(68)

DESIGN AND DEVELOPMENT OF DATA ENTRY SOFTWARE

DISSERTATION SUBMITTED TO THE JAWAHARLAL NEHRU UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF PHILOSOPHY
(COMPUTER SCIENCE)

D. PHANINATH

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110067

1987

11000001011000010
11000011111000011
11100010111100010
11010110011010110

ACS

APPLIED COMPUTER SCIENCES ORGANISATION PVT. LTD.

316, THIRD FLOOR, CHANDRALOK COMPLEX, SD ROAD, SECUNDERABAD-500 003. PHONE 70986 TELEX 0155-6105 ERA IN

July 2, 1987

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr.D.Phaninath has worked with ACS for the development of a Data Entry Package in FORTRAN on our SM-73 machine. His conceptualisation of the design of the package has been good. He has coded, debugged and demonstrated some modules of the package.

APPLIED COMPUTER SCIENCES ORGN. (P) LTD. HYDERABAD

(KA[illegible]DRA SHARMA)
Man[illegible] Director

DECLARATION

I hereby declare that the work presented in this dissertation has been carried out by me under the Supervision of Dr.S.Balasundaram and this has not been submitted to any other University for the award of any degree or diploma.

DATE: 7.7.87

(D.PHANINATH)

C E R T I F I C A T E

*This is to certify that the dissertation entitled "**DESIGN AND DEVELOPMENT OF DATA ENTRY SOFTWARE**" has been carried out by Mr. D. PHANINATH under my supervision.*

Prof. K.K. Nambiar
Dean,
School of Computer &
Systems Sciences,
Jawaharlal Nehru University,
NEW DELHI.

Dr. S.Balasundaram,
Asst. Professor,
School of Computer &
Systems Sciences,
Jawaharlal Nehru University,
NEW DELHI.

DATED:

A C K N O W L E D G E M E N T S

I wish to express my sincere appreciation and a deep sense of gratitude to Prof. S. Balasundaram for his constant encouragement and inspiring guidance.

I acknowledge specifically the excellent and extensive help given in providing Computer faciflities and giving useful suggestions by Dr. Kishore Budhiraju, Managing Director, of Applied Computer Sciences Organisation Pvt. Ltd., where this dissertation work was carried out. Thanks are due to Mr. Murali Krishna for his helpful suggestions. I would like to express my thanks to the Staff of ACS for their co-operation throughout my dissertation. Financial assistance from CSIR, New Delhi in the form of a JRF is gratefully acknowledged.

(D. PHANINATH)

A B S T R A C T

Data Entry Software is a Online Data Capture Package designed for SN or PDP Computer Systems working in RSX-11M Plus operating System environment.

It is an interactive package with neat screen displays. Top line is allotted for status message such as Field Number, Record Number and Name of the Field being Keyed-in. Second line is for error messages and for Data Input.

All Data Entry is done under the control of a format which should exist in the format file before any keying of Data Starts. Formats are themselves created under the Control of a Special format called 'Formatis'. The Software does several validation checks on Data before writing in a file. Formats are written in a Format file and Data in a Data file. The various validation checks are performed regarding Data Type, Character Range, Auto Duplication and Must Key option. Special care has been given to display Error messages. The display will be on the Screen till the user presses carriage return. Thus he will have sufficient time to understand the message. CTRL-Z is pressed to quit the package.

CONTENTS

CHAPTER - I

INTRODUCTION:

People have processed data ever since they became civilized. In the twentieth Century, however, mechanical and electronic aids to processing data have vastly increased in number and variety. The Chief Processor, the electronic computer, has flooded the modern society. Its usage has been so extensive that nearly every one now-a-days comes into contact with them in one way or another, as a business person in accounting, marketing, retailing or management, as an employee or customer of a firm that uses computers, as a citizen interested in the government's use of computers, and in so many ways.

Computers are used in both Business as well as Scientific data-processing. $Data_1$ is a general term used to denote any or all facts, numbers leters and symbols that refer to or describe an object, idea, condition, situation, or other factors.

A data processing job_2 is defined as any operation or combination of operations on data, including every thing that happs to data from the time they are observed or collected to the time they are destroyed. The purpose of data processing is to accept "raw" data and through processing operations convert those data into useful information.

Business data processing includes administrative applications (e.g. personnel, payroll, accounting) as well as broader business applications

(e.g. inventory control, sales forecasting etc). Most of the Business applications and few scientific applications of computers (e.g Numerical weather predictions) involve large amounts of data-input for processing. In such situations Data Entry Software (DES) is used for efficient , quick input of data. Ideally DES basically does the work of writing or reading of Information or Instructions to a processing medium for data processing.

Data Entry is basically of two types - Transcriptive Data Entry and Source Data Entry.

1.1 ***Transcriptive Data Entry:*** *This type covers all data preparation devices where data, prepared on documents at their source or origin, is than transcribed to another medium that is capable of being read and interpreted by a computer. In this category are the following data preparationdevices: Card punches, paper tape punches, magnetic tape encoders. key-to-key systems, key-to-disk systems, On-line key punch/verify systems and Magnetic character readers.*

1.2 ***Source Data Entry:*** *For devices in this category, data is prepared at its source in a machine-readable form such that it can be directly read by a computer without the requirement of an intermediate data transcription step. Data Entry Techniques that fall into this category*

Optical character reading, optical mark reading and the direct-entry of Information into a computer using terminals at the point of origin of the data.

The present Data Entry Software comes under the second type.

Before we study how the Sotftware was designed and developed, it will be helpful to have an overview of the Software.

1.3 **Overview of the present Software:** Two modes of Data Entry are present: (1) Entry mode and (2) Verify mode. In Entry mode data is entered under the control of a format. A format specifies Field descriptors such as Field Name, Field Size, Data Type, Auto Duplication, Justification Option, Must Key Option, Range of Data, Verification Option and Fill character option, for each field of data. In verify mode, data is verified character by character. The present software deals with Entry mode only.

Before entering Data, a format is created based on the nature of data to be keyed-in. Format creation is also done under a special format called 'Formatis".

User is promoted to give Job name, continue option ('yes') for continuing a previous job), Batch number (each job is divided into various

batches), Format Name under the control of which, the data is to be keyed-in and the operator's initials for his Identification.

The user can start keying when the prompt for Input appears in the screen. To know which field the user is keying, the field name is displayed on the screen. Any error in data such as error in Data Type, Range-fault etc., are displayed on the second line of the screen.

The user while creating a Format, has to give the Record length of his file. Records are numbered by a five-digit number. At the end of data entry CTRL-Z is typed to quit the package.

CHAPTER - II

PROGRAM DESIGN AND DEVELOPMENT[4]

According to Webster, the process of design involves "Concerning and Planning out in the mind" and "making a drawing, pattern or sketch of". Design is the bridge between Software requirements and an implementation that satisfies those requirements[3].

2.1 Definition of the Problem :

2.1.1 ***Statement of the problem:*** *The statement of a problem is a basic step in Design as it provides a realistic appraisal of the effort involved, inspite of not being precise.*

The present problem can be stated as to design a On-line Data Capture Package for SN/PDP Computer Systems, working in RSX-11M Plus Operating System.

In order to develop a Data Entry System which is stated above, the following goals need to be defined.

2.1.2 ***Definition of Goals:*** *(1) To design a Format - creation module.*

A format defines the field attributes such as Field Name, Field Size, Field Type, Auto Duplication, Must Key, Range, Justification, Verification and Fill Character. A format also specifies record length, Level (of a Format) Data of creation and a Coment field. The formats are to be written into a format file.

2) To design a Data Creation Module which accepts data under the control of a Format, validates the data and writes it in a Data file.

Besides the above two, a sub-goal which need to be mentioned is that Error Messages should be displayed on the top of the screen than bottom (as it is most often done) so that the operator notices it easily while keying-in Data.

2.2) ***Problem Analysis:*** *Problem Analysis is necessary to translate the original statement of the problem into a computer program written in a specific language.*

2.2.1) ***Development of Program Structure:*** *The primary consideration have is to identify segments and subroutines. A segment is a distinct portion of the program that can be considered largely independent of other portions. Segments are independent to the extent that they accept a set of data for processing and transmit results to the next segment.*

To develop a program structure the problem needs to be analysed. First preliminary analysis is made and it is refined further at a later stage. The final refined analysis is presented below.

Broadly speaking there are two modes of Data Entry viz., Entry and Verify. Verify Mode, though not covered in this package, is included for future use. The screen display which prompts the user to give his option is described below. The prompt 'Mode:' is displayed in the centre of the screen with the choices viz., Entry (default), Verify and Quit, on the right-side of the above prompt. The cursor is placed next to the prompt to read the user's option. The user need only enter the first two letters of his option mode ('En', 'Ve' or 'Qu').The Quit option is for quitting the package. The present display is shown in Fig.1. This whole display and reading the user's option of the Mode of Data Entry forms a Segment.

As it happens most of the time, data of a job will be so large that it requires several sessions for its entry. To allow such cases, an option called 'Continue' is provided. The choices are 'Yes' or 'No', 'No' being the default option. The user need to key-in the first letter of his option. The display is shown in Fig.2. It may be noted that the Mode Choosen. by the user is displayed above the 'Continue:' prompt. A 'Yes' option is choosen in the case where the data of an old job is being entered.

MODE:ENTRY

DEFAULT=ENTRY

ENTRY
VERIFY
QUIT

Fig.1

MODE: ENTRY
CONTINUE: NO

DEFAULT=NO

YES/NO

Fig.2

For new jobs, 'No' (default) option is choosen. Thus continue option can form another segment.

Each job is identified by a job name. Here job means a collection of data of a particular type for Data Entry. The job name is entered by the user to the prompt 'Job Name:'. Only six characters are allowed for the job name. It is stored in an array called 'Jbnm'. A job name is stored as the first element of a row. The rest of the row, contains the Batch numbers of the Batches which come under a job. The Batch number concept is explained below:

A job is divided into several Batches with Batch Numbers, depending upon the nature of data. The Batch number of the data to be entered, is to be provided by the user the prompt 'Batch No.:' Only three digit number is allowed.

The Fig's 3 and 4 display the job name and Batch number screens. Both Job Name and Batch Number options form a segment each.

The Data Entry is done by an operator. An operator is paid a salary based on the amount of Data entered by him. Hence to maintain operator's statistics, his initials should be noted down. This can be done by the prompt 'Operator:' This can be another segment.

MODE: ENTRY
CONTINUE: NO
JOB NAME: SBI-DD

DEFAULT=JOB-1

Fig.3

MODE: ENTRY
CONTINUE: NO
JOB NAME: SBI-DD
BATCH NO: 001

Fig.4

Data Entry is done under the control of a Format. The format's name should be provided by the user in response to the Prompt 'Format:' The format should already exist in memory in the Format file where all formats are stored. To facilitate easy search the following file structure for format files is adopted. There are three types of format files at three different levels. At the top level there is a format header file which contains the names for formats, which are already created. Below this level, there is another file called 'Level file' containing level and the name of the corresponding file containing the level.

The file containing the format name is searched. If the name is found then next prompt appears; Otherwise an error message is displayed and Format name is asked again.

Format option thus can be another segment.

The final prompt is level prompt. A job may involve more than one format.All the formats for one particular job are kept in one format file with different levels. Only 6 levels are allowed. Level option is another segment in the Software. The level number entered is checked against the numbers of levels existing in the file. If not found, an error message is displayed and level prompt appears again.

Fig's 5 and 6 show the operator, Format and Level Screens. Fig. 6 shows the final screen display at the end of Level Option.

MODE: ENTRY
CONTINUE: NO
JOB NAME: SBI-DD
BATCH NO: 001
OPERATOR:DPN
FORMAT :SBI-DD

DEFAULT=JOB NAME

Fig. 5

MODE: ENTRY
CONTINUE: NO
JOB NAME: SBI-DD
BATCH NO: 001
OPERATOR:DPN
FORMAT :SBI-DD
LEVEL :1

DEFAULT=FIRST LEVEL

Fig. 6

Now after acquiring the preliminary information, the actual data entry must start. In Data Entry we have two divisions viz., Format Creation and Data Creation. Hence it should be ascertained which division of Data Entry being done.

Format Creation is done under the control of special format called 'Formatis'. Hence the Format Name Keyed in by the user in response to the Format Option is checked if it is 'Formatis'. If the Format Creation is being done, then a subroutine 'Frmtis' is called which provides the format 'Formatis'.

An array called 'Buffer' of size 512 characters is used to store the data, which is keyed-in by the user. First initial record number (i.e., 00001) is written in the array.

We shall digress for the present to study about Data Creation.

If Data Creation is being done, then first 'Continue' option is checked. If continue is 'Yes', then the old Data File is opened. It may be noted here that all Data files are named by the Job name plus Batch number. The record number of the last record is read and then it is incremented by one and written in Buffer. Then the file containing the choosen format

is opened. The header record of the format-file is read, which contains Record Sign of Data file (which is going to be created).

From this point onwards, Format Creation and Data Creation follow the same flow of control except for a few minor changes.

The already opened format file is read a record at a time.Each record contains various field descriptors (of the Data Field which is to be keyed-in by the user). Each field descriptor is detailed below:

i) Field Name: Can contain 12 characters.

ii) Field Size: Can vary from 1 to the length of record.

iii) Data Type: Three types are allowed - Numeric ('N'),Alphabetic ('A') and unspecified ('U').

iv) Must Key: This field tells whether the field must be keyed or can be filled with a Fill character. Options: Yes ('Y') or No ('N', default).

v) Starting value and End value: Specify the range for the field. Maximum six characters are allowed.

vi) Verification: Tells whether the field is to be verified in Verify mode or not. Options: Yes ('Y', default) or No ('No').

vii) Justification: Tells whether the field should be left ('L') or right ('R'). Default is 'L' for Type 'N' and 'R' for Types 'A' and 'U'.

viii) *Auto Duplication: Option yes ('Y') for copying the contents of the corresponding field of previous record into the present field; otherwise No ('No', dafault).*

ix) *Fill Chnaracter: To fill the unkeyed characters in a field with zeros ('Z', dafault for 'N' Type) or Blanks ('B', default for 'A' and 'U' Types).*

After reading the above descriptors of the present field, the data keying will be done.First, Auto Duplication option is checked. If 'Yes' then, the previous record is read and the corresponding field is copied. Then another record is read from format field which contains descriptors, for the next field. If Auto Dup is 'No', then Data Keying is prompted by 'Input Data='. On the screen, field number is shown on the left, record number on the right , while Field name in the middle. The user keys in Data. It is checked for the type, character by character. If a type mismatch occurs then anerror message is flashed and the field is prompted to be rekeyed. The previous data is rejected.

In the case of Format Creation, the data entered has to be checked if it belongs to the allowed list of characters. For instance, for 'Type' only 'A', 'N', or 'U' are allowed.This is termed 'List-Checking', while list-checking, if any error has occured, then it will be flashed on the screen and the data has to be rekeyed.

Next for character data, character-range, for eg., from say I to N or A to K etc., is checked if present. As usual error message is flashed if an error occurs.

In the case of Format Creation, the default cases such as Must Key (default 'N'), Verification (default 'Y') etc. must be taken care of. This is done next to character-range checking.

Afterwards Must-Key validation is done. If must key is 'Yes' and no data is keyed-in then an error message is displayed.

Next the partially filled fields are filled with Fill Character.

The Software has two major counters - Record Counter and Buffer Counter. Record counter and Buffer counter are incremented by the size of the field, at the end of data input for eachfield. When Record Counter becomes equal to the record sign, then a subroutine named 'ADDRNO' is called to increment the present record number by one and is written in the Buffer. When Buffer counter equals 512 characters (which is its capacity) then the data is written into appropriate file. All files are sequential in this Software.In the case of data creation,if continue is 'yes',then relevent old file is opened and the present data is appended to it.Otherwise,

data is written into a new data file. In the case of Format Creation, Format header file, level file and the Format files are appended. The program ends by typing CTRL-Z.

Thus in the present design, stress has been on segmentation of the problem into several blocks for easy debugging and understanding.

2.2.2 **Pseudo Code**:- *The Pseudo Code describing the Program structure is given below:-*

```
*       Mode option segment
        Display Mode Option Screen
Mod:    Read Mode
        If (Mode ≠ Entry, verify or quit) then
            Display Error message
            Go To Mod
        Endif

*       Continue Option Segment
        Display Continue Option Screen
Contin: Read continue
        If (Continue ≠ yes or no) then
            Display Error Message
            Go To Contin
        Endif
*       Job Name Option Segment
        Display Job Name Option Screen
        Read Job Name
```

```
*       Batch Number Option Segment
        Display Batch number screen
        Read Batch Number
*       Operator Option Segment
        Display Operator Screen
        Read Operator's Initials
*       Format Option Segment
        Display Format Screen
Fmt:    Read Format
        Search the Format heeader file for the present Format.
        Open the Old Format header file
            Search for the present Format
            If (the Format not found) then
                Display Error Message
                Go To Fmt
            Endif
            Close the Format header file
*       Level Option Segment
        Display Level Option Screen
Lvl:    Read Level
*       Search the Level file to find out if the present level exists
*       or not
        Open the Level file
            Searech the Level file for the present level
            If (the level doesn't exist) then
                Display Error Message
                Go TO Lvl
            Endif
        Close the level file
```

```
*       Data Creation block starts from here. There are two division
*       in Entry Mode - Format Creation (for which "Formatis" is the
*       Format used) and Data Creation (user created format is used).
If (Format Creation is being done) then
    Write Record Number (00001) in Buffer.
Endif
If (Data Creation is being done) then
    If (Continue = Yes) then
    Open the previous Data file
        Read the last Record No.
        Increment the Record No.
    Close the Data file
    Write the new Record No. in Buffer
  Endif
Endif
Open the format file
Read the header record to note Record size
Read a record containing the field descriptors
If ((Record Counter + Field size)>Record size) then
  Fill the remaining spaces with blanks
  Increment Record number
  Write the new record no. and Buffer
Endif
```

```
*       Auto Dup Implementation
        If (Auto Dup = yes) then
            If (the field is in first record of the session) then
                Copy the contents of the corresponding field from
                                the old Data file
            Else
                Copy the contents of the corresponding field from
                                the last record
            Endif
        Endif

*       Reading the input data
Input: Read Input
*       Type checking
        Repeat for each character
            If (the ASCII value of the Input datum is not in the
                                ASCII range of the type) then
                Display Error Message
                Go To Input
            Endif
        Endofrepeat

*       List checking for format creation
*       For those field descriptors which have a list of characters as
*       their qualifiers, list checking is done
```

```
Repeat for the fields which are to be list checked
    If (the input is not in the list) then
        Display Error Message
        Go To Input
    Endif
Endofrepeat
*   Character - range checking
Repeat for each character of Input
    If (Input is out of range) then
        Display Message
        To Go Input
    Endif
Endofrepeat

*   Taking care of those fields for which default options are provided.
Repeat for each field with a default option
    If (carriage Return is pressed) then
    Accept the default option
    Endif
Endofrepeat

*   Must - key validation
If (must key = yes) then
    If (field counter = 0) then
        Display Error Message
        Go To Input
    Endif
Endif
```

```
*       Filling non-keyed spaces with Fill Character
        If (Justification is 'left') then
            Left justify Data
            If (Fill = 'Blank') then
                Fill with Blanks
        Else
            Fill with zeros
        Endif
        Else
            Right justify Data
            If (Fill = 'Blanks') then
                Fill with Blanks
            Else
              If (Fill = 'Zero') then
                    Fill with zeros
            Endif
        Endif
*       Field counter = 0
        Buffer Counter = Buffer Counter + Field Counter
        Record Counter = Record Counter + Field Counter
        If (Buffer Counter = 512) then
            If (Format Creation is being done) then
                Open Format file
                    Write the Buffer
                Close Format File
```

```
            Open Format Header file
                Write the Format name
            Close the Header file
            Open Level file
                Write the level and Format file name
            Close the level file
    Else
        If (continue = 'yes') then
            Open the old Data file
                Write the Buffer
            Close the file
        Else
            Open a new data file
                write the Buffer
            Close the data file
        Endif
    Endif
```

2.2.3 **The Computing Environment:** The Software is developed on SN-73 (PDP - 11 compatible) System. The operating system is RSX-11M Plus. The language used is PDP-ll FORTRAN-77 which offers all the features of Standard FORTRAN-77.

To transform a PDP-11 FORTRAN-77 source program into an executing task[5], one need to (a) compile the program (b) Task-build the program, to link the object module with necessary external routines and (c) Execute the program.

2.2.4 **Writing the Program**[6,7]: The Program has been written in PDP-11 FORTRAN 77. FORTRAN 77 offers standard structural constructs.

The program incorporates various blocks or segments for clarity and understanding the code. The blocks are: Variable identification block. Type Declaration and Storage allocation block, Main Program block and Error-handling block. Main Program itself is a major block which is divided into various segments such as Mode Segment, Continue Segment, Job Name Segment etc. Variable Identification block facilitates identification of variables used in the code. Besides, several comments are provided for clarity in understanding. Error handling block takes care of all major errors in Data Entry.

The most notable feature is the usage of VT 100 Terminal Sequences for cursor positioning, erasing the screen etc. These terminal sequences

In all the displays. For instance, to position the cursor, 'ESC [X; Y H' Squence is used. These are fed through FORTRAN. Subroutines curpos (cursor positioning), clrsrn (clear screen), and Eralin (Erase line) use these sequences.

Error messages are displayed in the second line of the screen. After the display, the cursor stays at the end of the message. To clear the message and get back to the program, user has to press carriage return. This gives the user ample time to read and understand the message.

CHAPTER - III

PROGRAM TESTING:

The program has been tested using the Orissa Secondary School Board (OSSB) data.

The output listing - 1 contains format and the output listing-2 contains these data created under the control of the above format.

Each record on the format is of length 38 characters. The format tests all the features of the Software development. For instance, the field GSC-No. is a Auto Dup field, field APP-AS uses character-range checking, most of the fields are must key fields and all fields use verification, justification and Fill character options.

The record length of the Data file has been choosen to be 128 characters. Four records have been keyed-in, which sum up to 512 characters, the optimum capacity of Buffer.

The Data Created under the format is shown in Appendix - C.

The first field i.e., GSC-No. is an Auto Dup field and hance we have the same data i.e., 00000 in all the four records. Actually this field data is not provided by OSSB and hence it was filled with zeros, making it Auto Dup.

The fields CAN-NAME (candidate name) and FAT-NAME (Father's Name) are left justified and filled with blanks in accordance with the

format specifications for these fields. The field APP-AS has to be 'A' for this Internal group of candidates. For this, character range has been used. The starting value and the End value are both 'A'.

The field EXT-OPT is not a Must-key field. Hence the fourth record does not contain any information in this field, while the above three contain 'BIO'. All the three Data types viz., 'A', 'N' & 'U' are used.

Thus the data tests all the features available with the present software.

CHAPTER - IV

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK:

The present Software is an open-ended package and is a modest effort to develop a reasonably good DES. It lacks some of the features of a Standard DES. Hence it can be developed further by incorporating several other modules such as Numerica-Ranege Checking, Hash Total Option, Global Field Option etc., which are general features of a Standard DES.

Also the present work can be expanded further by adding 'Verify Mode Program' for the verification of the entered Data, Insert/Delete Module for Insertion or Deletion of Data in a Data Files and Help Module for Online help.

REFERENCES

1) *Computer Dictionary & Handbook*

- *Charles J. Sippl & Roger J. Sippl. Howard W.Sams & Co. Inc. USA, PP.126*

2. *Mc Graw-Hill Dictionary on Computers*

- *Sybil P. Parker (Editor in Chief) PP. 96.*

3) *Software Engineering Concepts*

- *Richard Fairly, PP. 137 International Student Edition.*

4., *Techniques in Computer Programming*

- *Philip M. Sharman, P.309 Prentice-Hall Inc. New Jersey.*

5) *PDP-11 FORTRAN-77 User's Guide.*

6. *PDP-11 FORTRAN Language Reference Manual.*

7) *FORTRAN-77: A structured, Disciplined Style*

- *Davis & Hoffmann Internation Student Edition.*

A P P E N D I X - A

THE PROGRAM LISTING

```
*****************************************************************************
*******          D A T A   E N T R Y   S O F T W A R E
*****************************************************************************
*
*******                VARIABLE IDENTIFICATION
*
*INTEGER VARIABLES:
*
*                  I  = BUFFER SUBSCRIPT
*                  J  = DUPBUF SUBSCRIPT
*                 I1  = UNIT NO
*                 I2  = USED IN JOBNAME ARRAY
*                 I3  = USED IN JOB NAME ARRAY
*                 I4  = USED FOR DEFAULT JOB NAME
*                 I5  = USED IN CHECKING JOB NAME-LIST
*                 I6  = ----------DO----------------
*                 I7  = ----------DO----------------
*                 I8  = USED IN BATCH NO ARRAY
*                 I9  = ----------DO---------
*                 I10 = FORMAT COUNT
*                 I11 = FMT
*               FCOUNT= FIELD COUNT
*               RCOUNT= RECORD COUNT
*               BCOUNT= BUFFER COUNT
*               RECNO = RECORD NUMBER
*              RECSIZ = USED TO STORE RECORD SIZE OF DATA RECORDS IN
*                       INTEGER FORM NOTE THAT RSIZE STORES THE SAME
*                        IN CHARACTER FORM.
*              FLDSIZ = USED TO STORE FIELD NO.IN INTEGER FORM.NOTE
*                       THAT FSIZE STORES THE SAME IN CHARACTER FORM
*              FLDNO  = FIELD NUMBER
*              NRECNO = SESSION RECNO NUMBER(RECORD COUNT FOR A NEW
*                       SESSION).
*              NRANGE = USED TO STORE NUMERIC RANGE
*              IRANGE = INTEGER RANGE
*                 JNUM= USED IN DEFAULT JOBNAME
*                JUST1= USED IN IMPLEMENTING JUSTIFICATION
*                JUST2= -------------DO-----------------
*                JUST3= -------------DO-----------------
*               LIMIT = USED TO STORE DO LOOP RANGE
*                 LVL = LEVEL
*
*******                    CHARACTER VARIABLES
*
*                  ADUP = AUTO DUPLICATION
*                BUFFER = A BUFFER TO STORE THE KEYED-IN DATA
*                BATCH  = TO STORE BATCH NO.'S
*                CONTIN = TO STORE THE CONTINUE OPTION
*                COMENT = TO STORE THE CONTENTS OF COMMENTS - FIELD IN
*                         FORMAT- HEADER RECORD
*                CHAREC = KEYED-IN CHARACTER
*                DUPBUF = TO STORE THE CONTENTS OF A RETRIEVED FILE
*                  DATE = REPRESENTS DATE
*                  EVAL = END VALUE OF A RANGE
*                FIELD  = TO STORE THE DATA-FIELD KEYED-IN
```

```
*                FNAME = FORMAT NAME
*                 FMAT = FORMAT NAME
*               FOMAT  = FORMAT NAME
*               FRMAT1 = FORMAT FILE NAME(FORMAT NAME.FMT)
*               FRMAT2 =          ---- DO ----
*               FORMAT = USED TO STORE THE RETRIEVED FORMAT NAME
*                        FROM THE 'FORMAT.HDR' FILE.IT IS DIFFERENT
*                        FROM FNAME - FORMAT IS DIMENSIONED BUT
*                        FNAME IS NOT.
*                FSIZE = FIELD SIZE
*                 FILL = FILL CHARACTER
*               FILENM = FILE NAME OF DATA FILE
*               FLDNAM = FIELD NAME
*                INPUT =
*                 JUST = JUSTIFICATION
*                 JBNM = JOB NAME
*                LEVEL = REPRESENTS THE LEVEL OF FORMAT
*               LVLFLE = FILE NAME CONTAINING A LEVEL OF A FORMAT
*                 MKEY = MUST KEY
*                 MOD  = MODE OF OPERATION
*                 MODF = MODE OF OPERATION
*                 OPTR = OPERATOR'S INITIALS
*                RSIZE = RECORD SIZE OF DATA FILE
*                 SVAL = STARTING VALUE OF A RANGE
*                 TYPE = DATA TYPE OF THE KEYED-IN DATA
*                VERIF = VERIFICATION
*                 VAR1 = USED IN DEALING WITH ERROR MESSAGES
*                 VAR2 = USED IN DEALING WITH DEFAULT CASES IN
*                        DATA CREATION.
*
*************          CONSTANT IDENTIFICATION              ***************
*
*                 SP    = SPACE
*               DOTDAT = STORES '.DAT' CHARACTER STRING
*               DOTFMT = STORES '.FMT' CHARACTER STRING
*               JOBVAR = STORES 'JOB-' CHARACTER STRING ,USED
*                        FOR JOBNAME-LIST.
*               FRMAT3 = STORES 'FORMATIS' CHARACTER STRING
*
************     TYPE DECLARATION AND STORAGE ALLOCATION     ************
*
*** BELOW JOBNAME IS DECLARED AS AN ARRAY OF 100 AND TOTAL NO
*** OF FORMATS OF 50 AND UNDER EACH JOB ONLY 25 BATCHES ARE
*** ALLOWED.ALSO NOTE THAT MAXIMUM NO OF A FIELD IS RESTRICTED
*** TO 100 AS 'NRANGE' IS DECLARED AS AN ARRAY OF 100
*
*
0001            INTEGER       BCOUNT,SP,FLDSIZ,FCOUNT,FLDNO,IRANGE
0002            INTEGER       IBCONT,JRCONT,JNUM,JUST1,JUST2,JUST3,LVL
0003            INTEGER       NRECNO,RCOUNT,RECSIZ,RECNO
*
0004            CHARACTER*1   ADUP,BATCH,BUFFER,CHAREC,COMENT*14,CONT,CONTIN*3
0005            CHARACTER*6   DATE,DUPBUF*1,DUP*1,DOTDAT*4,DOTFMT*4,EVAL
0006            CHARACTER*3   EVL,FIELD*80,FSIZ,FLDNM*12,FNAME*9,FLDNAM*12
0007            CHARACTER*1   FMAT*9,FILL,FIL,FORMAT*9,FRMAT1*13
```

```
0008            CHARACTER*1  FRMAT2*13,FILENM*13,FOMAT*9,FRMAT3*8,FLDSZE
0009            CHARACTER*1  INPUT*5,JBNM*6,JOBVAR*6,JUS,JUST,LEVEL,LEVL
0010            CHARACTER*1  LVEL,LVLFLE*10,MKEY,MKY,MODE*2,MOD*6,NRANGE
0011            CHARACTER*1  NOREC*5,NUMREC*5,OPTR*3,RSIZE*3,SVAL*6,SVL*6
0012            CHARACTER*1  TYP,TYPE,VERIF,VERF,VAR1,VAR2
        *
        *******                        DIMENSION STATEMENTS
        *
0013            DIMENSION  BATCH(20)
0014            DIMENSION  BUFFER(512)
0015            DIMENSION  DUPBUF(512)
0016            DIMENSION  FORMAT(50)
0017            DIMENSION  JBNM(10)
0018            DIMENSION  RECNO(5)
0019            DIMENSION  CHAREC(80)
        *
        ********************************************************************************
        *
        *INITIALIZATION:
        *
        *INTEGERS:
0020            I =1
0021            I1=5
0022            I2=0
0023            I3=25
0024            I4=0
0025            I8=1
0026            I10=0
0027            FCOUNT=0
0028            RCOUNT=0
0029            BCOUNT=0
        *
        ******* BELOW SP IS INITIALIZED TO ITS ASCII DECIMAL VALUE    ********
        *
0030            SP        = 032
0031            RECNO(1)  = 0
0032            RECNO(2)  = 0
0033            RECNO(3)  = 0
0034            RECNO(4)  = 0
0035            RECNO(5)  = 1
0036            NRECNO    = 1
0037            FLDNO     = 1
        *
        *CHARACTERS:
        *
0038             DOTDAT='.DAT'
0039             DOTFMT='.FMT'
0040             JOBVAR='JOB-'
0041             FRMAT3='FORMATIS'
        *
        ***********************MAIN PROGRA********************************************
        ********************************************************************************
        *
        *
        *****************    MODE SEGMENT    ***********************************
```

```
      *
0042            CALL CLRSRN
0043           WRITE (5,10)
0044       10  FORMAT(////,50X,'DEFAULT = ENTRY',/,50X,'ENTRY',/,
          1 50X,'VERIFY',/,50X,'QUIT')
0045           CALL CURPOS(0,0)
0046       20  WRITE(5,25)
0047       25  FORMAT(///,20X,'MODE=',$)
0048           READ (5,30) MODE
0049       30  FORMAT(A2)
0050           IASC=ICHAR(MODE)
0051           IF((IASC.EQ.SP).OR.(MODE.EQ.'EN')) THEN
0052           CALL CURPOS(0,0)
0053           WRITE(5,26)
0054       26  FORMAT(///,20X,'MODE = ENTRY')
0055             MOD='ENTRY '
0056             ELSEIF (MODE.EQ.'VE') THEN
0057              CALL CURPOS(0,0)
0058               WRITE(5,32)
0059      32        FORMAT(///,20X,'MODE = VERIFY')
0060                MOD='VERIFY'
0061                ELSEIF (MODE.EQ. 'QU') THEN
0062                WRITE(5,33)
0063      33         FORMAT(///,20X,'MODE = QUIT')
0064           ELSE
0065              CALL CURPOS(0,0)
0066              TYPE*,' '
0067              WRITE(5,35)
0068      35      FORMAT(25X,'INVALID CHARACTER',$)
0069      40      READ(5,38) VAR1
0070      38      FORMAT(A)
0071              IDN=ICHAR(VAR1)
0072            IF (IDN .EQ.SP) THEN
0073             CALL CURPOS(0,0)
0074             TYPE*,' '
0075             CALL ERALIN
0076             GOTO 20
0077            ELSE
0078             GOTO 40
0079            ENDIF
0080           ENDIF

      *
      ***************        CONTINUE SEGMENT                    ******************
      *
0081           CALL CLRSRN
0082           WRITE(5,50)
0083        50 FORMAT(/////,50X,'DEFAULT = NO',/,50X,
          1 'YES/NO')
0084        55  CALL CURPOS(0,0)
0085           WRITE(5,56) MOD
0086        56 FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE =',$)
0087           READ (5,60) CONT
0088        60 FORMAT (A1)
0089           IDN = ICHAR(CONT)
```

```
0090            IF((IDN.EQ.SP).OR.(CONT.EQ.'N')) THEN
0091            CALL CURPOS(0,0)
0092            WRITE(5,61)
0093      61    FORMAT(////,20X,'CONTINUE = NO')
0094             CONTIN = 'NO'
0095             ELSEIF(CONT.EQ.'Y') THEN
0096              CALL CURPOS(0,0)
0097              WRITE(5,62)
0098      62      FORMAT(////,16X,'CONTINUE = YES')
0099              CONTIN = 'YES'
0100            ELSE
0101             CALL CURPOS(0,0)
0102             CALL CURPOS(2,25)
0103             WRITE(5,63)
0104      63     FORMAT(25X,'INVALID CHARACTER',/)
0105      70    READ(5,71) VAR1
0106      71    FORMAT(A)
0107              IASC=ICHAR(VAR1)
0108            IF (IASC.EQ.SP)  THEN
0109             CALL CURPOS(0,0)
0110             TYPE*,' '
0111              CALL ERALIN
0112              GOTO 55
0113             ELSE
0114              GOTO 70
0115             ENDIF
0116            ENDIF

        *
        *********          JOB NAME  SEGMENT          ****************
        *
0117           CALL CLRSRN
0118           WRITE(5,80)JNUM
0119        80  FORMAT(//////,50X,'DEFAULT=JOB-',I2)
0120            I2=I2+1
0121        85  CALL CURPOS(0,0)
0122        82  WRITE (5,83) MOD,CONTIN
0123        83  FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
           1        'JOBNAME=',/)
0124            READ(5,90) JBNM(I2)
0125        90  FORMAT(A6)
0126            IASC = ICHAR(JBNM(I2))
0127            IF (IASC.EQ.SP) THEN
0128            CALL CURPOS(0,0)
0129            WRITE(5,95) JNUM
0130        95  FORMAT(/////,20X,'JOB NAME = JOB-',I2)
0131            ENCODE(6,97,JBNM(I2)) JOBVAR,JNUM
0132        97  FORMAT(A4,I2)
0133            I4=I4+1
0134            ELSE
0135            CALL CURPOS(0,0)
0136            WRITE(5,100) JBNM(I2)
0137       100  FORMAT(/////,20X,'JOB NAME=',A6)
0138             ENDIF
        *
```

```
      ***************      BATCH NUMBER      ******************
      *
0139   165   CALL CLRSRN
0140   170   WRITE(5,172)MOD,CONTIN,JBNM(I2)
0141   172   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,
     1        /,20X,'JOB NAME=',A6,/,20X,'BATCH NO.=',$)
0142         I8= I8+1
0143         READ (5,175) BATCH(I8)
0144   175   FORMAT(A3)
      *
      *************          OPTERATOR SEGMENT          ******************
      *
0145   200   CALL CURPOS(0,0)
0146         WRITE(5,205)MOD,CONTIN,JBNM(I2),BATCH(I8)
0147   205   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
     1          'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
     1          'OPERATOR=',$)
0148         READ(5,210) OPTR
0149   210   FORMAT(A3)

      *
      ***********          FORMAT SEGMENT                ******************
      *
0150         CALL CLRSRN
0151         WRITE(5,220)
0152   220   FORMAT(/////////,50X,
     1 'DEFAULT = JOBNAME')
0153   225   CALL CURPOS(0,0)
0154         WRITE(5,227) MOD,CONTIN,JBNM(I2),BATCH(I8),OPTR
0155   227   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
     1          'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
     1          'OPERATOR=',A3,/,20X,'FORMAT=',$)
0156         READ (5,230) FMAT
0157   230   FORMAT(A9)
0158         I10 = I10 + 1
0159         IDN = ICHAR(FMAT)
0160         IF (IDN.EQ.SP) THEN
0161         CALL CURPOS(0,0)
0162         WRITE(5,235) MOD,CONTIN,JBNM(I2),BATCH(I8),OPTR,
     1              JBNM(I2)
0163   235   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
     1          'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
     1          'OPERATOR=',A3,/,20X,'FORMAT=',A9)
0164         ENCODE (6,236,FMAT) JBNM(I2)
0165   236   FORMAT(A6)
0166         ELSE
0167         CALL CURPOS(0,0)
0168         WRITE(5,237) MOD,CONTIN,JBNM(I2),BATCH(I8),OPTR,FMAT
0169   237   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
     1          'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
     1          'OPERATOR=',A3,/,20X,'FORMAT=',A9)
0170         ENDIF
      *
      *******      FIRST 'FORMAT.HDR'FILE IS SEARCHED FOR THE FORMAT
      *
```

```
0171            OPEN(UNIT=1, FILE='FORMAT.HDR',STATUS='NEW',
               1        RECL=9)
0172            WRITE (UNIT=1,FMT=238) FRMAT3
0173      238   FORMAT(A9)
0174             BACKSPACE 1
          *
          *******  FORMAT.HDR FILE IS SEARCHED HERE
          *
0175            DO 240 I11=1,I10+1
0176             READ(UNIT=1,FMT=238) FORMAT(I11)
0177             IF (FMAT.EQ.FORMAT(I11)) GOTO 245
0178      240   CONTINUE
0179      241   CALL CURPOS(0,0)
0180            TYPE*,' '
0181            WRITE(5,242)
0182      242   FORMAT(20X,'FORMAT NOT FOUND',$)
0183      243   READ(5,244) VAR1
0184      244   FORMAT(A1)
0185            IDN=ICHAR(VAR1)
0186            IF(IDN.EQ.SP) THEN
0187            CALL CURPOS(0,0)
0188            TYPE*,' '
0189            CALL ERALIN
0190             CLOSE (UNIT=1,DISPOSE='SAVE')
0191             GOTO 225
0192            ELSE
0193             GOTO 243
0194            ENDIF

          *************                    LEVEL                              *******
          *
          ******* IF FORMAT IS 'FORMATIS' THEN GOTO 310, AS NO
          ******* LEVEL EXISTS.
          *
0195      245   CLOSE(UNIT =1,DISPOSE='SAVE')
0196            IF (FMAT.EQ.FRMAT3) GOTO 315
0197            CALL CLRSRN
0198            WRITE(I1,260)
0199      260   FORMAT(/////////,50X,'DEFAULT= FIRST LEVEL')
0200            CALL CURPOS(0,0)
0201            WRITE(I1,265)MOD,CONTIN,JBNM(I2),BATCH(I8),OPTR,FMAT
0202      265   FORMAT(///,20X,'MODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
               1        'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
               1        'OPERATOR=',A3,/,20X,'FORMAT=',A9,/,20X,'LEVEL=',$)
0203      270   READ(I1,275)LVL
0204      275   FORMAT(I1)
0205            IF ((LVL.LT.0).OR.(LVL.GT.6)) THEN
0206             CALL CURPOS(0,0)
0207             TYPE*,' '
0208             WRITE (I1,277)
0209      277    FORMAT(20X,'LEVEL OUT OF RANGE',$)
0210             GOTO 290
0211            ENDIF
0212            ENCODE(13,280,FRMAT1)FMAT,DOTFMT
0213      280   FORMAT(A9,A4)
```

```
0214            OPEN(UNIT=2,FILE=FRMAT1,RECL=11,
               1 STATUS='NEW',ERR=1020)
0215            IDN=ICHAR(LEVL)
0216            IF(IDN.EQ.SP) THEN
0217            LVL=1
0218            ENDIF
0219            ENCODE(1,283,LEVL)LVL
0220      283   FORMAT(I1)
0221            WRITE(I1,284) MOD,CONTIN,JBNM(I2),BATCH(I8),OPTR,
               1                FMAT,LVL
0222      284   FORMAT(///,20X,'NODE=',A6,/,20X,'CONTINUE=',A3,/,20X,
               1     'JOB NAME=',A6,/,20X,'BATCH NO.=',A3,/,20X,
               1     'OPERATOR=',A3,/,20X,'FORMAT=',A9,/,20X,'LEVEL=',I1)
0223            DO 285 IJKL=1,6
0224            READ(UNIT=2,FMT=286,ERR=1030,END=287) LEVEL,LVLFLE
0225      286   FORMAT(I1,A9)
0226            IF (LEVL.EQ.LEVEL) GOTO 310
0227      285   CONTINUE
0228            CLOSE (UNIT=2)
0229      287   CALL CURPOS(0,0)
0230            TYPE*,' '
0231            WRITE(I1,281)
0232      281   FORMAT(25X,'NO MORE LEVELS EXIST',/)
0233      290   READ(I1,289) VAR1
0234      289   FORMAT(A1)
0235            IDN = ICHAR(VAR1)
0236            IF(IDN.EQ.SP) THEN
0237            CALL CURPOS(0,0)
0238            TYPE*,' '
0239            CALL ERALIN
0240            CLOSE(UNIT=2)
0241            GOTO 245
0242            ELSE
0243            GOTO 290
0244            ENDIF
        *
        *******                    DATA CREATION                              **
        ***************************************************************************
        *
        *
        ****       FORMAT CREATION IS DONE UNDER THE CONTROL OF A
        ****       FORMAT CALLED 'FORMATIS'. HENCE THE BELOW CHECK
        ****       IS TO SEE IF FORMAT-CREATION IS BEING DONE.'NRECNO'
        ****       VARIABLE IS INITIALIZED TO 1 AT THE START OF A FRESH
        ****       SESSION.IT IS INCREMENTED BY 1 AT THE END OF A RECORD
        *
0245     315     CLOSE (UNIT=2)
0246     310     IF (FMAT.EQ.'FORMATIS') THEN
        *
        ******* FRMTIS IS A SUBROUTINE FOR 'FORMATIS' FORMAT
        *

0247            CALL FRMTIS
0248             OPEN (UNIT=7,RECL=38,STATUS='OLD',ERR=1070)
0249            REWIND (UNIT=7)
```

```
0250              IF (NRECNO.EQ.1) THEN
                *
                *******   THE RECORD NO.IS WRITTEN IN THE BUFFER SINCE
                *******   FIRST FIVE CHARACTERS AT THE START OF THE A
                *******   RECORD IS THE RECORD NO
                *
0251               DO 410 JJ=1,5
0252                ENCODE(1,400,BUFFER(JJ)) RECNO(JJ)
0253     400        FORMAT(I1)
0254     410       CONTINUE
0255               I=I+5
0256               RCOUNT=RCOUNT+5
0257               BCOUNT=BCOUNT+5
0258              GOTO 450
0259              ELSE
0260               GOTO 450
0261              ENDIF
0262             ENDIF
                *
                ******* FOR DATA CREATION FIRST CONTINUE-OPTION IS CHECKED
                *
0263     415     IF((CONT.EQ.'Y').AND.(NRECNO.EQ.1)) THEN
                *
                *******                TO READ THE LAST RECORD NO.                    **
                *
0264              ENCODE(12,420,FILENM) JBNM(I2),BATCH(I8)
0265     420      FORMAT (A6,A3)
0266              OPEN(UNIT=4,RECL=RECSIZ,FILE=FILENM,STATUS='NEW',
                1         ERR=1040)
                *
                ******   BELOWA FILE IS BROUGHT TO END OF FILE. AND
                ******   THEN BACKSPACED.
                *
0267              ICHECK=0
0268              IF(ICHECK.EQ.0) THEN
0269              READ(UNIT=4,FMT=*,END=422)
0270              ICHECK = 0
0271              ENDIF
0272     422      BACKSPACE 13
                *
                *******  WRITING THE RECORD IN THE BUFFER'DUPBUF'
                *
0273              DO 423 J=1,5
0274              READ(UNIT=4 ,FMT=*,END=425) DUPBUF(J)
0275     423      CONTINUE
                *
                *******     NOW, 'RECNO' IS INITIALIZED TO THE PREVIOUS
                *******     RECNO AND INCREMENTED LATER.
                *
0276     425      DO 430 II=1,5
0277               ENCODE(1,426,RECNO(I)) DUPBUF(II)
0278     426       FORMAT(A1)
0279     430      CONTINUE
                *
                *******         TO POSITION THE FILE AT THE END
```

```
        *******          AND LATER CLOSE IT
        *
0280     437      CLOSE(UNIT=4,DISPOSE='SAVE',ERR=1060)
0281              GOTO 439
        *
        ******* TO INCREMENT RECORD BY ONE
        *
0282     439      CALL ADDRNO(RECNO,5)
        *
        *******  WRITING THE NEW RECORD NO.IN THE FILE
        *
0283              DO 441 KK=1,5
0284               ENCODE(1,440,BUFFER(KK)) RECNO(KK)
0285     440       FORMAT(I1)
0286     441      CONTINUE
0287               I=I+5
0288               RCOUNT= RCOUNT+5
0289               BCOUNT=BCOUNT+5
0290             ELSEIF((CONT.EQ.'N').AND.(NRECNO.EQ.1)) THEN
        *
        ********              WRITE THE RECORD NO.                    **
        *
0291              DO 443 KK=1,5
0292               ENCODE(1,442,BUFFER(KK)) RECNO(KK)
0293     442       FORMAT(I1)
0294     443      CONTINUE
0295              I=I+5
0296              BCOUNT=BCOUNT+5
0297              RCOUNT=RCOUNT+5
0298             ENDIF
        *
        ******* THE FILE CONTAING THE CHOSEN FORMAT LEVEL IS OPENED
        *
0299             OPEN(UNIT=7,RECL=38,STATUS='OLD',
             1       RECORDTYPE='FIXED',ERR=1070)
        *
        *******          READ THE HEADER RECORD OF FORMAT LEVEL FILE    **
        *
0300     445    READ(UNIT=7,FMT=447,ERR=1080,END=450)NUMREC,FNAME,LEVEL,
             1       RSIZE,DATE,COMENT
0301     447    FORMAT(A5,A9,A1,A3,A6,A14)
        *
        ******* FIRST WE CONVERT  RSIZE FROM CHARACTER TO INTEGER TYPE
        *
0302             ENCODE(3,449,RECSIZ) RSIZE
0303     449    FORMAT(A3)
        *
        *******     FROM HERE STARTS THE ACTUAL DATA ENTRY I.E.,KEYING
        *******     OF DATA UNDER A FORMAT                             ***
        *
        *******     THE BELOW CHECK IS TO SEE IF BUFFER IS FULL. IF
        *******     FULL THEN IT IS WRITTEN IN THE APPROPRIATE FILE
        *
0304     450    IF (BCOUNT.LT.512) THEN
0305              GOTO 455
```

```
0306            ELSE
0307             GOTO 777
0308            ENDIF
        *
        *******    THE RECORD CONTAING FIELD DESCRIPTIONS IS READ                ***
        *
0309     455    READ (UNIT=7,FMT=460,ERR=1080,END=465)NUMREC,FLDNAM,
             1        FSIZE,TYPE,MKEY,SVAL,EVAL,VERIF,JUST,ADUP,FILL
0310     460    FORMAT(A5,A12,A3,A1,A1,A6,A6,A1,A1,A1,A1)
0311            WRITE (5,460)NUMREC,FLDNAM,FSIZE,TYPE,MKEY,SVAL,EVAL,
             1            VERIF,JUST,ADUP,FILL
        *
        ******* WE CONVERT FSIZE FROM CHARACTER TO INTEGER TYPE                 *
        *
0312            DECODE(3,462,FSIZE) FLDSIZ
0313     462    FORMAT(I3)
0314            IF (FMAT.EQ.'FORMATIS') GOTO 495
        *
        *******    TO CHECK IF RCOUNT EXCEEDS RSIZE WITH THE ADDITION
        *******    OF THIS FIELD
        *
0315     465    IF ((RCOUNT+FLDSIZ).GT.RECSIZ) THEN
        *
        *******    FILL THE RECORD WITH BLANKS                                  ****
        *
0316             LL=RECSIZ-RCOUNT
0317             DO 470 LLJ=1,LL
0318             BUFFER(RCOUNT+LLJ) = ' '
0319     470     CONTINUE
0320             I=RCOUNT+LL+1
0321             RCOUNT=0
0322             BCOUNT=BCOUNT+LL
        *
        *******    NOW INCREMENT THE RECORD NO.                                 ****
        *
0323             CALL ADDRNO(RECNO,5)
        *
        *******         TO WRITE THE RECORD NO.                                 ****
        *
0324             DO 480 MM=1,5
0325              ENCODE(1,490,BUFFER(MM)) RECNO(MM)
0326     490      FORMAT(1X,I1)
0327     480     CONTINUE
0328             I=I+5
0329             RCOUNT=RCOUNT+5
0330             BCOUNT=BCOUNT+5
0331             NRECNO=NRECNO+1
0332            ENDIF
        *
        *******      DISPLAYING THE FIELD NO.,FIELD NAME, LEVEL
        *******      IN THE STATUS LINE
        *
0333     495    IF (FMAT.EQ.'FORMATIS') THEN
0334            LEVEL = '0'
0335            ENDIF
```

```
0336              CALL CURPOS(0,0)
0337       500    WRITE(I1,510) FLDNO,FLDNAM,LEVEL
0338       510    FORMAT(1X,'FIELD NO=',I3,30X,A12,15X,'LEVEL=',A1)
                  *
                  *******         SKIP ONE LINE  FOR ERROR MESSAGES                    ***
                  *
0339              TYPE*,' '
                  *
                  *******        AUTO DUP IMPLEMENTATION                               **
                  *
0340              IF (ADUP.EQ.'N') THEN
0341               GOTO 580
0342              ELSEIF((ADUP.EQ.'Y').AND.(NRECNO.EQ.1).AND.
                 1            (CONT.EQ.'Y')) THEN
                  *
                  *******        WRITING THE ALREADY OPENED FILE IN DUPBUF-BUFFER       ***
                  *
0343               READ(UNIT=4,FMT=*,END=532)(DUPBUF(J),J=1,512)
0344       532     ENCODE(12,531,FILENM) JBNM(12),BATCH(18)
0345       531     FORMAT(A6,A3)
0346               J=1
0347               DO 540 IJ=1,FLDSIZ
0348                ENCODE(1,533,BUFFER(I)) DUPBUF(J)
0349       533      FORMAT(A1)
0350               I=I+1
0351               J=J+1
0352       540     CONTINUE
0353       545     FCOUNT =FLDSIZ
0354               GOTO 760
0355              ELSEIF((ADUP.EQ.'Y').AND.(NRECNO.GT.1).AND.
                 1          (RECSIZ.LE.256)) THEN
0356               DO 550 NI=1,FLDSIZ
0357               JI=BCOUNT+1
0358               JIK=JI-RECSIZ
0359               ENCODE(1,546,BUFFER(JI)) BUFFER(JIK)
0360       546     FORMAT(A1)
0361               BCOUNT=BCOUNT+1
0362       550     CONTINUE
0363               I=I+FLDSIZ
0364               BCOUNT=BCOUNT-FLDSIZ
0365               FCOUNT= FLDSIZ
0366               GOTO 760
0367              ELSEIF((ADUP.EQ.'Y').AND.(NRECNO.GT.1).AND.
                 1            (RECSIZ.EQ.512)) THEN
0368               FCOUNT=FLDSIZ
0369               GOTO 760
0370              ELSEIF((ADUP.EQ.'Y').AND.(NRECNO.EQ.1).AND.
                 1            (CONT.EQ.'N')) THEN
0371               GOTO 572
0372              ELSE
0373       572     CALL CURPOS(0,0)
0374               TYPE*,' '
0375               WRITE(I1,573)
0376       573     FORMAT(25X,'AUTO DUP ERROR',$)
0377       575     READ(5,289) VAR1
```

```
0378              IDN=ICHAR(VAR1)
0379              IF(IDN.EQ.SP) THEN
0380              CALL CURPOS(0,0)
0381              TYPE*,' '
0382               CALL ERALIN
0383               GOTO 580
0384              ELSE
0385               GOTO 575
0386              ENDIF
0387             ENDIF
        *
        *******             DATA IS ACCEPTED HERE
        *
0388       580    CALL CURPOS(0,0)
0389              CALL CURPOS(8,0)
0390              WRITE (I1,583)
0391       583    FORMAT(1X,'INPUT=',$)
0392              READ(I1,585) FIELD
0393       585    FORMAT(A)
0394              DECODE (80,587,FIELD)CHAREC
0395       587    FORMAT(80A1)
0396              K=0
0397              DO 589 J=I,80
0398              IASC=ICHAR(CHAREC(K+1))
0399              IF((IASC.EQ.SP).OR.(IASC.EQ.013)) GOTO 591
0400              K=K+1
0401       589    CONTINUE
0402             DO 581 J=I,I+K
0403             BUFFER(J)=CHAREC(L)
0404             L=L+1
0405       581   CONTINUE
        *
        ******** TO NOTE DOWN THE FORMAT NAME & LEVEL INTO
        ******** TWO VARIABLES - FOMAT & LVEL
        *
0406             IF (FMAT.EQ.'FORMATIS') THEN
0407             IF (FLDNO.EQ.2) THEN
0408              ENCODE(9,603,FOMAT)FIELD
0409       603   FORMAT(A9)
0410             IF (FLDNO.EQ.3) THEN
0411              ENCODE(1,604,LVEL)FIELD
0412       604   FORMAT(A1)
0413             ENDIF
0414             ENDIF
0415             ENDIF
0416       591       IASC=ICHAR(FIELD)
        *
        *******  DECIMAL 26 IS ^Z
        *
        *******  TO CHECK IF ^Z IS DEPRESSED
        *
0417             IF (IASC.EQ.26) THEN
        *
        *******  1300 IS 'STOP' STATEMENT
        *
```

```
0418              GOTO 1300
0419               ENDIF
0420                FCOUNT=K
0421              I=I+k
                *
                ************             TYPE CHECKING                                  *****
                *
0422              N=I-K
0423              DO 611 L=N,N+K
0424              IASC=ICHAR(BUFFER(L))
0425              IF ((TYPE.EQ.'N').AND.((IASC.GE.'060'O).AND.
                1         (IASC.LE.'071'O))) THEN
0426              GOTO 611
0427              ELSEIF (((TYPE.EQ.'A').AND.(IASC.GE.'101'O).AND.
                1           (IASC.LE.'132'O)).OR.((TYPE.EQ.'A').AND.
                1          (IASC.GE.'141'O).AND.(IASC.LE.'172'O))) THEN
0428              GOTO 611
0429              ELSEIF (TYPE.EQ.'U') THEN
0430              GOTO 611
0431              ELSE
0432       592    CALL CURPOS(0,0)
0433              TYPE*,' '
0434              WRITE(5,590)
0435       590    FORMAT(25X,'INCORRECT DATA TYPE',/)
0436              FCOUNT=0
0437              I=I-K
0438       610    READ(5,289)VAR1
0439              IDN=ICHAR(VAR1)
0440              IF (IDN.EQ.SP) THEN
0441              CALL CURPOS(0,0)
0442              TYPE*,' '
0443              CALL ERALIN
0444              GOTO 580
0445              ELSE
0446              GOTO 610
0447              ENDIF
0448              ENDIF
0449       611    CONTINUE
                *
                ***************  LIST CHECKING FOR FORMAT CREATION                  ******
                *
0450       612    IF (FMAT.EQ.'FORMATIS') THEN
0451               N=I-K
0452              DO 665 M=N,N+K
0453              IF (FLDNO.EQ.10) THEN
                *
                **************   FIELD NO. IS TYPE                                ***********
                *
0454              IF((BUFFER(M).EQ.'A').OR.(BUFFER(N).EQ.'N').OR.
                1    (BUFFER(M).EQ.'U')) THEN
0455              GOTO 690
0456              ELSE
0457              CALL CURPOS(0,0)
0458              TYPE*,' '
0459              WRITE (I1,615)
```

```
0460      615     FORMAT(25X,'ONLY ''A'',''N'',''U'' ALLOWED',$)
0461              FCOUNT=0
0462              I=I-K
0463      620     READ(5,289)VAR1
0464              IDN=ICHAR(VAR1)
0465              IF(IDN.EQ.SP) THEN
0466              CALL CURPOS(0,0)
0467              TYPE*,' '
0468              CALL ERALIN
0469              GOTO 580
0470              ELSE
0471              GOTO 620
0472              ENDIF
0473              ENDIF
          *
          ******** FIELD NO. 11 IS MUST KEY, FIELD NO 14 IS
          ******** VERIFICATION AND FIELD NO.16 IS AUTO DUP
          *
0474              ELSEIF ((FLDNO.EQ.11).OR.(FLDNO.EQ.14).OR.
         1                (FLDNO.EQ.16)) THEN
0475              IF((BUFFER(M).EQ.'Y').OR.(BUFFER(M).EQ.'N')
         1     .OR.(IASC.EQ.SP)) THEN
0476              GOTO 690
0477              ELSE
0478              CALL CURPOS(0,0)
0479              TYPE*,' '
0480              WRITE(I1,625)
0481      625     FORMAT(25X,'ONLY ''Y''/''N''  ALLOWED',$)
0482              FCOUNT=0
0483              I=I-K
0484      630     READ (5,289)VAR1
0485              IDN=ICHAR(VAR1)
0486              IF (IDN.EQ.SP) THEN
0487              CALL CURPOS(0,0)
0488              TYPE*,' '
0489              CALL ERALIN
0490              GOTO 580
0491              ELSE
0492              GOTO 630
0493              ENDIF
0494              ENDIF
0495              ELSEIF (FLDNO.EQ.15) THEN
0496              IF ((BUFFER(M).EQ.'R').OR.(BUFFER(M).EQ.'M')
         1      .OR.(IASC.EQ.SP)) THEN
0497              GOTO 690
0498              ELSE
0499              CALL CURPOS(0,0)
0500              TYPE*,' '
0501              WRITE(I1,635)
0502      635     FORMAT(25X,'ONLY ''R'',''L'' ALLOWED',$)
0503              FCOUNT=0
0504              I=I-K
0505      640     READ (5,289)VAR1
0506              IDN=ICHAR(VAR1)
0507              IF(IDN.EQ.SP) THEN
```

```
0508            CALL CURPOS(0,0)
0509            TYPE*, ' '
0510            CALL ERALIN
0511            GOTO 580
0512            ELSE
0513            GOTO 640
0514           ENDIF
0515            ENDIF
      *
      **********   FIELD NO. 17 IS FILL CHARACTER                    **********
      *
0516            ELSEIF(FLDNO.EQ.17) THEN
0517            IF((BUFFER(M).EQ.'B').OR.(BUFFER(M).EQ.'Z')
     1       .OR.(IASC.EQ.SP)) THEN
0518            GOTO 690
0519            ELSE
0520            CALL CURPOS(0,0)
0521            TYPE*,' '
0522            WRITE(I1,660)
0523     660    FORMAT(25X,'ONLY ''B'',''Z'' ',$)
0524     650    READ (I1,289) VAR1
0525            IDN=ICHAR(VAR1)
0526            IF(IDN.EQ.SP) THEN
0527            CALL CURPOS(0,0)
0528            TYPE*,' '
0529            CALL ERALIN
0530            FCOUNT=0
0531            I=I-K
0532            GOTO 580
0533            ELSE
0534            GOTO 650
0535            ENDIF
0536            ENDIF
0537            ELSE
0538            GOTO 690
0539            ENDIF
0540     665    CONTINUE
0541            ENDIF
      *
      ************  CHARACTER RANGE CHECKING                          **********
      *
0542            J=I-K
0543            DO 685 M=J,J+K
0544            IASC=ICHAR(BUFFER(M))
0545     670    IASC1=ICHAR(SVAL)
0546            IASC2=ICHAR(EVAL)
0547            IF((TYPE.EQ.'A').AND.(IASC1.NE.032)) THEN
0548             IF((IASC1.LE.IASC).AND.(IASC.LE.IASC2)) THEN
0549              GOTO 690
0550             ELSE
0551              CALL CURPOS(0,0)
0552              TYPE*,' '
0553              WRITE(I1,675)
0554     675      FORMAT(25X,'DATA OUT OF RANGE',$)
0555              FCOUNT=0
```

```
0556                I=I-K
0557      680       READ(5,289)VAR1
0558                IDN=ICHAR(VAR1)
0559                IF(IDN.EQ.SP) THEN
0560                CALL CURPOS(0,0)
0561                TYPE*,' '
0562                CALL ERALIN
0563                GOTO 580
0564               ELSE
0565                GOTO 680
0566               ENDIF
0567              ENDIF
0568              ENDIF
0569      685     CONTINUE
          *
          ***********  DEFAULT CASES IN FORMAT CREATION                ****************
          *
0570      690     IASC=ICHAR(FIELD)
0571              IF(IASC.EQ.SP) THEN
0572              IF (FMAT.EQ.'FORMATIS') THEN
0573               IF((FLDNO.EQ.11).OR.(FLDNO.EQ.16)) THEN
0574                 VAR2='N'
0575               ELSEIF(FLDNO.EQ.14) THEN
0576                VAR2='Y'
0577               ELSEIF(FLDNO.EQ.17) THEN
          *
          ********          FIELD NO. 17 IS FILL CHARACTER                 ********
          *
0578                IF (TYPE.EQ.'N') THEN
0579                 VAR2='Z'
0580                ELSE
0581                 VAR2='B'
0582                ENDIF
0583               ELSEIF(FLDNO.EQ.15) THEN
0584                IF(TYPE.EQ.'N') THEN
0585                 VAR2='R'
0586                ELSE
0587                 VAR2='L'
0588                ENDIF
0589               ELSE
0590               GOTO 700
0591               ENDIF
0592               ELSE
0593               GOTO 700
0594               ENDIF
          *
          ***********            MUST-KEY VALIDATION                  *********
          *
0595      700     IF((MKEY.EQ.'Y').AND.(FCOUNT.EQ.0)) THEN
0596                CALL CURPOS(0,0)
0597                TYPE*,' '
0598                WRITE(11,705)
0599      705       FORMAT(25X,'MUST KEY FIELD',8)
0600      710       READ(11,289) VAR1
0601                IDN=ICHAR(VAR1)
```

```
0602              IF(IDN.EQ.SP) THEN
0603              CALL CURPOS(0,0)
0604              TYPE*,' '
0605              CALL ERALIN
0606              GOTO 580
0607             ELSE
0608              GOTO 710
0609             ENDIF
0610             ELSEIF((MKEY.EQ.'Y').AND.(FCOUNT.GE.1).AND.
                1           (FCOUNT.LT.FLDSIZ))  THEN
0611             GOTO 730
0612              ELSEIF((MKEY.EQ.'Y').AND.(FCOUNT.EQ.0)) THEN
0613               GOTO 730
0614              ENDIF
0615      720     IF ((FCOUNT.GT.1).AND.(FCOUNT.LT.FLDSIZ)) THEN
0616              GOTO 730
0617              ELSE
0618               GOTO 760
0619              ENDIF
0620             ELSE
0621              GOTO 760
0622             ENDIF
                *
                *******   TO FILL THE REST OF THE NON KETED CHARACTERS                ****
                *******   WITH FILL CHARACTER
                *
0623      730     IF(JUST.EQ.'L') THEN
0624              IF(FILL.EQ.'B') THEN
0625               BCOUNT=BCOUNT+FCOUNT
0626               DO 735 IJN=1,FLDSIZ-FCOUNT
0627                BUFFER(BCOUNT+IJN)=' '
0628      735       CONTINUE
0629              BCOUNT=BCOUNT-FLDSIZ
0630             ELSE
0631              BCOUNT=BCOUNT+FLDSIZ
0632              DO 740 IJN=1,FLDSIZ-FCOUNT
0633               BUFFER(BCOUNT+IJN)='0'
0634      740     CONTINUE
0635             BCOUNT=BCOUNT-FLDSIZ
0636             FCOUNT=FLDSIZ
0637             GOTO 760
0638             ENDIF
0639             ELSEIF (JUST.EQ.'R') THEN
0640              JUST1=BCOUNT
0641              JUST2=FLDSIZ-FCOUNT
0642              JUST3=1
0643              DO 745 KI=1,FCOUNT
0644               ENCODE(1,533,BUFFER(JUST1+JUST2+1))
                1             BUFFER(JUST1+JUST3)
0645               JUST2=JUST2+1
0646               JUST3=JUST3+1
0647      745     CONTINUE
                *
                *********       WRITING ZEROS AND BLANKS                          **********
                *
```

```
0648              JUST2=FLDSIZ-FCOUNT
0649              IF(FILL.EQ.'B') THEN
0650               DO 747 KL=1,JUST2
0651                BUFFER(JUST1+1)=' '
0652                JUST1=JUST1+1
0653        747     CONTINUE
0654               FCOUNT=FLDSIZ
0655               GOTO 760
0656              ELSE
0657               JUST1=BCOUNT
0658              DO 750 JUST1=JUST1,JUST2
0659               BUFFER(JUST1+1)='0'
0660        750   CONTINUE
0661              FCOUNT=FLDSIZ
0662              GOTO 760
0663              ENDIF
0664              ELSE
0665               GOTO 760
0666              ENDIF
          *
          *
0667        760  IF(FCOUNT.EQ.FLDSIZ) THEN
0668              FLDNO=FLDNO+1
0669              BCOUNT=BCOUNT+FCOUNT
0670              RCOUNT=RCOUNT+FCOUNT
0671              FCOUNT=0
0672              IF(RCOUNT.EQ.RECSIZ) THEN
0673              CALL ADDRNO(RECNO,5)
0674              I=I+5
0675              RCOUNT=RCOUNT+5
0676              BCOUNT=BCOUNT+5
0677              NRECNO=NRECNO+1
0678              I=BCOUNT
0679              ENDIF
0680              ENDIF
0681               GOTO 450
0682        777   IF (FMAT.EQ.'FORMATIS') GOTO 1260
0683               IF(CONT.EQ.'Y') THEN
0684                ENCODE(12,780,FILENM) JBNM(12),BATCH(18)
0685        780     FORMAT(2A6)
0686                OPEN(UNIT=4,ACCESS='APPEND',FILE=FILENM,STATUS='OLD',
                1     RECL=RECSIZ,ERR=1100)
0687                WRITE(UNIT=4,FMT=785) BUFFER
0688        785     FORMAT(1X,A)
0689                CLOSE(UNIT=4,DISPOSE='SAVE',ERR= 1110)
0690                ELSE
0691                OPEN(UNIT=4,STATUS='NEW',
                1     RECL=RECSIZ,ERR=1120)
0692                WRITE(UNIT=4,FMT=800)BUFFER
0693        800     FORMAT(1X,A)
0694                CLOSE(UNIT=4,DISPOSE='SAVE',ERR=1130)
0695                ENDIF
0696                GOTO 1300
0697        1260  IF (FMAT.EQ.'FORMATIS') THEN
0698              ENCODE (13,1265,FRMAT1) FOMAT,DOTFMT
```

```
0699     1265   FORMAT(A9,A4)
       *
       ********* APPENDING 'FORMAT.HDR' FILE WITH NEW FORMAT
       *
0700           OPEN (UNIT=1,ACCESS='APPEND',RECL=9,
      1          STATUS='OLD',ERR=1000)
0701             WRITE (1,1250) FOMAT
0702     1250  FORMAT (A9)
       *
       ******** LEVEL FILE IS ALSO APPENDED
       *
0703           OPEN (UNIT = 2,ACCESS='APPEND',FILE = FRMAT1,
      1          RECL=11,STATUS='OLD',ERR=1020)
0704           WRITE (2,1270) LVEL,FOMAT
0705     1270  FORMAT (1X,A1,A9)
0706           CLOSE (UNIT = 2)
0707           CLOSE (UNIT= 1,DISPOSE='SAVE')
       *
       ********* TO WRITE INTO FORMAT FILE - UNIT = 7
       *
0708            OPEN (UNIT=3,RECL=RECSIZ,STATUS='NEW',ERR=1070)
0709           WRITE (UNIT = 3,FMT = 1280,ERR = 1090 ,END = 1290)
      1           BUFFER
0710     1280  FORMAT(1X,A)
0711            CLOSE(UNIT=3)
0712           ENDIF
0713           GOTO 580
       **
       *
       ******* ERROR MESSAGES BLOCK ( I/O ERRORS ONLY )                    **
       *
       *
0714     1000 CALL CURPOS(0,0)
0715          TYPE*,' '
0716          WRITE(I1,1005)
0717     1005 FORMAT(25X,'ERROR IN OPENING ''FORMAT.HDR'' FILE',/)
0718          GOTO 1200
0719     1010 CALL CURPOS(0,0)
0720          TYPE*,' '
0721          WRITE(I1,1015)
0722     1015 FORMAT(25X,'ERROR IN READING ''FORMAT.HDR''  FILE',/)
0723          GOTO 1200
0724     1020 CALL CURPOS(0,0)
0725          TYPE*,' '
0726          WRITE(I1,1025)
0727     1025 FORMAT(25X,'ERROR IN OPENING ''FMAT.FMT''
      1          FORMAT FILE',/)
0728          GOTO 1200
0729     1030 CALL CURPOS(0,0)
0730          TYPE*,' '
0731          WRITE(I1,1035)
0732     1035 FORMAT(25X,'ERROR IN READING ''FMAT.FMT'' FORMAT
      1       FILE',/)
0733          GOTO 1200
0734     1040 CALL CURPOS(0,0)
```

```
0735          TYPE*,' '
0736          WRITE(I1,1045)
0737      1045 FORMAT(25X,'ERROR IN OPENING OLD DATA FILE FOR
             1     READING THE LAST RECORD',$)
0738          GOTO 1200
0739      1050 CALL CURPOS(0,0)
0740          TYPE*,' '
0741          WRITE(I1,1055)
0742      1055 FORMAT(25X,'ERROR IN POSITIONING THE DATA
             1        FILE AT THE END',$)
0743          GOTO 1200
0744      1060 CALL CURPOS(0,0)
0745          TYPE*,' '
0746          WRITE(I1,1065)
0747      1065 FORMAT(25X,'ERROR IN CLOSING THE OLD DATA FILE',$)
0748          GOTO 1200
0749      1070 CALL CURPOS(0,0)
0750          TYPE*,' '
0751          WRITE(I1,1075)
0752      1075 FORMAT(25X,'ERROR IN OPENING ''LVLFLE.LVL'' FORMAT
             1        LEVEL FILE',$)
0753          GOTO 1200
0754      1080 CALL CURPOS(0,0)
0755          TYPE*,' '
0756          WRITE(I1,1085)
0757      1085 FORMAT(25X,'ERROR  IN READING ''LVLFLE.LVL'' FORMAT
             1        LEVEL FILE',$)
0758          GOTO 1200
0759      1090 CALL CURPOS(0,0)
0760          TYPE*,' '
0761          WRITE (I1,1095)
0762      1095 FORMAT(25X,'ERROR IN WRITING  ''LVLFLE  FILE ''     ')
0763          GOTO 1200
0764      1100 CALL CURPOS(0,0)
0765          TYPE*,' '
0766          WRITE(I1,1105)
0767      1105 FORMAT(25X,'ERROR IN OPENING ''FILENM.DAT'' FILE FOR
             1        WRITING BUFFER CONTENTS',$)
0768          GOTO 1200
0769      1110 CALL CURPOS(0,0)
0770          TYPE*,' '
0771          WRITE(I1,1115)
0772      1115 FORMAT(25X,'ERROR IN CLOSING OLD ''FILENM.DAT''FILE
             1        OPENED FOR WRITING BUFFER CONTENTS',$)
0773          GOTO 1200
0774      1120 CALL CURPOS(0,0)
0775          TYPE*,' '
0776          WRITE(I1,1125)
0777      1125 FORMAT(25X,'ERROR IN NEW ''FILENM.DAT'' FILE FOR
             1       WRITING THE CONTENTS OF THE BUFFER',$)
0778          GOTO 1200
0779      1130 CALL CURPOS(0,0)
0780          TYPE*,' '
0781          WRITE(I1,1135)
0782      1135 FORMAT(25X,'ERROR IN CLOSING ''FILENM.DAT'' FILE
```

```
     1        OPENED FOR WRITING THE CONTENTS OF THE BUFFER',/)
0783      1200 READ(5,*)VAR1
0784             IDN=ICHAR(VAR1)
0785             IF(IDN.EQ.SP) THEN
0786             CALL CURPOS(0,0)
0787             TYPE*,' '
0788             CALL ERALIN
0789             ELSE
0790             GOTO 1200
0791             ENDIF
0792      1290   CLOSE (UNIT=7,DISPOSE='SAVE')
0793      1300   STOP
0794      1400   END
```

```
        *
        *
        *
        ********************************************************************************
        ***********          SUBROUTINES                                        ********
        *
        ***********          CURPOS POSITIONING                                 *********
        *
0001            SUBROUTINE CURPOS(N1,N2)
0002            CHARACTER*1 ESC,V1,N1*2,V2,N2*2,V3
0003            DATA ESC,V1,V2,V3/'033'O,'[',';','H'/
0004            TYPE*,ESC,V1,N1,V2,N2,V3
0005            RETURN
0006            END
```

```
        *
        ***********          CLEARING SCREEN                    ********
        *
0001            SUBROUTINE CLRSRN
0002            CHARACTER*1 ESC,V1,N1,V2,V3
0003            DATA ESC,V1,N1,V2,V3/'033'O,'[','2','J','H'/
0004            TYPE*,ESC,V1,N1,V2
0005            TYPE*,ESC,V1,V3
0006            RETURN
0007            END
```

```
        *
        **********              ERASING LINE                              **********
        *
0001            SUBROUTINE ERALIN
0002            CHARACTER*1 ESC,V1,N1,V2
0003            DATA ESC,V1,N1,V2/'033'O,'[','2','K'/
0004            TYPE*,ESC,V1,N1,V2
0005            RETURN
0006            END
```

```
      *
      **********   INCREMENTING THE RECORD NO.-'RECNO'              **********
      *
0001          SUBROUTINE ADDRNO(A,N)
0002          DIMENSION A(N)
0003          INTEGER N,A
0004          IF (A(5).EQ.9) THEN
0005            A(5)=0
0006          ELSE
0007            A(5)=A(5)+1
0008          ENDIF
0009          IF(A(4).EQ.9) THEN
0010             A(4)=0
0011          ELSE
0012            A(4)=A(4)+1
0013          ENDIF
0014          IF(A(3).EQ.9) THEN
0015              A(3)=0
0016          ELSE
0017             A(3)=A(3)+1
0018          ENDIF
0019          IF(A(2).EQ.99) THEN
0020           A(2)=0
0021          ELSE
0022              A(2) =A(2) +1
0023          ENDIF
0024          IF(A(1).EQ.9) THEN
0025                A(1)=0
0026          ELSE
0027                A(1)=A(1)+1
0028          ENDIF
0029          RETURN
0030          END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000270	92	RW,I,CON,LCL
3	$IDATA	000020	8	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TY
ADDRNO		1-000000								

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME
N	I*2	F-000004*							

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
A	I*2	F-000002*	**	(*)

TOTAL SPACE ALLOCATED = 000310 100

NO FPP INSTRUCTIONS GENERATED

```
      *
      ******        SUBROUTINE 'FRMTIS'
      *
0001        SUBROUTINE FRMTIS
0002        CHARACTER*1 DUP,EVL*6,FLDNM*12,FSIZ*3,FIL,JUS,MKY
0003        CHARACTER*1 NOREC*5,SVL*6,TYP,VERF,FORMA*8,LVLFLE*10
      *
0004        DIMENSION NOREC(17)
0005        DIMENSION FLDNM(17)
0006        DIMENSION FSIZ(17)
0007        DIMENSION TYP(17)
0008        DIMENSION MKY(17)
0009        DIMENSION SVL(17)
0010        DIMENSION EVL(17)
0011        DIMENSION VERF(17)
0012        DIMENSION JUS(17)
0013        DIMENSION DUP(17)
0014        DIMENSION FIL(17)

      *
0015        FORMA = 'FORMATIS'
      *
0016        NOREC(1) = '00001'
0017        FLDNM(1) = 'RECORD NO.   '
0018        FSIZ(1)  = '005'
0019        TYP(1)   = 'N'
0020        MKY(1)   = 'N'
0021        SVL(1) = '000000'
0022        EVL(1) = '999999'
0023        VERF(1) = 'Y'
0024        JUS(1)   = 'R'
0025        DUP(1) = 'N'
0026        FIL(1) = 'Z'
      *
0027        NOREC(2) = '00002'
0028        FLDNM(2) = 'FORMAT NAME '
0029        FSIZ(2)    = '009'
0030        TYP(2)     = 'U'
0031        MKY(2)     = 'Y'
0032        SVL(2)   = '       '
0033        EVL(2)     = '       '
0034        VERF(2)    = 'Y'
0035        JUS(2)     = 'L'
0036        DUP(2)     = 'N'
0037        FIL(2)     = 'B'
      *
0038        NOREC(3) = '00003'
0039        FLDNM(3) = 'LEVEL        '
0040        FSIZ(3)  = '001'
0041        TYP(3)   = 'N'
0042        MKY(3)   = 'Y'
0043        SVL(3)   = '000001'
0044        EVL(3)   = '000006'
0045        VERF(3)  = 'Y'
0046        JUS(3)   = 'R'
```

```
0047            DUP(3)    = 'N'
0048            FIL(3)    = 'Z'
      *
0049            NOREC(4)  = '00004'
0050            FLDNM(4)  = 'RECORD SIZE '
0051            FSIZ(4)   = '003'
0052            TYP(4)    = 'N'
0053            MKY(4)    = 'Y'
0054            SVL(4)    = '000000'
0055            EVL(4)    = '000999'
0056            VERF(4)   = 'Y'
0057            JUS(4)    = 'R'
0058            DUP(4)    = 'N'
0059            FIL(4)    = 'Z'
      *
0060            NOREC(5)  = '00005'
0061            FLDNM(5)  = 'DATE        '
0062            FSIZ(5)   = '006'
0063            TYP(5)    = 'N'
0064            MKY(5)    = 'Y'
0065            SVL(5)    = '      '
0066            EVL(5)    = '      '
0067            VERF(5)   = 'Y'
0068            JUS(5)    = 'R'
0069            DUP(5)    = 'N'
0070            FIL(5)    = 'Z'
      *
0071            NOREC(6)  = '00006'
0072            FLDNM(6)  = 'COMENT      '
0073            FSIZ(6)   = '014'
0074            TYP(6)    = 'U'
0075            MKY(6)    = 'N'
0076            SVL(6)    = '      '
0077            EVL(6)    = '      '
0078            VERF(6)   = 'Y'
0079            JUS(6)    = 'L'
0080            DUP(6)    = 'N'
0081            FIL(6)    = 'Z'
      *
0082            NOREC(7)  = '00007'
0083            FLDNM(7)  = 'RECORD NO   '
0084            FSIZ(7)   = '005'
0085            MKY(7)    = 'Y'
0086            TYP(7)    = 'N'
0087            SVL(7)    = '000000'
0088            EVL(7)    = '999999'
0089            VERF(7)   = 'Y'
0090            JUS(7)    = 'R'
0091            DUP(7)    = 'N'
0092            FIL(7)    = 'Z'
      *
0093            NOREC(8)  = '00008'
0094            FLDNM(8)  = 'FIELD NAME  '
0095            FSIZ(8)   = '012'
0096            TYP(8)    = 'U'
```

```
0097            MKY(8)    = 'Y'
0098            SVL(8)    = '      '
0099            EVL(8)    = '      '
0100            VERF(8)   = 'Y'
0101            JUS(8)    = 'L'
0102            DUP(8)    = 'N'
0103            FIL(8)    = 'B'
        *
0104            NOREC(9)  = '00009'
0105            FLDNM(9)  = 'FIELD SIZE  '
0106            FSIZ(9)   = '003'
0107            TYP(9)    = 'N'
0108            MKY(9)    = 'Y'
0109            SVL(9)    = '000000'
0110            EVL(9)    = '000999'
0111            VERF(9)   = 'Y'
0112            JUS(9)    = 'R'
0113            DUP(9)    = 'N'
0114            FIL(9)    = 'Z'
        *
0115            NOREC(10)= '00010'
0116            FLDNM(10)= 'TYPE        '
0117            FSIZ(10)  = '001'
0118            TYP(10)   = 'A'
0119            MKY(10)   = 'Y'
0120            SVL(10)   = '      '
0121            EVL(10)   = '      '
0122            VERF(10)  = 'Y'
0123            JUS(10)   = 'R'
0124            DUP(10)   = 'N'
0125            FIL(10)   = 'A'
        *
0126            NOREC(11)= '00011'
0127            FLDNM(11)= 'MUST KEY    '
0128            FSIZ(11)  = '001'
0129            TYP(11)   = 'A'
0130            MKY(11)   = 'Y'
0131            SVL(11)   = '      '
0132            EVL(11)   = '      '
0133            VERF(11)  = 'Y'
0134            JUS(11)   = 'R'
0135            DUP(11)   = 'N'
0136            FIL(11)   = 'B'
        *
0137            NOREC(12)= '00012'
0138            FLDNM(12)= 'START VALUE '
0139            FSIZ(12)  = '006'
0140            TYP(12)   = 'U'
0141            MKY(12)   = 'Y'
0142            SVL(12)   = '      '
0143            EVL(12)   = '      '
0144            VERF(12)  = 'Y'
0145            JUS(12)   = 'L'
0146            DUP(12)   = 'N'
0147            FIL(12)   = 'B'
```

```
      *
0148          NOREC(13)= '00013'
0149          FLDNM(13)= 'END VALUE    '
0150          FSIZ(13) = '006'
0151          TYP(13)  = 'N'
0152          MKY(13)  = 'Y'
0153          SVL(13)  = '      '
0154          EVL(13)  = '      '
0155          VERF(13) = 'Y'
0156          JUS(13)  = 'L'
0157          DUP(13)  = 'N'
0158          FIL(13)  = 'B'
      *
0159          NOREC(14)= '00014'
0160          FLDNM(14)= 'VERIFICATION'
0161          FSIZ(14) = '001'
0162          TYP(14)  = 'A'
0163          MKY(14) = 'Y'
0164          SVL(14) = '      '
0165          EVL(14) = '      '
0166          VERF(14)= 'Y'
0167          JUS(14) = 'L'
0168          DUP(14) = 'N'
0169          FIL(14) = 'B'
      *
0170          NOREC(15)= '00015'
0171          FLDNM(15)= 'JUSTIFICATIN'
0172          FSIZ(15) = '001'
0173          TYP(15)  = 'A'
0174          MKY(15)  = 'Y'
0175          SVL(15)  = '      '
0176          EVL(15)  = '      '
0177          VERF(15) = 'Y'
0178          JUS(15)  = 'L'
0179          DUP(15)  = 'N'
0180          FIL(15)  = 'B'
      *
0181          NOREC(16)= '00015'
0182          FLDNM(16)= 'AUTO DUP     '
0183          FSIZ(16) = '001'
0184          TYP(16)  = 'A'
0185          MKY(16)  = 'Y'
0186          SVL(16)  = '      '
0187          EVL(16)  = '      '
0188          VERF(16) = 'Y'
0189          JUS(16)  = 'R'
0190          DUP(16)  = 'N'
0191          FIL(16)  = 'B'
      *
0192          NOREC(17)= '00017'
0193          FLDNM(17)= 'FILL CHAR.   '
0194          FSIZ(17) = '001'
0195          TYP(17)  = 'A'
0196          MKY(17)  = 'Y'
0197          SVL(17)  = '      '
```

```
0198            EVL(17)  = '      '
0199            VERF(17) = 'Y'
0200            JUS(17) = 'R'
0201            DUP(17) = 'N'
0202            FIL(17) = 'B'
        *
0203            ENCODE (10,15,LVLFLE) FORMA
0204         15 FORMAT (A10)
        *
0205            OPEN (UNIT=7,RECL=38,STATUS='NEW')
0206            DO 50 I=1,17
0207             WRITE(UNIT=7,FMT=40,END=90) MOREC(I),FLDNM(I),
               1      FSIZ(I),TYP(I),MKY(I),SVL(I),EVL(I),VERF(I),
               1           JUS(I),DUP(I),FIL(I)
0208         40 FORMAT(A5,A12,A3,A1,A1,A6,A6,A1,A1,A1,A1)
0209         50 CONTINUE
0210            CLOSE (UNIT=7,DISPOSE='SAVE')
0211         90 RETURN
0212             END
```

```
      *
      ********************************************************************************
      ******************     END   OF   D E S      ************************
      *
```

A P P E N D I X - B

THE OUTPUT LISTING - 1

00000	2M10001A	JAY K V SATAPATHY	UMA CHARAN SATAPATHY	11640285M11A	E O	PHYCHEM BIO15
00000	2M10002A	SIT KUMAR NANDA	NABA KUMAR NANDA	11637785M11A	E O	PHYCHEM BIO15
00000	2M10003A	SIM KUMAR DAS	BRAJA BEHARI DAS	11635185M11A	E O	PHYCHEM BIO11
00000	2M10004A	JIT KUMAR DAS	JADUNATH DAS	11635385M11A	E O	PHYCHEM 11