is            39
col                 1       =                10
col                 2       =                73
col                 3       =                17
Are those OK?   no

Enter new column widths
Width for col                   1       = 10

Width for col                   2       = 63

Width for col                   3       = 27
r 1325 1.399 44.114 759
```

Table 11.--Modified execution of program using data set three

```
table

Enter name of output file elk

Enter width of table 100

Enter name of input segment para

Enter number of columns 3

Enter number of header columns 0

Enter number of lines per page 48

Enter total number of columns to get leaders 1

Enter column number to get leaders 2

What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 1

word (              1       ) is              8
word (              2       ) is             15
word (              3       ) is             16
wide is            39
col                 1       =                10
col                 2       =                73
col                 3       =                17
Are those OK? yes
r 1316 0.947 25.894 365
```

Table 12.--Resultant table using data set three

Table 2.-Description of fossil localities in the Elk Ridge area

[Numbers refer to collections listed in tables 1 and 3]

Colln. No.	Stratigraphic position or lithology	Geographic location
	Paradox Member of Hermosa Formation	
1	Uppermost beds, bioclastic limestone	Gypsum Canyon, near mouth.
2	Upper part, fossils in sandy limestone	Do.
	Upper Member of Hermosa Formation	
3	Black cherty limestone within 100 ft of top	Mouth of Clearwater Canyon.
4	Light-gray bioclastic limestone, 5-9 ft above colln. 3	Do.
5	Black chert limestone (above colln. 3)	Do.
6	5-ft bed of dense light-gray limestone, many microfossils	Mouth of Dark Canyon.
7	Sandstone, brown to gray, 20-25 ft thick, above colln. 6	Mouth of Dark Canyon, about 50 ft above creek level.
8	Cherty limestone with abundant fossils, 200 ft below top	Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W.
9	Limestone within 30 ft of the top of the formation	Intersection of Trail and Dark Canyons. Lat 37°49'45'' N., long 109°55'23'' W.
10	Dark-gray limestone forming top ledge of the formation	Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W.
11	100 ft below colln. 10	Do.

Table 12.--Resultant table using data set three--Continued

Table 2.-Description of fossil localities in the Elk Ridge area--Continued

Colln. No.	Stratigraphic position or lithology	Geographic location
	Upper Member of Hermosa Formation--Continued	
12	Upper part	Upper part of Gypsum Canyon.
	Rico Formation	
13	About 300 ft below top (near base)	Gypsum Canyon. Lat 37°55'56" N., long 109°57'50" W.
14	Upper part	Hill on south side of Cross Canyon. Lat 38°00'35" N., long 109°55'26" W.
15	Limestone	Cross Canyon, 1½-2 mi north of intersection with Imperial Valley.
16	___do	Do.
17	100-500 ft below top	Cross Canyon.
18	Topmost bed	West end of Ruin Park.
19	Near top	In Peavine Canyon, about midway between its junctions with Kigalia and Dark Canyons.
20	___do	Do.

Table 13.--Modified resultant table using data set three

Table 2.-Description of fossil localities in the Elk Ridge area

[Numbers refer to collections listed in tables 1 and 3]

Colln. No.	Stratigraphic position or lithology	Geographic location
	Paradox Member of Hermosa Formation	
1	Uppermost beds, bioclastic limestone.....	Gypsum Canyon, near mouth.
2	Upper part, fossils in sandy limestone.....	Do.
	Upper Member of Hermosa Formation	
3	Black cherty limestone within 100 ft of top.....	Mouth of Clearwater Canyon.
4	Light-gray bioclastic limestone, 5-9 ft above colln. 3.....	Do.
5	Black chert limestone (above colln. 3).....	Do.
6	5-ft bed of dense light-gray limestone, many microfossils...	Mouth of Dark Canyon.
7	Sandstone, brown to gray, 20-25 ft thick, above colln. 6....	Mouth of Dark Canyon, about 60 ft above creek level.
8	Cherty limestone with abundant fossils. 200 ft below top....	Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W.
9	Limestone within 30 ft of the top of the formation.....	Intersection of Trail and Dark Canyons. Lat 37°49'45'' N., long 109°55'23'' W.
10	Dark-gray limestone forming top ledge of the formation.....	Cataract Canyon at west end of Imperial Valley. Lat 38°01'55'' N., long 110°03'04'' W.
11	100 ft below colln. 10.....	Do.
12	Upper part.....	Upper part of Gypsum Canyon.
	Rico Formation	
13	About 300 ft below top (near base).....	Gypsum Canyon. Lat 37°55'56'' N., long 109°57'50'' W.

Table 13.--Modified resultant table using data set three--Continued

Table 2.-Description of fossil localities in the Elk Ridge area—Continued

Colln. No.	Stratigraphic position or lithology	Geographic location
	Rico Formation—Continued	
14	Upper part	Hill on south side of Cross Canyon. Lat 38°00'35'' N., long 109°55'26'' W.
15	Limestone	Cross Canyon, 1½-2 mi north of intersection with Imperial Valley.
16	...do	Do.
17	100-500 ft below top	Cross Canyon.
18	Topmost bed	West end of Ruin Park.
19	Near top	In Peavine Canyon, about midway between its junctions with Kigalia and Dark Canyons.
20	...do	Do.

Table 14.--Execution of program using data set four

```
table
Enter name of output file sands1
Enter width of table 130
Enter name of input segment tp1a
Enter number of columns 11
Enter number of header columns 7
Enter number of lines per page 45
Enter total number of columns to get leaders 2
Enter column number to get leaders 1
Enter column number to get leaders 2
What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 0
Enter beginning and ending column numbers that encompass header          1 1 2
Enter beginning and ending column numbers that encompass header          2 3 4
Enter beginning and ending column numbers that encompass header          3 5 6
Enter beginning and ending column numbers that encompass header          4 7 8
Enter beginning and ending column numbers that encompass header          5 9 9
Enter beginning and ending column numbers that encompass header          6 10 10
Enter beginning and enning column numbers that encompass header          7 11 11

word (          1       ) is            16
word (          2       ) is            11
word (          3       ) is            10
word (          4       ) is             9
word (          5       ) is             8
word (          6       ) is             8
word (          7       ) is             8
word (          8       ) is             9
word (          9       ) is             7
word (         10       ) is             8
word (         11       ) is             7
wide is       101
col             1       =               18
col             2       =               29
col             3       =               11
col             4       =               10
col             5       =                9
col             6       =                9
col             7       =                9
col             8       =               10
col             9       =                8
col            10       =                9
col            11       =                8
Are those OK? yes
r 1351 2.085 45.586 874
```

Table 15.--Resultant table using data set four

Table 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base

[Localities are listed alphebetically by country, and specific area where known. The computer index number and McKee's index number are also listed. ND indicates no data.]

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SkI	K'G
Algeria	Ouargla	4018	26	1.935	0.2615	ND	ND	1.560	-0.471	0.404
Egypt	Ayn Amur	4130	767	-0.683	1.6055	ND	ND	0.179	0.702	0.668
Jordan	Ram Fort	594	685	1.403	0.3789	1.200	0.4353	1.177	0.318	0.489
Lybia	Fezzan	622	702	2.190	0.2192	2.200	0.2176	0.812	0.015	0.507
	...do	639	703	2.350	0.1961	2.490	0.1780	0.710	0.240	0.492
	...do	556	727	2.370	0.1934	2.350	0.1961	0.971	0.006	0.422
	...do	616	728	1.570	0.3368	1.290	0.4090	1.180	0.316	0.421
	...do	4106	729	1.754	0.2965	ND	ND	0.913	0.117	0.439
	...do	4107	729	1.578	0.3349	ND	ND	0.936	0.425	0.462
	...do	595	730	2.333	0.1989	3.000	0.1250	1.630	0.208	0.371
	...do	4109	730	2.279	0.2060	ND	ND	1.569	-0.758	0.334
	...do	597	731	2.213	0.2161	2.000	0.2500	1.456	0.169	0.360
	...do	4111	731	1.684	0.3112	ND	ND	1.392	0.541	0.361
	...do	561	732	2.003	0.2500	1.700	0.3078	1.420	0.261	0.366
	...do	573	733	1.839	0.2793	1.400	0.3789	1.253	0.435	0.382
	...do	585	734	1.450	0.3660	0.700	0.6156	1.348	0.860	0.441
	...do	4115	734	1.155	0.4491	ND	ND	1.170	0.725	0.731
	...do	523	735	2.886	0.1350	3.680	0.1560	1.305	0.641	0.486
	...do	4117	735	2.412	0.1879	ND	ND	1.408	-0.565	0.362
	...do	564	736	1.563	0.3392	1.400	0.3789	0.848	0.087	0.346
	...do	524	737	2.670	0.1571	3.400	0.0947	1.386	0.667	0.356
	...do	598	738	1.230	0.4263	0.350	0.7846	1.616	0.540	0.668
	...do	4121	739	2.468	0.1807	ND	ND	1.397	-0.748	0.363
	...do	4122	739	2.044	0.2425	ND	ND	1.472	-0.294	0.343
	...do	4123	740	1.677	0.3127	ND	ND	1.553	0.067	0.377
	...do	4124	743	0.961	0.5137	ND	ND	1.464	0.273	0.459
	...do	599	744	1.560	0.3392	0.800	0.5743	1.671	0.497	0.333
	...do	519	745	2.450	0.1830	2.850	0.1387	1.390	0.369	0.342

Table 15.--Resultant table using data set four--Continued

Table 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base--Continued

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SkI	K'G
Lybia	Fezzan	576	746	2.000	0.2500	1.900	0.2679	1.417	0.082	0.351
	Ghat	559	675	2.006	0.2483	2.620	0.1627	1.636	0.448	0.368
	Sabha	553	664	1.553	0.3415	1.550	0.3415	0.940	0.146	0.505
	...do	525	667	2.300	0.2031	2.950	0.1294	1.588	0.515	0.434
	Sedales	4061	672	1.769	0.2934	ND	ND	1.431	-0.080	0.462
	Three Mountains	596	680	2.236	0.2117	2.000	0.2500	1.164	0.246	0.391
	...do	526	681	2.210	0.2161	2.150	0.2253	1.113	0.068	0.370
New Mexico(U.S.A.)	White Sands National Monument	4137	828	2.047	0.2420	ND	ND	1.035	0.126	0.510
	...do	4191	0	2.702	0.1537	ND	ND	0.820	-0.113	0.460
Peru	Pampe Corre Viento, Lomitas	4031	247	-0.093	1.0666	ND	ND	0.694	0.701	0.378
	...do	4032	248	0.787	0.5795	ND	ND	1.411	0.251	0.387
Tunisia	Bir Sultan	1621	693	2.416	0.1882	2.350	0.1961	0.941	0.056	0.405

Table 16.--Modified execution of program using data set four

```
table

Enter name of output file sands2

Enter width of table 130

Enter name of input segment tp1a

Enter number of columns 11

Enter number of header columns 7

Enter number of lines per page 45

Enter total number of columns to get leaders 2

Enter column number to get leaders 1

Enter column number to get leaders 2

What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 0

Enter beginning and ending column numbers that encompass header          1 1 2

Enter beginning and ending column numbers that encompass header          2 3 4

Enter beginning and ending column numbers that encompass header          3 5 6

Enter beginning and ending column numbers that encompass header          4 7 8

Enter beginning and ending column numbers that encompass header          5 9 9

Enter beginning and ending column numbers that encompass header          6 10 10

Enter beginning and ending column numbers that encompass header          7 11 11

word (          1       ) is          16
word (          2       ) is          11
word (          3       ) is          10
word (          4       ) is          9
word (          5       ) is          8
word (          6       ) is          8
word (          7       ) is          8
word (          8       ) is          9
word (          9       ) is          7
word (          10      ) is          8
word (          11      ) is          7
wide is         101
col             1       =             18
col             2       =             29
col             3       =             11
col             4       =             10
col             5       =             9
col             6       =             9
col             7       =             9
col             8       =             10
col             9       =             8
col             10      =             9
col             11      =             8
Are those OK? no
```

Table 16.--Modified execution of program using data set four--Continued

```
Enter new column widths
Width for col  1  = 20
Width for col  2  = 27
Width for col  3  = 11
Width for col  4  = 10
Width for col  5  = 9
Width for col  6  = 9
Width for col  7  = 9
Width for col  8  = 10
Width for col  9  = 8
Width for col  10 = 9
Width for col  11 = 8
r 1433 2.264 61.672 1225 level 2, 13
```

Table 17.--Modified resultant table using data set four

Table 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base

[Localities are listed alphabetically by country, and specific area where known. The computer index number and McKee's index number are also listed. ND indicates no data.]

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SKI	K'G
Algeria	Ouargla	4018	26	1.935	0.2615	ND	ND	1.560	-0.471	0.404
Egypt	Ayn Amur	4130	767	-0.683	1.6055	ND	ND	0.179	0.702	0.668
Jordan	Ram Fort	594	685	1.403	0.3789	1.200	0.4353	1.177	0.318	0.489
Lybia	Fezzan	622	702	2.190	0.2192	2.200	0.2176	0.812	0.015	0.507
	do	639	703	2.350	0.1961	2.490	0.1780	0.710	0.240	0.492
	do	556	727	2.370	0.1934	2.350	0.1961	0.971	0.006	0.422
	do	616	728	1.570	0.3368	1.290	0.4090	1.180	0.316	0.421
	do	4106	729	1.754	0.2965	ND	ND	0.913	0.117	0.439
	do	4107	729	1.578	0.3349	ND	ND	0.936	0.425	0.462
	do	595	730	2.333	0.1989	3.000	0.1250	1.630	0.208	0.371
	do	4109	730	2.279	0.2060	ND	ND	1.569	-0.758	0.334
	do	597	731	2.213	0.2161	2.000	0.2500	1.456	0.169	0.360
	do	4111	731	1.684	0.3112	ND	ND	1.392	0.541	0.361
	do	561	732	2.003	0.2500	1.700	0.3078	1.420	0.261	0.366
	do	573	733	1.839	0.2793	1.400	0.3789	1.253	0.435	0.382
	do	585	734	1.450	0.3660	0.700	0.6156	1.348	0.860	0.441
	do	4115	734	1.155	0.4491	ND	ND	1.170	0.725	0.731
	do	523	735	2.886	0.1350	3.680	0.1560	1.305	0.641	0.486
	do	4117	735	2.412	0.1879	ND	ND	1.408	-0.565	0.362
	do	564	736	1.563	0.3392	1.400	0.3789	0.848	0.087	0.346
	do	524	737	2.670	0.1571	3.400	0.0947	1.386	0.667	0.356
	do	598	738	1.230	0.4263	0.350	0.7846	1.616	0.540	0.668
	do	4121	739	2.468	0.1807	ND	ND	1.397	-0.748	0.363
	do	4122	739	2.044	0.2425	ND	ND	1.472	-0.294	0.343
	do	4123	740	1.677	0.3127	ND	ND	1.553	0.067	0.377
	do	4124	743	0.961	0.5137	ND	ND	1.464	0.273	0.459
	do	599	744	1.560	0.3392	0.800	0.5743	1.671	0.497	0.333
	do	519	745	2.450	0.1830	2.850	0.1387	1.390	0.369	0.342

Table 17.--Modified table using data set four--Continued

Table 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base--Continued

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SkI	K'G
Lybia	Fezzan	576	746	2.000	0.2500	1.900	0.2679	1.417	0.082	0.351
	Ghat	559	675	2.006	0.2483	2.620	0.1627	1.636	0.448	0.368
	Sabha	553	664	1.553	0.3415	1.550	0.3415	0.940	0.146	0.505
	...do	525	667	2.300	0.2031	2.950	0.1294	1.588	0.515	0.434
	Sedales	4061	672	1.769	0.2934	ND	ND	1.431	-0.080	0.462
	Three Mountains	596	680	2.236	0.2117	2.000	0.2500	1.164	0.246	0.391
	...do	526	681	2.210	0.2161	2.150	0.2253	1.113	0.068	0.370
New Mexico(U.S.A.)	White Sands National Monument.	4137	828	2.047	0.2420	ND	ND	1.035	0.126	0.510
	...do	4191	0	2.702	0.1537	ND	ND	0.820	-0.113	0.460
Peru	Pampe Corre Viento, Lomitas	4031	247	-0.093	1.0666	ND	ND	0.694	0.701	0.378
	...do	4032	248	0.787	0.5795	ND	ND	1.411	0.251	0.387
Tunisia	Bir Sultan	1621	693	2.416	0.1882	2.350	0.1961	0.941	0.056	0.405

Table 18.--Execution of program using data set five

```
table
Enter name of output file sands3
Enter width of table 130
Enter name of input segment tp1b
Enter number of columns 11
Enter number of header columns 7
Enter number of lines per page 45
Enter total number of columns to get leaders 2
Enter column number to get leaders 1
Enter column number to get leaders 2
What type of leader do you want? Enter 0 for dot. Enter 1 for underscore. 0
Enter beginning and ending column numbers that encompass header          1 1 2
Enter beginning and ending column numbers that encompass header          2 3 4
Enter beginning and ending column numbers that encompass header          3 5 6
Enter beginning and ending column numbers that encompass header          4 7 8
Enter beginning and ending column numbers that encompass header          5 9 9
Enter beginning and ending column numbers that encompass header          6 10 10
Enter beginning and ending column numbers that encompass header          7 11 11

word (          1       ) is          16
word (          2       ) is          11
word (          3       ) is          10
word (          4       ) is           9
word (          5       ) is           8
word (          6       ) is           8
word (          7       ) is           8
word (          8       ) is           9
word (          9       ) is           9
word (         10       ) is          10
word (         11       ) is          10
wide is       108
col             1       =             18
col             2       =             22
col             3       =             11
col             4       =             10
col             5       =              9
col             6       =              9
col             7       =              9
col             8       =             10
col             9       =             10
col            10       =             11
col            11       =             11
Are those OK? no
```

Table 18.--Execution of program using data set five--Continued

```
Enter new column widths
Width for col  1  = 20
Width for col  2  = 23
Width for col  3  = 11
Width for col  4  = 10
Width for col  5  = 9
Width for col  6  = 9
Width for col  7  = 9
Width for col  8  = 10
Width for col  9  = 9
Width for col  10 = 10
Width for col  11 = 10
r 1442 2.367 57.242 1200
```

Table 19.--Resultant table using data set five

Table 7.--Graphical measures of the sand samples from interdune and serir sands used as a data base

[Localities are listed alphabetically by country, and specific area where known. The computer index number and McKee's index number are also listed. ND indicates no data.]

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SkI	K'G
Algeria	Ouargla	4018	26	1.935	0.2615	ND	ND	1.560	-0.471	0.404
Egypt	Ayn Amur	4130	767	-0.683	1.6055	NO	ND	0.179	0.702	0.668
Jordan	Ram Fort	594	685	1.403	0.3789	1.200	0.4353	1.177	0.318	0.489
Lybia	Fezzan	622	702	2.190	0.2192	2.200	0.2176	0.812	0.015	0.507
	...do	639	703	2.350	0.1961	2.490	0.1780	0.710	0.240	0.492
	...do	556	727	2.370	0.1934	2.350	0.1961	0.971	0.006	0.422
	...do	616	728	1.570	0.3368	1.290	0.4090	1.180	0.316	0.421
	...do	4106	729	1.754	0.2965	ND	ND	0.913	0.117	0.439
	...do	4107	729	1.578	0.3349	ND	ND	0.936	0.425	0.462
	...do	595	730	2.333	0.1989	3.000	0.1250	1.630	0.208	0.371
	...do	4109	730	2.279	0.2060	ND	ND	1.569	-0.758	0.334
	...do	597	731	2.213	0.2161	2.000	0.2500	1.456	0.169	0.360
	...do	4111	731	1.684	0.3112	ND	ND	1.392	0.541	0.361
	...do	561	732	2.003	0.2500	1.700	0.3078	1.420	0.261	0.366
	...do	573	733	1.839	0.2793	1.400	0.3789	1.253	0.435	0.382
	...do	585	734	1.450	0.3660	0.700	0.6156	1.348	0.860	0.441
	...do	4115	734	1.155	0.4491	ND	ND	1.170	0.725	0.731
	...do	523	735	2.886	0.1350	3.680	0.1560	1.305	0.641	0.486
	...do	4117	735	2.412	0.1879	ND	ND	1.408	-0.565	0.362
	...do	564	736	1.563	0.3392	1.400	0.3789	0.848	0.087	0.346
	...do	524	737	2.670	0.1571	3.400	0.0947	1.386	0.667	0.356
	...do	598	738	1.230	0.4263	0.350	0.7846	1.616	0.540	0.668
	...do	4121	739	2.468	0.1807	ND	ND	1.397	-0.748	0.363
	...do	4122	739	2.044	0.2425	ND	ND	1.472	-0.294	0.343
	...do	4123	740	1.677	0.3127	ND	ND	1.553	0.067	0.377
	...do	4124	743	0.961	0.5137	ND	ND	1.464	0.273	0.459
	...do	599	744	1.560	0.3392	0.800	0.5743	1.671	0.497	0.333
	...do	519	745	2.450	0.1830	2.850	0.1387	1.390	0.369	0.342
	...do	576	746	2.000	0.2500	1.900	0.2679	1.417	0.082	0.351

Table 19.--Resultant table using data set five--Continued

Table 7.--Graphical measures of the sand samples from interdune and seric sands used as a data base--Continued

Locality				Mean grain size		Median grain size		Sorting	Skewness	Kurtosis
Principal country or area	Specific area	Computer Index No.	McKee's Index No.	phi	mm	phi	mm	rho I	SkI	K'G
Lybia.....	Ghat.....	559	675	2.006	0.2483	2.620	0.1627	1.636	0.448	0.368
	Sabha.....	553	664	1.553	0.3415	1.550	0.3415	0.940	0.146	0.505
	...do.....	525	667	2.300	0.2031	2.950	0.1294	1.588	0.515	0.434
	Sedales.....	4061	672	1.769	0.2934	ND	ND	1.431	-0.080	0.462
	Three Mountains.....	596	680	2.236	0.2117	2.000	0.2500	1.164	0.246	0.391
	...do.....	526	681	2.210	0.2161	2.150	0.2253	1.113	0.068	0.370
New Mexico(U.S.A.)	White Sands National Monument.	4137	828	2.047	0.2420	ND	ND	1.035	0.126	0.510
	...do.....	4191	0	2.702	0.1537	ND	ND	0.820	-0.113	0.460
Peru.....	Pampe Corre Viento, Lomitas.	4031	247	-0.093	1.0666	ND	ND	0.694	0.701	0.378
	...do.....	4032	248	0.787	0.5795	ND	ND	1.411	0.251	0.387
Tunisia.....	Bir Sultan.....	1621	693	2.416	0.1882	2.350	0.1961	0.941	0.056	0.405

Appendix C: Listing of program

```
table: proc;
dcl  close_file entry (char (*));
dcl  get_wdir_ entry returns (char (168));
dcl  com_err_ entry options (variable);
dcl  term_$seg_ptr entry (ptr, fixed bin (35));
dcl  store char (255) varying;
dcl  sel (10, 3) fixed bin (24) init ((10) ((3)0));
dcl (wholeline, nextline, line) char (1000) varying;
dcl (LAB1, LAB2, LAB3, LAB4) label;
dcl (colpart, colval) char (255) varying;
dcl (spaces, liny, title) char (255) varying;
dcl  x (25) fixed bin (24) init ((25)0);
dcl (holdit1 (25), tiarray (25)) char (255) varying;
dcl  tlin fixed bin (24) init (1);
dcl (rp, hl, i, nextchar, colwid) fixed bin (24)  init  (0);
dcl (widthcol, no, lin_len) fixed bin (24) init (0);
dcl  ans char (3);
dcl (del, r, sym) char (1);
dcl  page fixed bin (24) init (1);
dcl  (ld, nx1, avg1, j, hd, pline, wide) fixed bin (24) init
(0);
dcl (maxchar, maxword, num, t, no1) fixed bin (24) init (0);
dcl (avg (50), nx (50)) fixed bin (24) init ((50)0);
dcl (dataw (50), datas (50), data (50)) fixed bin (24)  init
((50)0);
dcl (wordx (50), word (50)) fixed bin (24) init ((50)0);
dcl (headar (50), char (50)) fixed bin (24) init ((50)0);
dcl  (lnerror  (50),  cherror  (50))  fixed  bin  (24)  init
((50)0);
dcl  sysin file;
dcl (dd, chx, ww, erno, oo, ch, col, w, n, lin, ncol, o,  p)
fixed bin (24);
dcl (pcol, scol, nlines, paper_width, ndots) fixed bin (24);
dcl (search, substr, before, divide) builtin;
dcl  hcs_$initiate_count entry (char (*), char (*),
     char (*), fixed bin (24), fixed bin (2), ptr, fixed bin
(35));
dcl  bc fixed bin (24);
dcl  sysprint file;
dcl  code fixed bin (35);
dcl  dname char (168);
dcl (ename, oname) char (32);
dcl  nl char (1) aligned int static init ("
");
dcl  print char (500)varying;
dcl  input_ptr ptr;
dcl  input_chars char (input_len) based (input_ptr);
dcl  input_len fixed bin (24);
dcl  zz file;
          del = "/";
          dname = get_wdir_ ();
```

```
          put skip list ("Enter name of output file");
          get list (oname);
          put skip list ("Enter width of table");
          get list (paper_width);
          put skip list ("Enter name of input segment");
          get list (ename);
          put skip list ("Enter number of columns");
          get list (ncol);
          put  skip list ("Enter number of header columns");
          get list (scol);
          put skip list ("Enter number of lines per  page");
          get list (nlines);
          put  skip  list ("Enter total number of columns to
get leaders");
          get list (ndots);
          if ndots = 0 then go to BY;
          do i = 1 to ndots;
               put skip list ("Enter column  number  to  get
leaders");
               get list (dd);
               x (i) = dd;
          end;
          put  skip  list ("What type of leader do you want?
Enter 0 for dot. Enter 1 for underscore.");
          get list (ld);
          if ld = 0 then sym = ".";
          else sym = "_";
BY:       open file (zz)  title  ("vfile_ "||oname)  stream
output print linesize (158) pagesize (nlines);
          put file (zz) page;
          call hcs_$initiate_count (dname, ename, "", bc, 2,
input_ptr, code);
          if input_ptr = null () then do;
               call com_err_ (code, "table");
               go to endit;
          end;
          input_len = divide (bc, 9, 21, 0);
          LAB1 = subhed1;
          LAB2 = hed1;
          LAB3 = addcol;
          LAB4 = whichone;
          erno = 0;
          lin = 0;
          n = 0;
start1:   col = 0;
          pcol = 0;
start:    if n+1 >= input_len then go to out;
          n = n+1;
          lin = lin+1;
          w = 0;
          ch = 0;
          ww = 0;
          chx = 0;
```

```
/* only lines beg with S,H,L are used to compute table width
*/
          if substr (input_chars, n, 1) = "S" then do;
               n = n+1;
               go to countline1;
          end;
          if substr (input_chars, n, 1) = "L" |substr
(input_chars, n, 1) = "H" then do;
               r = substr (input_chars, n, 1);
               n = n+1;
               go to countline;
          end;
          else;
findend:  if substr (input_chars, n, 1) = nl then go to
start;
          n = n+1;
          go to findend;
/* if space found go to wordcount, if delimeter found go to
*/
/* colcount, otherwise add up no. of characters in the word
*/
countline: if substr (input_chars, n, 1) = " " then go to
wordcount;
          else if substr (input_chars, n, 1) = del then go
to colcount;
          else n = n+1;
          w = w+1;
          ch = ch+1;
          go to countline;
/*  if line beg with L program checks for period in the word
*/
/* if . present the word is broken into 2 parts; oo is the
value of */
/*  the no. of places to the rt of the decimal plus the
decimal */
/* i is the no. of places to left of the decimal */
/* array dataw containslargest value of oo, array datas
contains */
/* largest value of i -- for each column */
wordcount: if r = "L" then do;
               store = substr (input_chars, n-w, w);
               if index (store, ".") = 0 then go to LA1;
               p = index (store, ".");
               oo = w-p+1;
               i = w-oo;
               if dataw (col+1)<oo then dataw (col+1) = oo;
               if datas (col+1)<i then datas (col+1) = i;
               go to LA2;
LA1:           if data (col+1)<w then data (col+1) = w;
/* array data contains longest nondecimal word in each
column */
LA2:
          end;
```

```
          if word (col+1)<w then do;
               word (col+1) = w;
/* stores largest word for each column */
               n = n+1;
               ch = ch+1;
               w = 0;
          end;
          else do;
               w = 0;
               n = n+1;
               ch = ch+1;
          end;
          go to countline;
/* store largest no. of characters in each column &
increments */
/* number of columns -- for line that begins with H or L */
colcount: if char (col+1) < ch then do;
               char (col+1) = ch;
               n = n+1;
               w = 0;
               ch = 0;
               col = col+1;
          end;
          else do;
               n = n+1;
               ch = 0;
               w = 0;
               col = col+1;
          end;
          if substr (input_chars, n, 1) = nl then go to
erfind;
          else go to countline;
erfind:   if col = ncol then go to start1;
          else erno = erno+1;
          lnerror (erno) = lin;
          cherror (erno) = col;
          go to start1;
/* if space found go to wordcount1, if delimeter found go
to */
/* colcount1, otherwise add up no. of characters in the word
*/
countline1: if substr (input_chars, n, 1) = " " then go to
wordcount1;
          else if substr (input_chars, n, 1) = del then go
to colcount1;
          else n = n+1;
          ww = ww+1;
          chx = chx+1;
          go to countline1;
/* this loop stores the largest word for each column for a
data line */
/* that begins with S */
wordcount1: if wordx (pcol +1) <ww then do;
```

```
                wordx (pcol+1) = ww;
                n = n+1;
                chx = chx +1;
                ww = 0;
           end;
           else do;
                ww = 0;
                n = n+1;
                chx = chx+1;
           end;
           go to countline1;
/* this loop increments no of columns by adding  occurrences
of delimeter */
colcount1: n = n+1;
           chx = 0;
           ww = 0;
           pcol = pcol+1;
           if  substr  (input_chars,  n,  1)  = nl then go to
erfind1;
           else go to countline1;
erfind1:   if pcol = scol then go to start1;
           else erno = erno +1;
           lnerror (erno) = lin;
           cherror (erno) = pcol;
           go to start1;
out:       if erno = 0 then go to out1;
           else do n = 1 to erno;
                put skip list ("line ", lnerror (n), "has  ",
cherror (n), "delimiters");
           end;
           go to endit;
out1:      do i = 1 to ncol;
                if  datas  (i)<data (i) then datas (i) = data
(i);
/* computes longest word possible for each data  column  and
stores in array word */
                p = datas (i)+dataw (i);
                if p ^= 0 then do;
                     if word (i)<p then word (i) = p;
                     data (i) = p;
                end;
           end;
           if scol = 0 then go to outx;
           do i = 1 to scol;
                put  skip  list  ("Enter beginning and ending
column numbers that encompass header ", i);
                get list (oo, p);
                sel (i, i) = oo;
                sel (i, i+1) = p;
/* adjusts width of minor header column so that the  sum  of
the widths */
/*  of  the  minor columns is not less than the width of the
major header */
```

```
/* column -- stores in array word */
/* value in array word */
               if oo = p then do;
                    if wordx (i)> word (oo) then word (oo) =
wordx (i);
                    go to incol;
               end;
               do j = oo to p;
                    hd = word (j)+hd;
               end;
               if wordx (i)> hd then do;
                    nx1 = wordx (i)-hd;
                    hd = wordx (i);
                    j = p-oo+1;
                    if nx1-j <= 0 then do;
LB1:                     word (oo) = word (oo)+1;
                         nx1 = nx1-1;
                         if nx1>0 then do;
                              oo = oo+1;
                              go to LB1;
                         end;
                         else go to incol;
                    end;
                    else;
LB2:                do j = oo to p;
                         word (j) = word (j)+1;
                         nx1 = nx1-1;
                    end;
                    if nx1>j then go to LB2;
                    else do;
LB3:                     word (oo) = word (oo)+1;
                         nx1 = nx1-1;
                         if nx1>0 then do;
                              oo = oo+1;
                              go to LB3;
                         end;
                         else;
                    end;
               end;
incol:         hd = 0;
          end;
outx:     do n = 1 to ncol;
/* increments each column width by 2 */
               wide = 2+word (n)+wide;
               word (n) = word (n)+2;
               put skip list ("word  (", n, ")  is", word
(n));
          end;
          put skip list ("wide is", wide);
          if wide > paper_width then do;
               put  skip list ("minimum width for this table
is  ", wide);
               put skip list ("Enter  new  value  for  table
```

```
width");
                get list (paper_width);
           end;
           if wide>paper_width then do;
                put skip list ("Minimum width for this  table
is", wide);
                go to endit;
           end;
           if wide = paper_width then go to next1;
fitit:     no = paper_width - wide;
           if no < ncol then go to fit2;
           do n = 1 to ncol;
                word (n) = word (n)+1;
           end;
           wide = wide + ncol;
           no = no- ncol;
           if no = 0 then go to next1;
           go to fit2;
answer:    get list (ans);
           if ans = "no" then go to newcol;
           if ans = "yes" then go to next1;
answer1:   if ans ^= "yes" | ans ^= "no" then do;
                put  skip  list  ("Please  answer  'yes'  or
'no'");
                go to answer;
           end;
newcol:    put skip list ("Enter new column widths");
           do n = 1 to ncol;
                put skip list ("Width for col", n, "=");
                get list (no);
                word (n) = no;
                maxword = maxword+no;
           end;
           if maxword > paper_width then do;
                put skip list  ("those  are  wider  than  the
table -- try again");
                maxword = 0;
                go to newcol;
           end;
           if maxword < paper_width then do;
                put  skip  list  ("Those  are  less  than the
table_width--try again");
                maxword = 0;
                go to newcol;
           end;
           else call valus (LAB4);
/* finds data or minor header column  with  largest  no.  of
characters */
fit2:      do n = 1 repeat n+1 while (n<ncol);
                if  maxchar  <  char  (n) then maxchar = char
(n);
                else;
           end;
```

```
          do n = 1 to ncol;
               if char (n) > word (n) then word (n)  =  word
(n)+1;
               wide = wide +1;
               if wide = paper_width then go to next1;
               if char (n) = maxchar then do;
                    no = paper_width -wide;
                    word (n) = word (n) + no;
                    go to next1;
               end;
               else;
          end;
next1:    do n = 1 to ncol;
               put skip list ("col", n, "=  ", word (n));
          end;
          put skip list ("Are those OK?");
          get list (ans);
          if ans = "yes" then call valus (LAB4);
          if ans = "no" then go to newcol;
          else go to answer1;
whichone: if num >= input_len then go to endit;
          if  hl+  tlin +pline >= nlines then go to newpage;
          lin_len = index (input_chars, nl);
          num = num+lin_len;
          line = before (input_chars, nl);
          input_chars = after (input_chars, nl);
          nextline = "";
          if substr (line, 1, 1) = "Q" then go to inc;
          if substr (line, 1, 1) = "I" then go to intit;
          if  substr  (line,  1,  1)  =  "W"  then  go  to
insertline;
          if substr (line, 1, 1) = "B" then go to blankline;
          if substr (line, 1, 1) = "X" then go to title1;
          if substr (line, 1, 1) = "T" then go to title1;
          if substr (line, 1, 1) = "A" then go to ntitle;
          if  substr  (line, 1, 1) = "H" then go to hednote;
          if substr (line, 1, 1) = "U" then go to under;
          if substr (line, 1, 1) = "D" then go to dash;
          if substr (line, 1, 1) = "S" then go  to  subhead;
          if substr (line, 1, 1) = "L" then go to line1;
          if  substr  (line,  1,  1)  =  "F" then call ftnot
(LAB4);
          else do;
               spaces = substr (line, 1, 10);
               put skip list ("line starts  with  the  wrong
character--", spaces);
               go to endit;
          end;
          go to whichone;
title1:   if substr (line, 1, 1) = "T" then t = 1;
          else t = 0;
          hd = 0;
          title = substr (line, 2, lin_len-2);
```

```
          no = length (title);
          if t = 1 & no+11 < paper_width then go to  center;
          else if t = 0 & no <paper_width then go to center;
          else line = substr (line, 2, lin_len-2);
          lin_len = length (line);
          no = 1;
title2:   no1 = paper_width;
          if t = 1 & lin_len < no1 then no1 = no1-11;
          if no-1+no1 >= lin_len then do;
               title = substr (line, no, lin_len-no+1);
               hd = 0;
               no1 = 0;
               go to center;
          end;
          else no1 = no+no1;
          call loop (no1, line);
          title = substr (line, no, no1-no);
          no = length (title) + no+1;
          hd = 1;
center:     spaces   =   copy   (" ",   ((paper_width-length
(title))/2));
          liny = spaces || title;
          if rp = 1 then go to inx;
          call put (liny);
          if t = 0 then hl = hl+1;
          if pline>0 then pline = pline+1;
          if t = 1 then do;
               if hd = 0 & length (title) < (paper_width-11)
then go to contin;
          end;
inx:      if t = 1 then do;
               tiarray (tlin) = liny;
               tlin = tlin+1;
               if rp = 1 then go to contin;
          end;
          if no1 = 0 then go to whichone;
          go to title2;
contin:    spaces  =  copy   (" ",   (paper_width-   (length
(title)+11))/2);
          liny = spaces||title||"--Continued";
          if rp = 1 then go to indo;
          if t = 1 then do;
               tiarray (tlin) = liny;
               call line8 (LAB4, tlin);
          end;
          else go to whichone;
hednote:  wholeline = "";
          line = substr (line, 2, lin_len-2);
          lin_len = length (line);
          n = 1;
hed1:     if n> ncol then call hed2 (LAB2);
          colval = before (line, del);
          colwid = length (colval);
```

```
          line = after (line, del);
          if colwid > word (n) then call lineb (LAB2, word
(n));
          else call linea (word (n));
          n = n+1;
          go to hed1;
insertline: liny = copy (" ", paper_width);
          call put (liny);
          call line8 (LAB4, pline);
blankline: liny = copy (" ", paper_width);
          call put (liny);
          call put (liny);
          tiarray (tlin) = liny;
          tlin = tlin+1;
          tiarray (tlin) = liny;
          call line8 (LAB4, tlin);
under:    wholeline = "";
          do i = 1 to scol;
               if sel (i, 3) = 0 then do;
                    spaces = copy (" ", 1);
                    wholeline = wholeline ||copy ("_",
headar (i)-1)||spaces;
               end;
               else if sel (i, 3) = 1 then wholeline =
wholeline||copy (" ", headar (i));
          end;
          call put (wholeline);
          tiarray (tlin) = wholeline;
          go to whichone;
ntitle:   tlin = tlin - 1;
          go to whichone;
intit:    if nlines- (hl+tlin+pline) >4 then go to whichone;
          rp = 1;
          oo = 0;
indo:     lin_len = index (input_chars, nl);
          num = num+lin_len;
          line = before (input_chars, nl);
          input_chars = after (input_chars, nl);
          oo = oo+1;
          if oo = 3 then do;
               tlin = tlin-1;
               go to title1;
          end;
          if oo = 4 then go to newpage;
          go to indo;
dash:     liny = copy ("_", paper_width);
          call put (liny);
          if pline>0 then pline = pline+1;
          tiarray (tlin) = liny;
          go to whichone;
inc:      call line8 (LAB4, tlin);
subhead:  wholeline = "";
          line = substr (line, 2, lin_len -2);
```

```
          lin_len = length (line);
          do i = 1 to scol;
               if sel (i, i) ^= sel (i, i+1) then do j = sel
(i, i) to sel (i, i+1);
                    headar (i) = word (j)+headar (i);
               end;
               else headar (i) = word (sel (i, i));
          end;
          n = 1;
          j = 0;
subhed1:  if n>scol then call hed2 (LAB1);
          else colval = before (line, del);
          colwid = length (colval);
          if j = 0 then do;
               if colwid = 1 then sel (n, 3) = 1;
               else sel (n, 3) = 0;
          end;
          line = after (line, del);
          if colwid>headar (n) then call lineb (LAB1, headar
(n));
          else call linea (headar (n));
          n = n+1;
          go to subhed1;
line1:    line = after (line, "L");
          if substr (line, 1, 1) ^= " " then holdit1  (1)  =
before (line, del)||"/";
          colval = after (line, del);
          do n = 2 to ncol;
               if  substr  (colval,  1,  3) = "do " | substr
(colval, 1, 4) = "Do. " then holdit1 (n) = holdit1 (n);
               else holdit1 (n) = before (colval, del)||"/";
               colval = after (colval, del);
          end;
          if pline = 0 then do;
               colval = "";
               do n = 1 to ncol;
                    colval = colval|| holdit1 (n);
               end;
               line = colval;
          end;
          j = 0;
line4:    liny = line;
          t = 0;
          nextchar = 0;
          n = 1;
          oo = 0;
          do i = 1 to ncol;
               colwid = length (before (liny, del))-1;
               if colwid > word (i) then do;
                    nextchar = 1;
                    colval = substr (liny, 1, colwid);
rd:                 widthcol = word (i);
                    colpart = substr (colval, 1, word  (i));
```

```
                    call loop (widthcol, colpart);
                    t = t+1;
                    colval    =    " "||substr     (colval,
widthcol+1, length (colval)-widthcol);
                    colwid = length (colval);
                    if colwid>word (i) then go to rd;
               end;
               if oo<t then oo = t;
               t = 0;
               liny = after (liny, del);
          end;
line6:    pline = pline+1;
          if  oo+hl+pline  +tlin >nlines then go to newpage;
          else;
line2:    if n = 1 then wholeline = "";
line3:    colval = before (line, del);
          colwid = length (colval);
          if j = 0 then do;
               colval = substr (colval, 1, colwid-1);
               colwid = colwid - 1;
          end;
          if n ^= ncol then line = after (line, del);
          if index (collate (), substr (colval, 1,  1))  >57
|index (collate (), substr (colval, 1, 1))-1 = 40 then go to
alpha;
          if substr (colval, 1, 1) ^= " " then go to number;
          else spaces = copy (" ", word (n));
          wholeline = wholeline || spaces;
          nextline = nextline || " " || del;
addcol:   n = n+1;
          if n > ncol then go to printit;
          go to line3;
number:   if nextchar = 1 | j>0 then go to alpha;
          if  n  =  1  |index  (colval, " ") ^= 0 then go to
alpha;
          if index (colval, ".") = 0 then go to intg;
          call linew (LAB3);
intg:     if index (collate (), substr  (colval,  1,  1))>47
then call linec (LAB3);
          else call linea (word (n));
          go to addcol;
alpha:    if j>0 then do;
               colval = "  "||colval;
               colwid = colwid+2;
               if colwid>word (n) then nextchar = 1;
          end;
          if colwid > word (n) then go to fititin;
          if word (n) > colwid+3 then do i = 1 to ndots;
               if n = x (i) & j = 0 then do;
                    if substr (colval, 1, 2) = "do" then do;
                         colval = copy (sym, 3)||colval;
                         colwid = colwid+3;
                    end;
```

```
widthcol) || spaces;
          if substr (colval, word (n)+1, 1) = " " then
nextline = nextline||substr (colval, widthcol+2, length
(colval)- (widthcol+1))||del;
          else nextline = nextline||substr (colval,
widthcol+1, length (colval)-widthcol)||del;
          go to addcol;
newpage:  if oo>0 then do i = 1 to nlines-
(pline+tlin+hl-1);
               wholeline = copy (" ", paper_width);
               call put (wholeline);
          end;
          hl = 0;
          oo = 0;
          do n = 1 to tlin;
               call put (tiarray (n));
          end;
          if rp = 1 then do;
               rp = 0;
               tiarray (tlin-1) = liny;
          end;
          pline = 0;
          if nextchar = 1 then do;
               colval = "";
               do n = 1 to ncol;
                    colval = colval|| holdit1 (n);
               end;
               n = 1;
               line = colval;
               go to line6;
          end;
          go to whichone;
printit:  call put (wholeline);
          if nextchar = 1 then do;
               nextchar = 0;
               j = j+1;
               line = nextline;
               nextline = "";
               go to line4;
          end;
          else go to whichone;
endit:    call close_file ("zz");
          close file (zz);
          call term_$seg_ptr (input_ptr, code);
          if code ^= 0 then do;
               call com_err_ (code, "table", "Unable to
terminate input_ptr");
          end;
linea:    proc (wd);
dcl  wd fixed bin (24);
               spaces = copy (" ", (wd-colwid)/2);
               if length (spaces) *2+colwid = wd then
wholeline = wholeline||spaces||colval||spaces;
```

```
                    spaces = copy (sym, word (n)-colwid-3);
                    wholeline = wholeline || colval ||
spaces ||"   ";
                    nextline = nextline || " " || del;
                    go to addcol;
               end;
               else if n = x (i) & j>0 then do;
                    colval = colval||".";
                    colwid = colwid +1;
                    call line9 (LAB3);
               end;
          end;
          if index (collate (), substr (colval, 1, 1))-1 =
32| index (collate (), substr (colval, 1, 1))-1 = 40 then
call line9 (LAB3);
          if index (collate (), substr (colval, 1, 1))-1 <58
& index (colval, " ") ^= 0 then call line9 (LAB3);
          if substr (colval, 1, 3) = "Do."| substr (colval,
1, 3) = "do " then do;
               if word (n)<9 then call linea (word (n));
               else do;
                    if substr (colval, 1, 3) = "do " then
spaces = copy (" ", word (n)-5);
                    else spaces = copy (" ", word (n)-6);
                    wholeline = wholeline||"  "||
colval||spaces;
                    nextline = nextline||" "||del;
               end;
               go to addcol;
          end;
          if substr (colval, 1, 2) = "NA" then do;
               call linea (word (n));
               go to addcol;
          end;
          else if substr (colval, 1, 2) = "ND" | index
(collate (), substr (colval, 1, 1))-1 < 58 then do;
               if index (colval, ".") = 0 then go to intg1;
               call linew (LAB3);
intg1:          if index (collate (), substr (colval, 1,
1))>47 &index (collate (), substr (colval, 1, 1))-1<58 then
call linec (LAB3);
               else call linea (word (n));
               go to addcol;
          end;
          else;
          call line9 (LAB3);
fititin: colpart = substr (colval, 1, word (n));
          widthcol = word (n);
          if substr (colval, word (n)+1, 1) = " " then go to
fititin1;
          call loop (widthcol, colpart);
fititin1: spaces = copy (" ", word (n)-widthcol);
          wholeline = wholeline || substr (colpart, 1,
```

```
                else wholeline = wholeline||" "||spaces||
colval||spaces;
                nextline = nextline||" "||del;
                return;
           end;
lineb:     proc (lab, wd);
dcl  lab label;
dcl  wd fixed bin (24);
dcl  hd1 char (255) varying;
                nextchar = 1;
                colwid = wd;
                call loop (colwid, colval);
                colwid = colwid-1;
                hd1 = substr (colval, 1, colwid);
                spaces = copy (" ", (wd-colwid)/2);
                if   length  (spaces)  *2+colwid  =  wd  then
wholeline = wholeline||spaces||hd1||spaces;
                else                wholeline                =
wholeline||spaces||hd1||spaces||" ";
                colwid = colwid+1;
                nextline    =    nextline||substr    (colval,
colwid+1, length (colval)-colwid)||del;
                n = n+1;
                go to lab;
           end;
linew:     proc (lab);
dcl  lab label;
                i = index (colval, ".");
                avg1 = nx (n)- (colwid-i+1);
                spaces = copy (" ", avg1);
                colval = colval||spaces;
                avg1 = i-1;
                avg1 = nx (n)+avg1;
                call linex;
                go to lab;
           end;
linec:     proc (lab);
dcl  lab label;
                if dataw (n) ^= 0 then do;
                     colval = colval||copy (" ", nx (n));
                     avg1 = colwid+nx (n);
                     call linex;
                     go to lab;
                end;
                else call lined (LAB3);
           end;
lined:     proc (lab);
dcl  lab label;
                spaces = copy (" ", avg (n));
                colval = colval||spaces;
                avg1 = colwid+avg (n);
                call linex;
                go to lab;
```

```
          end;
linex:    proc;
               spaces = copy (" ", (word (n)-avg1));
               wholeline = wholeline||spaces||colval;
               nextline = nextline||" "||del;
               return;
          end;
line9:    proc (lab);
dcl  lab label;
               spaces = copy (" ", word (n)-colwid);
               wholeline = wholeline || colval || spaces;
               nextline = nextline || " " || del;
               go to lab;
          end;
put:      proc (alph);
dcl  alph char (*) varying;
               put file (zz) edit (alph) (a);
               put file (zz) skip;
               return;
          end;
loop:     proc (zj, alph);
dcl  alph char (*) varying;
dcl (tt, zj) fixed bin (24);
               do tt = 1 by 1 while (substr (alph, zj, 1) ^=
" ");
                    zj = zj-1;
               end;
               return;
          end;
line8:    proc (lab, tt);
dcl  lab label;
dcl  tt fixed bin (24);
               tt = tt+1;
               go to lab;
          end;
hed2:     proc (lab);
dcl  lab label;
               tlin = tlin+1;
               tiarray (tlin) = wholeline;
               call put (wholeline);
               if nextchar = 1 then do;
                    if lab = LAB1 then j = j+1;
                    line = nextline;
                    nextchar = 0;
                    wholeline = "";
                    nextline = "";
                    n = 1;
                    go to lab;
               end;
               call line8 (LAB4, tlin);
          end;
valus:    proc (lab);
dcl  lab label;
```

```
                do i = 1 to ncol;
                     avg (i) = (word (i)-data (i))/2;
                     if 2*avg (i)+data (i)<word (i) then  avg
(i) = avg (i)+1;
                     nx (i) = avg (i)+dataw (i);
                end;
                go to lab;
           end;
ftnot:     proc (lab);
dcl  lab label;
                line = substr (line, 2, lin_len-2);
                hd = length (line);
                if hd+2<paper_width then do;
                     liny = "  "||line;
                     call put (liny);
                     call line8 (lab, pline);
                end;
                no = 1;
                no1 = paper_width-2 + no;
                call loop (no1, line);
                liny = "  "||substr (line, no, no1-no);
                call put (liny);
                pline = pline+1;
                title = substr (line, no, no1-no);
B7:             no = length (title)+no;
                if hd-no <paper_width then do;
                     liny = substr (line, no+1, hd-no);
                     call put (liny);
                     call line8 (lab, pline);
                end;
                no1 = no+paper_width;
                call loop (no1, line);
                title = substr (line, no+1, no1-no);
                call put (title);
                pline = pline+1;
                go to B7;
           end;
      end table;
```