NASA CR-
141595

AEROPHYSICS RESEARCH CORPORATION
TECHNICAL NOTE

JTN-09
VOLUME II

GTM: GEOMETRY TECHNOLOGY MODULE
VOLUME II - PROGRAMMERS' MANUAL

(NASA-CR-141595) GEOMETRY TECHNOLOGY MODULE (GTM). VOLUME 2: PROGRAMMERS' MANUAL Final Report, Jun. 1973 - Dec. 1974 (Aerophysics Research Corp., Houston Tex.) 123 p HC CSCL 09B
N75-17121
Unclas
G3/61 09642

By: S. J. Reiners, G. N. Hirsch
G. E. Alford and C. R. Glatt

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Spacecraft Center
Houston, Texas 77058

December 1974

122

NOTICE

AEROPHYSICS RESEARCH CORPORATION
TECHNICAL NOTE

JTN-09
VOLUME II

GTM: GEOMETRY TECHNOLOGY MODULE
VOLUME II - PROGRAMMERS' MANUAL

By: S. J. Reiners, G. N. Hirsch, W. N. Colquitt,
G. E. Alford and C. R. Glatt

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Spacecraft Center
Houston, Texas 77058

December 1974

1. Report No. NASA CR-	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle GTM: GEOMETRY TECHNOLOGY MODULE VOLUME II - PROGRAMMERS' MANUAL		5. Report Date December 1974
		6. Performing Organization Code
7. Author(s) S. J. Reiners, G. N. Hirsch, W. N. Colquitt G. E. Alford and C. R. Glatt		8. Performing Organization Report No. JTN-09.- VOLUME II
9. Performing Organization Name and Address Aerophysics Research Corporation Houston, Texas 77058		10. Work Unit No.
		11. Contract or Grant No. NAS9-13584
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Johnson Space Center Engineering Analysis Division,		13. Type of Report and Period Covered Final Report 6-73 thru 12-74
		14. Sponsoring Agency Code EX4

15. Supplementary Notes
Final Report

16. Abstract

The Geometry Technology Module (GTM) is a system of computerized elements residing in the EDIN (Engineering Design Integration) System library developed for the generation, manipulation, display, computation of mass properties and data base management of panelled geometry. The GTM is composed of computer programs and associated data for performing configuration analysis on geometric shapes. The program can be operated in batch or demand mode and is designed for interactive use.

17. Key Words (Suggested by Author(s)) EDIN, geometry, design, technology, data base.	18. Distribution Statement Unclassified - Unlimited

19. Security Classif. (of this report) Unclassified	20 Security Classif. (of this page). Unclassified	21. No. of Pages 118	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FORWARD

This final report describing the formulation of the Geometry Technology Module (GTM) is provided in accordance with NASA Contract NAS9-13584. The report is presented in two volumes as follows:

VOLUME I Geometry Technology Module Engineering Description and Utilization Manual.

VOLUME II Geometry Technology Module Programmers' Manual

This work was conducted under the direction of Mr. Robert Abel of the Engineering Analysis Division, National Aeronautics and Space Administration, Johnson Spacecraft Center.

The authors wish to express their appreciation to Sigma Corporation, Houston, Texas, and its employees for significant contributions to the preparation of these reports.

TABLE OF CONTENTS

GEOMETRY TECHNOLOGY MODULE (GTM)

VOLUME II - PROGRAMMERS' MANUAL

By: S. J. Reiners, G. N. Hirsch, W. N. Colquitt, G. E. Alford and C. R. Glatt

Aerophysics Research Corporation

SUMMARY

This volume documents the program logic, subroutine descriptions and other information concerning the Geometry Technology Module of interest to the programmer.

INTRODUCTION

The program was written to provide complete generality wherever possible without sacrificing computational speed or computer storage. The guidelines used were as follows:

1. Computer core size of approximately 24000 decimal (overlayed).
2. Fortran V programming language.
3. Minimum program execution language.
4. Modular program construction.
5. Generalized routines to allow creation or manipulation of geometry.
6. Generalized routines for interfacing to the EDIN System.

Information pertinent to the programmer is presented in the following sections of this report. Included are descriptions of the program logic and overlay structure, flow diagrams and subroutine descriptions.

PROGRAM STRUCTURE

The Geometry Technology Module is coded in Fortran V. Overlays are used to minimize computer core requirements on the Exec 8 system. The total program requires approximately 24000 decimal of computer storage.

The GTM is composed of several major executive levels. These levels are called by the GTM executive. The major executive levels are the input module, cluster edit module and segment edit module. Figure 1 illustrates the GTM executive structure.

The MASTER module (GTM Executive) is the control point in the GTM from which all sublevel executives are accessed. It contains its own language set which allows the user to perform data base management functions, access sublevel executives and general program control. Three primary sublevel languages are available, input, segment edit and cluster.

The INPUT sublevel executive is provided for reading data which is stored in specific geometry formats. Two are available, the Gentry format of reference 1 and the GTM format. GTM format allows free-field data to be entered. The data may be any type of information. This data is read in and stored in the data base geometry tree structure. The INPUT module contains its own language set and associated menus, which can be displayed upon command.

The CLUSTER EDIT Module contains a language subset and instructions necessary for creating and maintaining the geometric data tree structure. Functions are also provided for translation, rotation and scaling of tree stored data and output of the data in forms for interfacing with other EDIN technology modules. In addition, it contains the necessary logic to display geometry for image viewing. The display functions have a number of features which allow the user to zoom in on a specific region, overlay geometry, scale geometry and filter geometry for resolution. Mass properties evaluations are also commanded from the CLUSTER EDIT Module.

The SEGMENT EDIT Module provides the capability to compose geometric shapes, manipulate geometry at the segment level and display of geometric segments. Specific operations include translation, rotations, scaling, point redistributions, segment cutting, point edit commands and display. The module contains its own language subset addressable by the user.

Unit Designation

Unit 1	Internal file designation for the geometry data base.
Unit 3	Output file for Gentry geometry.
Unit 5	The system card reader.
Unit 6	The system printer.

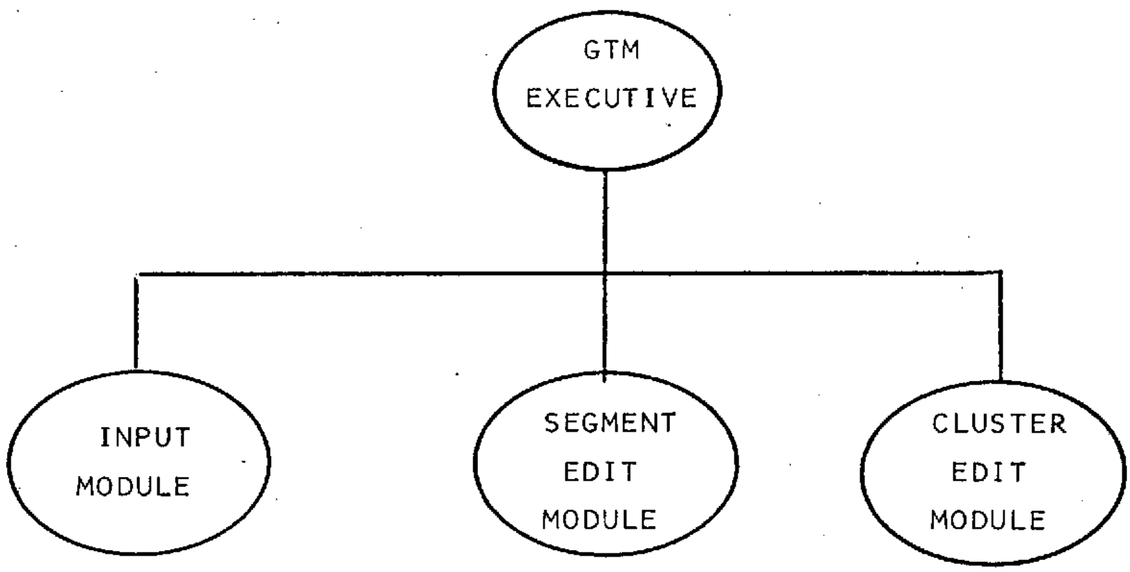
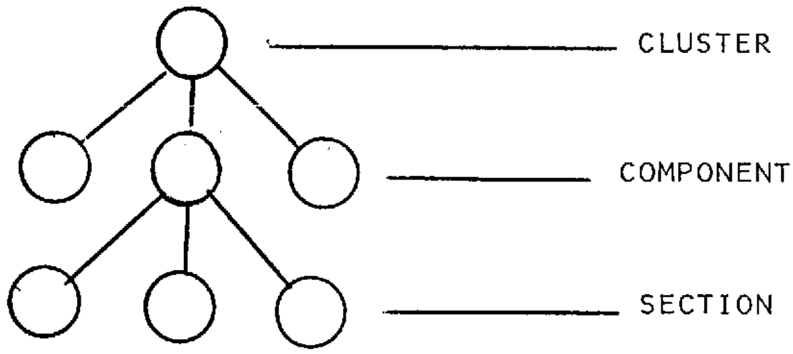


FIGURE 1 GTM EXECUTIVE STRUCTURE.

Tree Structured Data

The GTM uses a tree structuring method which allows data to be stored independently but can be associated with other stored data. The association may be permanent or temporary depending upon user specified structuring. The GTM data structuring technique is referred to as tree structuring and is defined at three levels (branches or nodes). The sketch below illustrates the several levels:



One level below the section level, called a segment level, may be defined but this data is freely stored and not associated with a tree. One level above the cluster level is the vehicle which can consist of several clusters but is not included in the tree structure logic. This section deals primarily with the tree structure logic in the GTM.

The discussion begins with the description of the control registers, those core locations reserved for indices to the actual data stored on disk. Throughout the GTM there is a labeled common block called TREEV which contains the control registers. The most significant registers are maintained in the IACV array of dimension 40. IACV contains the locations of an access address as to the tree structured data used in the GTM. The array is broken up into eight groups of five word groups. The meaning of these eight words in each group is as follows:

The first position is a pointer of the cluster name in the list of available clusters, data block.

The second position is the name and first word address of the cluster.

The third position is the position within the vehicle of the data block of the name and address of the current or accessed component name.

The fourth position is the component name and first word address of the component data.

The fifth position of the pointer is the component data block of the currently accessed section.

The sixth position is the section name and first word address of the section which has been currently accessed.

The seventh data block is the current position pointer of the data within the section that has been accessed.

The eighth position block is independent of the tree structure but is necessary information. This block contains the next available address location in the packed data structure used within the GTM. This location is used to control the access limit of all of the functions dealing with tree data withinin the GTM.

The access or the limits of the operation are controlled by the lowest level of the access register which is filled. For example, if an operation is to apply to a whole vehicle, then the controlling is done by having the only position filled within the access register being the second address block, the rest being zero. If only a section is to be processed, then all positions must be filled through and including the sixth block or the section named in the first named address of that section. All operations within the GTM are controlled by an implied limitation and that limitation is that the extent of the tree to be addressed and processed begins at the lowest level of the access register that is filled. In other words, if a vehicle name is the lowest level filled, then the entire vehicle is to be processed. If a component name is the lowest level filled, then only a component is to be processed. If the section name is the lowest level filled, then only a section name is to be processed. A number of routines within the GTM have been written to create, maintain and utilize this particular access register. These will be described in this document.

DMAN Software Package

DMAN provides all of the basic data management functions to handle variable length data pages while allowing them to be referenced by name. A data page may be stored on any file which has been established for data base use. All or portions of a data page contents may be retrieved. Modification of the contents of a data page is permitted, including that which requires increasing or decreasing the size of a page. Finally, removal of a data page from a file may be accomplished.

DMAN Usage. - The DMAN data management system is a Fortran callable software package which has been written for access and retrieval of data from the EDIN data base. The package consists of the following subroutines which must be included in the calling program:

DMAN	Basic Read/Write Controller.
NXTAD	Extend File Routine.
UPACK7	Character Unpack Routine.
RITBF	Write Routine.
PACK7	Character Packing Routine.
REDBF	Read Routine.
NWBLK	Create a New Block for Data.

The use requires the following declarations in the user program:

```
COMMON/UNITS/IAREA(273)
DATA IAREA/0,n,271*0/
INTEGER IT(5),IBUF(256)
```

where n is the file number where the data base is stored. The usage is as follows:

```
CALL DMAN(IOP,IT,N,IDATA,IBUF,IAREA(1),IAREA(2))
```

IOP	The read/write option. A further discussion of these options is given later.
IT	A five word array containing the data title. A further discussion of the titles is given below.
N	This variable contains the number of words in IDATA to be read or written. When reading, and the requested list cannot be satisfied, this value is reset to the number of words actually read, so this item must always be a variable when reading data.
IDATA	An integer or real array containing the data to be stored in the data base. There is no restriction on the length of this array.
IBUF	A 256 word buffer area for use by DMAN.

IAREA — This is a unit dependent area needed by DMAN. It must be dimensioned 273. One IAREA is required for each unit using DMAN. The double appearance of this array in the calling sequence is required for interal addressing purposes. This area must be protected, such as in COMMON, and must be reserved for use by DMAN while this file is being used.

A Discussion of IT. - There are two significant portions to the five word array IT. The first three words of the title are user supplied hollerith words which represent the name of the data item which is to be accessed or stored in the data base. If this is the first access of this data in the data base, the fourth word must be set to zero. This zeroing of the fourth title word will also return access to the beginning of the data set stored under the title given in the first three words.

The fourth and fifth words of the title are reserved for use by DMAN. If the fourth word is zero, a search is made of index arrays to find the address of the desired data set. This address is then inserted into these two words. Each time some activity occurs using this title, the address stored in these two words is updated so that this address always refers to the next word after the last word accessed. This eliminates the need to search the index arrays for each access of the data.

A Discussion of IOP. - IOP controls the type of reading or writing done by DMAN. The I/O options are:

IOP = 10 - write a matrix. The complete data set to be stored under the title IT is present in IDATA.

= -10 - read a matrix.

= 20 - write a single fixed length record.

= -20 - read a single fixed length record.

= 21 - write a single variable length record. Using this type of write option, an end-of-record mark is inserted after the end of the record. Any variable length record read will not pass this mark when reading. If the read is a fixed length record read, however, this mark will be ignored.

= -21 - read a variable length record. In this case, N is the number of words requested. The read will continue until N words have been read, and end-of-record mark is found, or the data set is exhausted, whichever comes first. The value of N will be set to the number of words actually returned.

= 30 - extend a data set with a fixed length record. The data in IDATA is to be appended to the existing data set stored under the title in IT.

= 31 - extend a data set with a variable length record.

NOTE: If a read attempt is made, which will extend the read past the end of the stored data set, or the data set requested has not been stored, the following values will be returned by DMAN:

N=0 and IDATA(1)=3LEOD.

IOP = 6HPURGE - this option will cause the title given in IT to be purged from the index array.

IOP = 6HCLEAR - this action will cause the buffer IBUF to be cleared. That is output to disc if necessary. This action is necessary before releasing the buffer to other uses, or existing a subroutine or overlay under conditions which will not protect the buffer.

IOP = 6HCLOSE - this action conditions the data base so that the entire contents of the data base do in fact reside on disc. It is necessary to execute this statement on any catalogued data base to insure that its entire contents are on disc. Normal activity may proceed after the function is called, and this function may be called as many times as desired.

Subroutine Descriptions

The GTM is divided into three classes of subroutines categorized by the functions which they perform. Applications subroutines process the input commands. Tree structuring routines construct and maintain the system of pointers called registers which control the access to related sets of geometry data. General utility routines are used for performing utility functions such as string processing, sorting and merging operations, vector and matrix arithmetic, etc.

This section presents a selected set of subroutine descriptions for the GTM. Flow charts are presented in Appendix A.

Subroutine ADDCOM. - The purpose of this routine is to execute the command, ADD component.

Subroutine ADDSEC. - The purpose of this routine is to execute the command, ADD section.

Subroutine ADDSEG. 0 The purpose of this routine is to execute the command, ADD segment.

Subroutine ADDVEH. - The purpose of this routine is to execute the command, ADD vehicle or ADD cluster.

Subroutine AFILT. - The purpose of this routine is to filter out geometric panels for drawing purposes. All panels less than a given area are deleted from the plot buffer.

Subroutine BCOMLS. - The purpose of this routine is to execute the command, BUILD component.

Subroutine BLDACC. - The purpose of this routine is to execute the command, ACCESS component.

Subroutine BLDACS. - The purpose of this routine is to execute the command, ACCESS section.

Subroutine BLDACV. - The purpose of this routine is to execute the command, ACCESS cluster.

Subroutine BLDAPT. - The purpose of this routine is to execute the command, ADD point.

Subroutine BLDCOM. - The purpose of this routine is to execute the command, BUILD component.

Subroutine BLDCSC. - The purpose of this routine is to execess the command, COPY section.

Subroutine BLDCSG. - The purpose of this routine is to execute the command, COPY segment.

Subroutine BLDDPT. - The purpose of this routine is to execute the command, DELETE point.

Subroutine BLDEQR. - The purpose of this routine is to execute the command, EQLEN.

Subroutine BLDEXT. - The purpose of this routine is to execute the command, EXTERNAL name. The command will cause control to be transferred from the GTM to the ED Processor for the purpose of editing a segment of data. The routine first copies the segment data to a BCD file (logical Unit 8) so that it can be edited. An operations stack internal to the GTM is employed to accomplish the actual transfer out of the GTM and back again.

Subroutine BLDFPT. - The purpose of this routine is to execute the command, FIND point.

Subroutine BLDHLG. - The purpose of this routine is to execute commands of the highest level language. The commands processed by this routine are IMAGE INPUT, CLUSTER EDIT, SEGMENT EDIT, INPUT, CALCULATOR, MENU, STOP, SAVE DATA BASE, OPSTACK and EXTERNAL. The commands IMAGE INPUT, CLUSTER EDIT, SEGMENT EDIT, INPUT and CALCULATOR all transfer control from this routine to other language processors. The remaining commands are executed within this routine.

Subroutine BLDIPT. - The purpose of this routine is to execute the command, INSERT point.

Subroutine BLDLSC. - The purpose of this routine is to execute the command, LIST section.

Subroutine BLDLSG. - The purpose of this routine is to execute the command, LIST segment.

Subroutine BLDMNU. - The purpose of this routine is to execute the command, MENU.

Subroutine BLDRPT. - The purpose of this routine is to execute the command, REPLACE point.

Subroutine BLDSEC. - The purpose of this routine is to execute the command, BUILD section.

Subroutine BLDSTP. - The purpose of this routine is to execute the command, STOP.

Subroutine BLDSTR. - The purpose of this routine is to execute the command, START.

Subroutine BLDVEH. - The purpose of this routine is to execute the command, BUILD cluster.

Subroutine BLOCCM. - The purpose of this routine is to execute the command, LOCATE component.

Subroutine BLOCSC. - The purpose of this routine is to execute the command, LOCATE section.

Subroutine BLOCSG. - The purpose of this routine is to execute the command, LOCATE segment.

Subroutine BLOCVH. - The purpose of this routine is to execute the command, LOCATE cluster.

Subroutine BUFSET. - The purpose of this routine is to initialize a data area as a buffer pool for use by DMAN.

Subroutine CALCUL. - The purpose of this routine is to execute the command, CALCUL. The routine provides the user a quickly accessible calculator type functions within the GTM.

Subroutine CLEAR. - The purpose of this routine is to clear the buffer data area assigned to DMAN prior to releasing the core area

Subroutine COMROT. - The purpose of this routine is to execute the rotation, scaling and translation functions on tree stored clusters, components and sections.

Subroutine COPLST. - The purpose of this routine is to execute the commands, COPY cluster, COPY component.

Subroutine COPYCM. - This routine performs the data transfer for the command, COPY component.

Subroutine COPYTH. - This routine writes geometry data stored in the data base in Gentry format.

Subroutine COPYVH. - This routine executes the command COPY cluster.

Subroutine COPY3. - The purpose of this routine is to perform the copying of tree stored data into a sequential Gentry formatted data file.

Subroutine CUTSEG. - The routine will cut a segment with a plane and define a new segment containing all of the points either above or below the specified plane.

Subroutine DELCOM. - The purpose of this routine is to execute the command, DELETE component.

Subroutine DELSEG. - The purpose of this routine is to execute the command, DELETE section.

Subroutine DELVEH. - The purpose of this routine is to execute the command, DELETE cluster.

DIRCOS. - This routine will compute the direction cosines of a plane or a vector having the principal angles of PSI, THETA and PHI. The calling arguemnts are:

```
CALL DIRCOS(PSI,THETA,PHI,DXG,DYG,DZG)
```

PSI	Yaw angle, degrees.
THETA	Pitch angle in degrees.
PHI	Roll angle in degrees.
DXG	X-direction cosine.
DYG	Y-direction cosine.
DZG	Z-direction cosine.

Subroutine DISCF. - The purpose of this routine is to calculate the scale factor for plotting or for displaying the geometry. The scaling is based on the maximum vehicle dimensions and the allowed display area on the display device.

Subroutine DISERC. - The calling parameters for this routine are:

```
CALL DISERC (IOP,IT,LEN,IBUF,FOUND,IUNDAT)
```

Definitions of the calling parameters are:

IOP	(Operation Control) This may have anyone of the following values: 3LDEL, 3LREP, 3LINS, 3LADD.
IT	Contains the title or address position of the level of which this operation is to take place.

LEN	This is the length of the record which is to be inserted or handled.
KT	This is the record. LEN is the length of KT. KT contains the name which is to be inserted, if necessary.
IBUF and IUNDAT	These are DMAN requirements.
FOUND	A logical variable, if true, the operation was successful. If false, the operation was not successful.

This routine is the main utility called by the routines TREECH, TREEV8 and TREESC. It will accomplish the table search and in addition, will perform the instructions as specified in the following table:

3LDEL	Delete the value if given in KT from the block named in IT.
3LREP	Replace the value found previously by a call to this routine with an option of 3LFND with the present value given in KT.
3LINS	Insert the record or title given in KT into the block IT as a position which must have been previously specified by a 3LFND operation.
3LADD	Add the title given in KT to the block IT, either at the end of the location.
3LFND	This is a FIND or access command in this routine to locate and return the correct addresses in IT and KT.

The entire address is not expected in the array KT but rather the first three hollerith names of the DMAN title are expected. The basic storage technique used in the tree structure is straight linear type. The searches are linear and the data is put into the table in a linear manner.

Subroutine DISPLY. - The purpose of this routine is to execute the commands, DISPLY+, DISPLY-, DISPLAY and ZOOM.

Subroutine EXTERN. - The purpose of this routine is to transfer control from the GTM to another program.

GETLNG. - This is a short utility routine which can be used to retrieve the proper language from the permanent data base. In many cases, the same core storage area will be used for several different languages. This utility can be used to test this area to see if the proper language is in core, and if not, retrieve the proper language from the data base. The calling sequence for this subroutine is as follows:

CALL GETLNG(IT,LGVAL,IBUF,IER)

IT — A three word array containing the name of the language desired.

LGVAL — The storage array described in LANG. Note: For the present time this area must be dimensioned at 1503 words.

IBUF — A buffer area for use by DMAN.

IER — An error flag set to .TRUE., if the desired language could not be found in the data base.

Subroutine GETPAN. - The purpose of this routine is to read a five point panel of data from the data base. The data returned by the subroutine is five points comprising four vectors which represent the boundary of a geometric panel

Subroutine GTMINP. - This is an executive subroutine for controlling input data in Gentry format.

Subroutine GTMPLT. - The purpose of this routine is to plot an array of data points. This array is assumed to consist of two dimensional points after having been rotated into the viewing plane.

IDIREC. - The purpose of this function subroutine is to determine whether or not the line or the direction of the line defined by the array LINE is facing towards or away from the plane as defined in the array PLANE (see PIERCE). The calling arguments are:

I = IDIREC(LINE,PLANE)

LINE — 6-Word array defining two points (x,y,z,X_2,Y_2Z_2) on a line.

PLANE — 6-Word array defining a point in the direction cosines of a plane (see PIERCE).

I — If I is returned as negative, the normal of the plane and the line points in the opposite direction.

If I is returned as positive, the normal and the line are in the same direction.

IDSRT. - This subroutine will sort a section in a radial manner. The average point of the section is computed using the routine PTAVG. This point then forms the center about which the data is sorted. Angles are computed between the average point and every other point on the section. The points are then sorted into order based on the value of the angle. This sort routine can only be used if the average point is interior to all section points. The calling parameters are:

CALL IDSRT(IOP,IT,IBUF,KBUF,IUNDAT)

IOP Integer identification of the sort type.

If IOP = 1, Sort the section based on its projection on the x-y plane.

If IOP = 2, Sort the section based on its projection on the x-z plane.

If IOP = 3, Sort the section based on its projection on the y-z plane.

If IOP is positive, the section is to be sorted into increasing values in a clockwise manner.

If IOP equals a negative value, execute a counter-clockwise sort.

IT This is a five word array containing the title of the section to be sorted.

IBUF, KBUF and IUNDAT are data management requirements (see DMAN).

Function IGOOD. - This function checks the validity of a point to determine if the point falls within the boundaries of the plot window.

Subroutine INPUT. - The purpose of this routine is to read input data from the data base.

INTBCD. - This function subroutine has one calling argument. The purpose of this routine is to convert an integer binary value into a hollerith character string. The calling argument is the binary value which is to be converted. The subroutine returns the string left justified and blank filled. That is an integer 1 being input will be returned from the function INTBCD as '1 '.

INTERO. - This subroutine is the primary input interrogation subroutine. Its purpose is to accept a list of characters and construct a list of words based on previously defined word boundary delimiters. The calling arguments for subroutine INTERO are:

```
CALL INTERO(NC,IC,NVL,IVL,XVL,ITP,IDEL,IOP)
```

NC — This is the number of characters in array IC.

IC — An array containing the characters to be interrogated. The characters must be stored in this array in 1R format.

NVL — This is the number of words in the character sequence given in IC.

IVL — An array returned containing the hollerith representation of the words found. These are right justified blank filled words. For use on Univac 1100 series computers, this array must be dimensioned twice the size of XVL. Each word can be up to 12 characters in length so this requires two Univac words.

XVL — An array returned containing the numeric values of this word if the word is a numeric word. The values returned are always real. If integers are required, the values must be converted in the calling program.

ITP — An integer array returned containing a type key as to the type of values found. The values returned and their meanings are as follows:

- 0 Nul word inserted between two non-blank delimiters if the option was specified.
- 1 Word found, was a numeric word.
- 2 Word found, was an illegal numeric word (a word which begins as a number and ends with alpha characters).
- 3 Word found was a legal name.
- -N These values are reserved for delimiter identifications. If the correct option is specified delimiters will be included in the returned list of recognized words. See the documentation on delimiters. NOTE: blank always a delimiter.

IDEL An array containing the non-blank delimiters to be used. See the documentation of subroutine STING for further information.

IOP The interrogation control option. The meaning of the four acceptable options are:

0 Return words and values found only.

1 Insert nul words in the returned list of words between back-to-back non-blank delimiters.

2 Return both words and non-blank delimiters found.

3 Return both words and non-blank delimiters found with nul words inserted between back-to-back non-blank delimiters.

The following example illustrates the use of the routines. The input card is:

TEST = 1, ,, 3.999764E-10

This card has been read using an 80R1 format. The results are given for using INTERO with each of the four options.

IOP = 0

NVL = 3

IVL		XVL	ITP
6HTEST	6H	0.	3
6H1	6H	1.	1
6H3.9997	6H	( )	1

IOP =1

NVL = 5

IVL		XVL	ITP
6HTEST	6H	0.	3
6H1	6H	1.	1
6H	6H	0.	0
6H	6H	0.	0
6H3.99997	6H64E-10	( )	1

IOP = 2

NVL = 7

IVL		XVL	ITP
6HTEST	6H	0.	3
6H=	6H	0.	-1
6H1	6H	1.	1
6H,	6H	0.	-2
6H,	6H	0.	-2
6H,	6H	0.	-2
6H3.9997	6H64E-10	( )	1

IOP = 3

NVL = 9

IVL		XVL	ITP
6HTEST	6H	0.	3
6H=	6H	0.	-1
6H1	6H	1.	1
6H,	6H	0.	-2
6H	6H	0.	0
6H,	6H	0.	-2
6H	6H	0.	0
6H,	6H	0.	-2

INTER2. - Function to determine the type of an input character. This function is dependent upon the Univac collating sequence which determines whether the character input is a number or an ALPHA character. The calling argument to this function is:

I = INTER2(IC)

IC Input character right justified and blank filled (1R format).

If it is a number, the integer value is returned. A minus one indicates that the character IC is not numeric.

INTER3.. - A routine for packing characters into a two word array. This is a machine dependent subroutine using the Univac Fortran IV FLD function. The three calling arguments are:

I = INTER3(NVL,IVL,KC)

NVL

IVL A two word array into which the hollerith representation of a word is to be packed.

KC The number of characters to be packed.

INTGR. - This is a function included in the GTM to insure that the conversion from real to integer is rounded up in such a manner that the correct value of the integer form. A floating point number is always returned. This function is used frequently in conjunction with INTERO which always returns real values. The calling sequence is:

I = INTGR(F)

IRXBCD. - This subroutine converts a real value into a hollerith string. It will automatically select the format which will yield the maximum number of characters possible and still yield the correct value. The format will be F for values which will fit within that field. If the field is not sufficient for a F format, then it automatically switches to an E format. The routine is primarily used in the GTM to output the Harris formatted information. The routine is not particularly efficient but it does function very well. It should also be noted that this routine will round correctly in all cases except where an additional character is added. That is, if ten is internally represented as 9.9999, this routine can not round it up to 10 but will output it as 9.999. In all other cases, the numbers are corrently rounded. The calling arguments for subroutine IRXBCD are:

CALL IRXBCD (NCHAR,XVAL,WRD,IOPT)

NCHAR — The number of characters which are to appear in the converted string. It should be noted that the minimum number of characters which can be displayed and give a correct representation for any range of values is 7. This would allow for a sign, an argument, a decimal, an exponent, if necessary, and a sign on the exponent. Therefore, the minimum should be 7.

XVAL — The binary real word which is to be converted into a character string.

WRD — An array which will contain the output string. It should be noted that this is an array because the output character string will, most likely, exceed the character capacity of one machine word. It automatically continues on to the next word. Therefore, for the Univac 1110, a 7 character return would have the first six characters in WRD(1) and 1 character in WRD(2).

IOPT — This is an option which allows the suppression of the letter E in the exponent designation.

If IOPT is 0, the exponent is supressed.
If IOPT is 1, the character E is included if the exponential form is selected.

<u>Subroutine ISRTCM</u>. - The purpose of this routine is to execute the command, INSERT component.

<u>Subroutine ISRTSC</u>. - The purpose of this routine is to execute the command, INSERT section.

<u>Subroutine ISRTVH</u>. - The purpose of this routine is to execute the command, INSERT cluster.

<u>KEYF</u>. - The purpose of the function is to provide an interface for the expansion of the RANDAC hash code computation for collision avoidance. A coded return value of the function KEYF is the first value of KEY. It is intended to hash code a multiword key. The calling parameters are:

K = KEYF(KEE,MKEY)

LANG. - Function LANG is the primary language statement recognition routine. This function will accept INTERO output and return a determination as to whether the statement contains a phrase which is part of an established language. The calling arguments for function LANG are:

IVAL = LANG(NVL,IVL,I,LGVAL)

IVAL — The phrase number found in this statement. If no phrase was found, this variable is set to zero.

NVL — The number of words in this statement. This value is returned by INTERO.

IVL — The hollerith representation of the statement (returned by INTERO).

I — The first word in this statement to be part of the phrase. If a phrase is found, this pointer is reset to the next word which was not part of the phrase found.

LGVAL — This array contains the stored phrase information. This information is packed into this array in the proper manner by subroutine LANGST. For use with GTM, a data storage program has been written which will execute the proper functions and enter any language into the data base in its proper stored manner.

LANGST. - This is a subroutine which interfaces the information from the language routine INECO to a format acceptable to the RANDAC directory routine. The definitions of the calling parameters are:

CALL LANGST(NVL,IVL,XVL,LGVAL)

NVL, IVL and XVL are all values returned from subroutine INTERO.

LGVAL — This is an array set aside for storing the language information in a RANDAC form.

When language blocks are being created, they are created from separate elements. Each language element consists of two portions. The first is the non-zero numeric value followed by one or more words of information. The information is read in BCD format and interpreted by INTERO. The information output from INTERO is passed by the array LGVAL for use by RANDAC.

LGPREP. - This is an initialization subroutine for establishing the RANDAC directory and associated attributes. The definitions of these calling parameters are:

CALL LGPREP(IATRIV,LGVAL,LGVLEN)

IATRIV This is the attribute table for the RANDAC addressable block. The attributes which must be input are:

$IATRIV_1$ The length of the key name.

$IATRIV_2$ The number of unique key names.

$IATRIV_3$ The maximum length of the longest title.

$IATRIV_4$ The number of unique titles.

LGVAL This is the directory which will be set up for the RANDAC calls.

LGVLEN This is the length of the directory LGVAL.

LINTRV. - This routine generates a modified distribution of surface points based on a given geometric point set and input value of x,y or z. The calling parameters are:

CALL LINTRV(IOP,IFIRST,IT,VALI,VALO,VALH,IBUF,IUNDAT)

IOP Integer identification of interpolation type.

IOP = 1, then x is input interpolating for y.

IOP = 2, then x is input interpolating for z.

IOP = 3, then y is input interpolating for z.

IOP = 4, then y is input interpolating for x.

IOP = 5, then z is input interpolating for x.

IOP = 6, then z is input interpolating for y.

If IOP is positive, then the sequence input is defined by the title IT as an increasing sequence.

If IOP is negative, then the sequence is decreasing.

IFIRST This routine will allow an envelope type interpolation. That is, the function input can be double valued function. If FIRST is set to 0, then this is the first call to the routine. It means that the title put in IT is to be reset to its initial value and the search is to begin from the beginning for this value.

IT The 5 word title of the section to search.

VALI This contains the input argument which is the value to be interpolated.

VALO This contains the interpolative value found. It should be noted that VALO is set = 3LERR, then the interpolation was out of range. Extrapolation is not allowed in this routine.

VALH An 11 word array required by this routine but is used internally. No initialization requirements are placed on this particular array.

IBUF and IUNDAT are data management requirements (see DMAN).

Subroutine LISTAV. - The purpose of this routine is to execute the command, LIST available clusters.

Subroutine LISTCM. - The purpose of this routine is to execute the command, LIST component.

Subroutine LISTVH. - The purpose of this routine is to execute the commands, LIST cluster and TREE list.

Subroutine LOCBUF. - The purpose of this routine is to manage the buffer area assigned to DMAN and to control I/O to satisfy DMAN requests.

Subroutine LSTDAT. - This routine is the main data retrieval routine for tree structured data. The routine is called by all subroutines extracting data from a tree structure.

MAXMIN. - The use of the routine is to determine maximum-minimum extent for use with plotting, etc. It should be noted that the array VALS must be initialized prior to being called by this routine. The calling parameters to this subroutine are:

CALL MAXMIN(IT,VALS,IBUF,IUNDAT)

IT This is the title of the item to search for MAXMIN values. This is a dimension array, dimension 5.

VALS This is a dimension 6. This array will contain the maximum and minimum x,y,z values for data information. The maximums and minimums are stored in the following manner:

POSITION 1	Maximum x
POSITION 2	Minimum x

POSITION 3	Maximum y
POSITION 4	Minimum y
POSITION 5	Maximum z
POSITION 6	Minimum z

IBUF and IUNDAT — These are requirements of the data base management system.

<u>MXV</u>. - This subroutine multiplies a 3 x 3 matrix by a 3 x 1 matrix resulting in a 3 x 1 matrix (vector) result. The calling sequence is:

CALL MXV(M,V,VV)

M — 3 x 3 input matrix.

V — 3 x 1 input matrix.

VV — 3 x 1 output vector.

All three arrays are real and V and VV may be the same array, if so desired.

<u>Subroutine MXVALS</u>. - The purpose of this routine is to determine the maximum and minimum points of a geometric set.

<u>Subroutine NWBLK</u>. - The purpose of this routine is to store the pointers to the data blocks written on disks.

<u>PACKWG</u>. - This routine is used for packing partial word information. It can be used for packing characters into full words or for packing small integers into larger words. The GTM uses it for both purposes. The routine calls the Univac function FLD, for the bit manipulation. The calling arguments of this routine are:

CALL PACKWG (IVL,IC,NC,IBP,IBPW)

IVL — An array into which the characters will be packed. This array must be long enough to take the packed string of characters IC.

IC — An array containing the characters or integers which are to be packed. Characters must be stored in a right justified format. That is an 1R format, one character per word.

NC — The number of characters in the array IC which are to be packed.

IBPB The number of bits per bite (or character) used for packing for the Univac 1110. This value is six, meaning six bits in each bite.

IBPW The number of bits per word. Again, with the Univac 1110, the standard value is 36.

PIERCE. - This is a routine written by NASA Langley Research Center for use in the GTM. Given a plane defined by its directive cosines and a point in the plane, the routine computes the intersecting point of a line defined by two arbitrary points. Calling parameters are:

CALL PIERCE(PLANE,LINE,POINT,INTSER)

PLANE This is an array of dimension 6. The first three locations contain the x,y and z coordinates of a point on the plane. The last three positions contain the three direction cosines of the plane.

LINE This is a 6 word array containing two points which define a line in space. The first three positions contain the x,y and z of the first point. The last three positions contain the x,y and z of the second point.

POINT POINT is an output array of dimension 3 which contains the x,y and z values of the point at which the line intersects the plane.

INTSEC This is a code returned by the subroutine. If INTSEC = 0, then the line pierces the plane but not between its two defining points.

If INTSEC = 1, then the plane intersects the line between the two input points.

If INTSEC $\neq$ 1 or 0, then the line does not intersect the plane.

Subroutine PLTGTM. - The purpose of this routine is to draw a single vector. It also checks the end points of the vector to determine the position of the vector with respect to a specified window. The vector is truncated to fit within the window.

POINTR. - The purpose of this subroutine is to provide a pointer to a new or unused block of disk storage in the DMAN data base system. This particular routine is called from the TREEV structuring routines and its purpose is to provide the address of disk space for titles or for information which is not addressed from DMAN itself but is rather addressed from the tree structured data system. The calling arguments for this routine are:

CALL POINTR(IT,IUNDAT)

IT	A five word title array.
IUNDAT	The fieldata array IUNDAT which is discussed in the documentation of DMAN.

PTAVG. - This routine will compute the average point of a section. The calling arguments are:

CALL PTAVG(KT,VALS,IFIRST,IBUF,IUNDAT)

KT	An array dimension 5 containing the title of the section for which the average point is to be computed.
VALS	An array dimension 3 which contains the x,y and z coordinates of the average point of the section.
IFIRST	IFIRST must be set equal to 0 prior to the first call. Including this particular parameter in the calling sequence allows you to compute a running average over several sections of values encompassed by more than one section.
IBUF and IUNDAT	These are data management requirement values. (See DMAN).

Subroutine PTPAIR. - The purpose of this routine is to provide a pair of points from parallel sections in a component of data in a tree structured data array.

Subroutine PUSHDW. - The purpose of this routine is to establish and maintain a push down stack.

RANDAC. - RANDAC is an access and retrieval subroutine for maintaining a directory of data base information. RANDAC uses the hash and collision methods of entering a table of data by keys. The table may contain the actual data or refer to alternate storage location (and/or devices). A chaining method assures uniqueness of all entries in the directory. The definitions of the calling parameters are:

CALL RANDAC(JOB,KEY,NAME,FOUND,FSL)

JOB — JOB may have an integer value of 1 to 5. The meaning of the values are:

If JOB = 1, the option is to initialize the directory.

If JOB = 2, the option is locate the KEYed entry.

If JOB = 3, the option is install the KEYed entry.

If JOB = 4, the option is delete the KEYed entry.

If JOB = 5, the option is write the directory.

KEY — A dimensional array containing the multiword hollerith key to be used for entering the table. The names can be variable length for different directories but must be fixed for the given directory.

NAME — The value which is to be stored in the table associated with the name of the variable found in KEY. It can be data or a reference to other data locations.

FOUND — FOUND is a logical variable which is set equal to TRUE and if the JOB option is successful.

FSL — An array used for storage of the directory. FSL must be prepared prior to usage by a call to a subroutine LGPREP.

Subroutine REDBF. - The purpose of this routine is to read geometry data blocks from a disk.

Subroutine RITBF. - The purpose of this routine is to write geometry data blocks on a disk.

Subroutine RPCPT. The purpose of this routine is to replace a point in a section or a point in a segment of data.

SCMVRT. = This routine will sort a section of geometric data, that is x, y and z points, into an increasing or decreasing sequence of x, y or z depending upon the parameters. The definitions of the calling parameters are:

CALL SCMVRT(IOP,LEN,ISRTWD,IT,IBUF,IUNDAT)

IOP — If IOP = 1, then an ascending order sort is to be executed.

If IOP = -1, then a descending order sort is to be accomplished.

LEN — This is the record length. This is equal to 3 for geometric data.

ISRTWD — This is the key on which to sort. The sort will work only for fixed length records. In the case of geometric data, if this variable is set to 1, then we are to sort on x; if equal to 2, then sort on y; if equal to 3, then sort on z.

IT — This is the 5 word title array containing the first word address of the section or data block to sort.

IBUF and IUNDAT are data management requirements (see DMAN).

SECARE. - This routine will determine the projected area of any section of data as it is projected on any one of the three principal planes. Following is a definition of the calling parameters:

CALL SECARE (IOP,IT,AVG,AREA,IBUF,IUNDAT)

IOP — If IOP = 1, the area is to be computed on the x-y projection.

If IOP = 2, the area is to be computed as projected on the x-z plane.

If IOP = 3, the area is to be computed as projected on the y-z plane.

IT — This is the title of the section for which the area is to be computed.

AVG — This is an array, dimension 3, containing a point which is to be used as the HUB of the area projection. The area is computed by making a triangle of this point and pairs of points of the section from which the area is to be computed. The area of this triangle is then computed and summed over the entire section.

AREA This is the value of the area which is returned.

IBUF and IUNDAT These are requirements of the data management system (see DMAN).

It should be noted that negative areas are not computed. Areas are always positive. Therefore, correct areas can only be arrived at if the AVG point is internal to the area encompassed by the section.

Subroutine SEGDIS. - The purpose of this routine is to execute the command, DISPLAY segment.

Subroutine SEGNAM. - The purpose of this routine is to check the names of segments to determine if certain types of operations may be performed on them.

SETIT. - This subroutine is an utility used to establish data base title names or DMAN title names. The purpose is to place titles into the representative title positions when they are input from the keyboard. If the title is input, and is less than five words long, the remainder of the words are blank filled. The calling arguments for this subroutine are:

```
CALL SETIT(IT,NVL,IVL,JPOS)
```

IT A five word array into which the title is to be placed.

NVL and IVL These are values returned. IVL is an array and NVL is the number of words in IV.

JPOS The position of the first unused word in IVL.

SETVAL. - The purpose of this routine is to copy the contents of one array to another. The parameters are:

```
CALL SETVAL(N,IDATIN,IDATOT)
```

N The length of the arrays IDATIN and IDATOT.

IDATIN An array containing the data which is to be copied to the array IDATOT.

IDATOT Output array for contents of IDATIN.

STRING. - This routine is a string recognization routine. However, it functions a little differently than most commonly used string routines. These differences will be pointed out. The calling arguments for routine STRING are:

```
CALL STRING (LISTCH,IVGN,ILAST,IUSED,LISTDL,NVSTR)
```

LISTCH — An array containing the list of characters which are to be examined. Note that these characters must be stored in an 1R format or a right justified one character per word.

IVGN — The position in the array LISTCH which must be the first character of a string. In other words, the first character of every string which can be recognized must be in this position.

ILAST — The last character position which can be compared. (i.e. the length of the string of characters of input in LISTCH)

LUSED — This routine can handle strings of varying lengths. Therefore, this parameter is the number of characters that were found.

LISTDL — This is an array containing the strings which are to be searched. A further description of how the data is stored in this array will be given later.

NVSTR — Every string is assigned or must be given a non-zero code number. When a string is found, this number will be returned and will also serve as a flag that a string has been encountered.

Storing the string information into an array LISTDL: Position 1 always contains the number of strings which are in the array. If the number is positive, then all of the strings are single character strings. That is, there is only one character to be recognized. In this case, the data is stored into the array in the following manner:

POSITION 1 — Contains the number of strings.

POSITION 2 — Contains the first character which is a string.

POSITION 3 — Contains the code number associated with that string.

POSITION 4 — Contains the second single character string.

POSITION 5 Contains the code number associated with that string and etc. throughout the length of the string.

If multiple delimiters using characters of more than one string are to be input, it must be done in the following manner. The first position of LISTDL will be the number of delimiters. In this case, it must be assigned a negative value. Position 2 contains the number of characters in the string. Positions 3 through n+2, where n is the number of characters in the string, contain the string itself. Position n+3 contains the code word which identifies this string. This sequence is repeated for every delimiter which is to be placed in this list. Special attention must be paid to delimiters which carry the same sequence of letters; one being longer than the other. In this case, in order to recognize them, the longest one must be input first. For example, AAA and AA can be recognized as legitimate strings if AAA is input into this list before the double A string. This routine uses a linear search of both the data and the delimiter list.

Subroutine TISWAP. - The purpose of this routine is to exchange the titles on two data names.

Subroutine TITST. - This is an error test routine used to determine whether two titles are the same.

Subroutine TREECM. - This routine has the identical calling parameters and the identical definitions of the calling parameters to the routine TREEV8. There is only one difference in its operation. The routine operates at the component level rather than the vehicle or cluster level. In all cases the vehicle must have been accessed before any component operation can occur.

Subroutine TREESC. - This routine has the identical calling parameters and calling parameter definitions to TREECM and TREEV8. Again there are no differences in the way this routine operates from the other routines except that this operates at the section level instead of the component or cluster level.

<u>Subroutine TREEV8</u>. - The calling parameters of this routine are:

CALL TREEV8 (IOP, IT, FOUND, IVTREE, IBUF, IUNDAT)

Definitions of calling parameters of this routine are:

IOP	This is the option controlling the action to be taken by this routine. The values which IOP may have are: INSERT, REPLACD, ADD, FIND and DELETE. These values are all hollerith names which are input to this routine.
IT	This is a five word DMAN which is to be entered or accessed from the tree depending upon the value of IOP.
FOUND	Is a logical variable returned TRUE if the action specified is completed, returned FALSE if the action specified could not be completed.
IVTREE	This is the 40 word access array.
IBUF and IUNDAT	Are used by DMAN.

The command INSERT results in the insertion of a new vehicle or cluster into the list of available clusters. It also implies that a position for this insertion has been specified by a previous access at the cluster level so that the cluster array has information or title positions in the access register. In this case, the vehicle or cluster is inserted in a position in front of the currently accessed vehicle.

REPLACE is a command to replace the vehicle which has been previously accessed and whose name is in register 2 with the value input under the title IT.

ADD adds the title input in title (1) IT, to the list of available clusters. In this case, the cluster need not have been accessed prior to this call. It should be noted that if this title duplicates a title that is already in the list of available clusters, this title replaces the one which is currently there. This is done automatically and no message is given.

FIND is the access operation of this routine. FIND will locate the vehicle given by the title in IT and will establish the access register to reflect the fact that this cluster has been located.

DELETE will delete the clusters specified in IT from the list of available clusters.

UNPKRG. - This subroutine has an identical calling sequence to the routine PACKWG and it is the inverse routine of PACKWG. The arguments are identical. This routine will unpack full word arrays and place the information into the specified array. It can be used for unpacking characters read in A6 format into 1R format. It is also used in certain places for unpacking small integers which are stored several to a word.

Subroutine USEOPS. - The purpose of this routine is to transfer control to a stored operations stack.

XFORM. - This subroutine computes the transformation matrix for a yaw, pitch and roll rotation sequence. The calling sequence is:

CALL XFORM(A,B,C,X)

A Yaw angle in degrees.

B Pitch angle in degrees.

C Roll angle in degrees.

X The output 3 x 3 matrix and transformation cosines.

ZERO. - The purpose of this routine is to set an array equal to 0. The calling arguments are:

CALL ZERO(N,IDAT)

N The number of values of IDAT to be set to zero.

IDAT The array to be zeroed.

Subroutine ZOOMSC. - The purpose of this routine is to compute a new scale factor and zeros for plotting of geometric data.

OVERLAY STRUCTURE

```
@MAP,L *U,TM.SEGG,GTM
MAP 0026-12/21-14:07 -(0,)

     1.        LIB *TEKLIB
     2.        NOT CHARS/TEKPLOT
     3.        NOT TKCH
     4.        LIB *ODIN-PETE
     5.        SEG MAIN
     6.        IN GTM
     7.        IN TKTRNX
     8.        IN TEKOM
     9.        IN TREEV
    10.        IN GTMBUF
    11.        SEG AA*,(MAIN)
    12.        IN GTMSED
    13.        SEG BB*,(MAIN)
    14.        IN BLDHLG
    15.        SEG CC*,(MAIN)
    16.        IN GTMINP
    17.        SEG DD*,(MAIN)
    18.        IN SEGEDT
    19.        SEG FF*,(MAIN)
    20.        IN INTTAL
    21.        SEG GG*,(MAIN)
    22.        IN CALCUL
    23.        SEG CCC*,(AA,BB,CC,DD,FF,GG)
    24.        IN INPUT
    25.        IN STRING
    26.        IN RANDAC
    27.        IN KEYF
    28.        IN INTER3
    29.        IN INTER2
    30.        IN LANG
    31.        IN GETLNG
    32.        IN INTERO
    33.        SEG DDA*,(DD)
    34.        IN SEGDIS
    35.        SEG DDB*,(DD,AA)
    36.        IN BLDMNU
    37.        SEG DDC*,(DD)
    38.        IN BLDFPT
    39.        SEG DDD*,(DD,AA)
    40.        IN BLDSTR
    41.        SEG DDZ*,(DD,AA)
    42.        IN BLDSTP
    43.        SEG DDE*,(DD)
    44.        IN BLDEQR
    45.        IN EQARC
    46.        IN SEGNAM
    47.        IN TISWAP
    48.        SEG DDF*,(DD)
```

```
[illegible]   IN [illegible]
[illegible]   IN [illegible]
51.   SEG DDG*,(DD)
52.   IN BLDDPT
53.   SEG DDH*,(DD)
54.   IN BLDIPT
55.   SEG DDI*,(DD,AA)
56.   IN BLDCSG
57.   SEG DDY*,(DD)
58.   IN SCNVRT
59.   SEG DDJ*,(DD,AA)
60.   IN BLDLSG
61.   SEG DDK*,(DD,AA)
62.   IN BLDEXT
63.   IN USEOPS
64.   SEG DDL*,(DD)
65.   IN CUTSEG
66.   IN IDIREC
67.   IN DIRCOS
68.   SEG DDM*,(DD,AA)
69.   IN BLDACV
70.   SEG DDN*,(DD,AA)
71.   IN BLDACC
72.   SEG DDO*,(DD,AA)
73.   IN BLDACS
74.   SEG DDP*,(DD,AA)
75.   IN BLDCSC
76.   SEG AAA*,(AA)
77.   IN COMROT
78.   SEG AAB*,(AA)
79.   IN ISRTVH
80.   SEG AAC*,(AA)
81.   IN BLDLSC
82.   SEG AAD*,(AA)
83.   IN BCOMLS
84.   IN BLDVEH
85.   SEG AAE*,(AA)
86.   IN BLDCOM
87.   SEG AAF*,(AA)
88.   IN ADDSEG
89.   IN ADDSEC
90.   SEG AAG*,(AA)
91.   IN ADDCOM
92.   IN ADDVEH
93.   SEG AAH*,(AA)
94.   IN ISRTSC
95.   SEG AAI*,(AA)
96.   IN ISRTCM
97.   SEG AAJ*,(AA)
98.   IN DELSEC
99.   SEG AAK*,(AA)
```

```
IN DELCOM
SEG AAM*,(AA)
IN DELVEH
SEG AAN*,(AA)
IN LISTCM
SEG AAO*,(AA)
IN LISTVH
SEG AAW*,(AA)
IN COPYCM
SEG AAX*,(AA)
IN COPYVH
SEG AAY*,(AA)
IN BLOCSG
SEG AAZ*,(AA)
IN BLOCSC
SEG ABA*,(AA)
IN BLOCCM
SEG ABB*,(AA)
IN BLOCVH
SEG ABC*,(AA)
IN BLDSEC
SEG ABD*,(AA)
IN LISTAV
SEG ABE*,(AA)
IN COPYTH
IN COPY3
SEG BBA*,(BB)
IN EXTERN
SEG CCA*,(CC)
IN INTBCD
SEG ZZ2*,(DDY,AAA,BBBC,BBBD)
IN XFORM
IN MXV
SEG ZZ4*,(AA,DDO,DDP,CC,DDM,DDN)
IN TREESC
IN DISERC
IN TREEVH
IN TREECM
SEG ZZ8*,(DDK,CCC)
IN PUSHDN
SEG BBB*,(AA)
IN DISPLY
SEG BBBA*,(AA)
IN PLINIT
SEG BBBC*,(BBB)
IN ZOOMSC
IN GETPAM
IN PTPAIR
IN XNORM
IN GTMSLT
IN AFILD
```

```
151.        IN [illegible]
152.        IN [illegible]
153.        IN CALPLT
154.        SEG JSBD*,(BBB)
155.        IN DISCF
156.        IN MAXVAL

R2 ELEMENT NOT FOUND:    TEKOM
AFCM STATUS OF OUTPUT ELEMENT=CLRAFCM

ADDRESS LIMITS     001000 036051       14890 IBANK WORDS DECIMAL
                   040000 066047       11304 DBANK WORDS DECIMAL
SEGMENT LOAD TABLE        040000 040347
INDIRECT LOAD TABLE       040350 041117
STARTING ADDRESS   025076


            SEGMENT MAIN            001000 025210       041120 057072


SYS$*RLIB$.NSWTC$/FOR69
                              $(1)   001000 001024
      EXTERNAL REFERENCES:   NTAB$, FMCTB$, IOCOD$, WRBLK$

SYS$*RLIB$.NWEF$/JSC69
                              $(1)   001025 001232       $(2)   041120 041137
      EXTERNAL REFERENCES:  NTAB$, NS11$, NHPFA$, IOCOD$, NFCHK$,
      NBFMG$, PACKT$, RDBLK$, UNIT$, UPDDA$, WAIT$, BS1BL$, DRAIN$,
      NBFGT$, NIOER$, NBFRL$, NSWTC$, NRBFA$, PUNCH$, PNCHA$, STREG$,
      PRINT$, NWALK$, CLOSE$, WEF$, IO$

SYS$*RLIB$.NBDCV$/FOR64
                              $(1)   001233 001360       $(2)   041140 041202
      EXTERNAL REFERENCES:  NC1UL0, NFDP$, NC1UL1

SYS$*RLIB$.NFTV$/FOR
                              $(1)   001361 001403

SYS$*RLIB$.NCNVT$/FOR68
                              $(1)   001404 001625       $(2)   041203 041277
      EXTERNAL REFERENCES:  STREG$, NSTSV$, NSTAT$, NCOM3$, NERCR$,
      NFTGL$, NCDOF$, NERCT$

SYS$*RLIB$.NRWND$/FOR68
                              $(1)   001626 001707       $(2)   041300 041311
      EXTERNAL REFERENCES:  NTAB$, NS11$, NHPFA$, IOCOD$, NFCHK$, WAIT$,
      NIOER$, NB$, DRAIN$, NRBF$, REW$, IO$, STREG$, PRINT$, NWALK$

SYS$*RLIB$.NCLOS$/FOR68
                              $(1)   001710 002100       $(2)   041312 041342
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      EXTERNAL REFERENCES:  NTAB$, NS11$, UNIT$, CSF$, IOW$, MB$, NWEF$,
      WAIT$, NREW$, NRBF$, STREG$, NCEF$, PRINT$, NWALK$, NTBSZ$,
      NIOER$, W$, IO$

SYS$*RLIB$.NBF00$/FOR
                                                    $(2)   041343 043544

SYS$*RLIB$.NFIND$/FOR68
                              $(1)   002101 002250        $(2)   043545 043615
      EXTERNAL REFERENCES:  NTAB$, NS11$, IOCOD$, NBFRS$, NEIRN$,
      NBFMG$, R$, IO$, WAIT$, W$, IOW$, NIOER$, NERU$, NTBSZ$, UNIT$,
      PACKT$, STREG$, NBTOD$, NSTAT$, NERCT$

SYS$*RLIB$.NININ$/FOR68
                              $(1)   002251 002441        $(2)   043616 043621
      EXTERNAL REFERENCES:  NTAB$, PACKT$, NFRH$, NREC$, NERU$, NRD$,
      NKLN$, NKL2$, NFRA$, NLLM$, NRTR$, NFTC2$, TEMP$, UNIT$, NFTCH$,
      NBCN$, NIIC$, NCSP$, NBIPA$, NEFCL$, READA$

SYS$*RLIB$.NINPT$/FOR69
                              $(1)   002442 003450        $(2)   043622 043652
      EXTERNAL REFERENCES:  NNG90$, NFGT$, IOCOD$, NR92$, NR93$, NLLC$,
      NFM96$, NFAR$, NFRZ$, NPW2$, NP91$, STREG$, NSTSV$, NSTAT$,
      NCOM3$, NFTGL$, NERCR$, NFCI$, NCNV9$, NSF$, NFSG$, NFDB$, NDBF1$,
      NDBCV$, NFRC$, NFRH$, NEFCL$, NFCM$, NDBIN$, NGC9$, NPCT$, NT10$,
      NFGC$, NRTR$, NFRG$, NDBLT$, READ$, NCSP$, NVEC$

SYS$*RLIB$.NRBLK$/FOR68
                              $(1)   003451 003473
      EXTERNAL REFERENCES:  NTAB$, UNIT$, WAIT$, NIOER$, R$, UPDDA$,
      IO$

SYS$*RLIB$.NFTCH$/FOR69
                              $(1)   003474 003756        $(2)   043653 043666
      EXTERNAL REFERENCES:  NTAB$, RDBLK$, WAIT$, NIOER$, IOCOD$,
      NBFRL$, NBFGT$, NBFMG$, R$, NF2Y1$, NIOER$A, NBFRS$, NFRONF$, MB$,
      UNIT$, MF$, IOW$, FNCTB$, UPDDA$, STREG$, NSTAT$, NERCT$

SYS$*RLIB$.NBSBL$/FOR68
                              $(1)   003757 004017
      EXTERNAL REFERENCES:  NTAB$, MB$, WAIT$, NIOER$, IOW$, UPDDA$

SYS$*RLIB$.NUPDA$/FOR68
                              $(1)   004020 004053
      EXTERNAL REFERENCES:  NTAB$, WAIT$, MB$

SYS$*RLIB$.NWBLK$/FOR68
                              $(1)   004054 004165
      EXTERNAL REFERENCES:  NTAB$, UNIT$, WAIT$, NIOER$, W$, UPDDA$,
      IO$
```

```
SYS$*RLIB$.[illegible]/FOR69
                                  $(1)   004166 004462        $(2)   043567 043672
      EXTERNAL REFERENCES:  NTAB$, NFRJ$, NREC$, NFPC$, NSTSU$, PACKT
      NERU$, NPU$, NPR$, NKLN$, NKL2$, NFRA$, NOLN$, NTEND$, NBFNG$,
      NC1UL0, NBFGT$, NBFRS$, WAIT$, NIOER$, UPIDA$, BS1BL$, FNCT2$,
      PNCHA$, NEXIT$, NCCC$, PPP$, PRNTA$, NCSP$, TEMP$, DRAIN$, UNIT$,
      NBFRL$, NCJNI02$, CFE, NI02$

SYS$*RLIB$.NOUT$/FOR69
                                  $(1)   004463 005637        $(2)   043673 043731
      EXTERNAL REFERENCES:  NCSP$, NFRJ$, NFPC$, IOCOD$, NPCT$, NR92$,
      NR93$, NRN92$, NFN96$, NFAR$, NFRZ$, NP91$, NBI$, NFNS1$, FNTOP,
      NFNS2$, NFNS3$, NDIG$, NSL$, NDOUT$, NIND$, NFGC$, NGC9$, NT10$,
      NFRA$, XFOR$, NR91$, NFMT$, PRNTA$, PRINT$, PUNCH$, NVEC$

SYS$*RLIB$.NFMT$/FOR69
                                  $(1)   005640 006514        $(2)   043732 044006
      EXTERNAL REFERENCES:  NTAB$, NFRZ$, NFRZ$S, NFMTR$, NFTGL$,
      NIO1U$, NFNIO1$, NIO3U$, NFNIO1D$, NIO3UA$, NDBI$, NAB7$, NAB0$,
      NAB4$, NAB2$, NAB5$, NAB3$, NAB1$, NAB6$, STREG$, NSTAT$, NERCR$,
      NFCS$, NDBCV$, NHVC$, NDBIN$, NXVC$, NAVC$, NFRG$, NRTR$, PRINT$,
      NFCA$, NVEC$, NI02$, IOCOD$, NCA$, NCHAR$, NSTSU$

SYS$*RLIB$.NIOER$/FOR69
                                  $(1)   006515 006704        $(2)   044007 044145
      EXTERNAL REFERENCES:  NTAB$, STREG$, UNIT$, NLRT$, NLTB$, NSTAT$,
      NCJNI02$, NTEND$, NS11$, NRSF$, NSAO$, PRINT$, PACKT$, NWALK$

SYS$*RLIB$.NFCHK$/FOR69
                                  $(1)   006705 007672        $(2)   044146 044321
                                                              $(4)   044322 044373
      EXTERNAL REFERENCES:  NTAB$, NERU$, NTBSZ$, UNIT$, NBTOD$, FITEM$,
      PL$, BL$, PACKT$, IOCOD$, STREG$, NSTAT$, PRINT$, NWALK$, NS11$,
      CSF$, WAIT$, NIOER$, W$, IOW$, UPIDA$, BS1BL$, MB$, TEMP$, DRAIN$,
      WRBLK$, NC1UL0, NC1UL1, B2L$, B20$, B10$, B1L$, CLOSE$, EXIT

SYS$*RLIB$.NTAB$/JSC
                                                              $(2)   044374 044433

SYS$*RLIB$.ERU$/SYS69
UOMSYS(COMMONBLOCK)                                                  044434 044450

SYS$*RLIB$.H$MONITOR/RFOR69
                                  $(1)   007673 011133        $(2)   044451 045257
                                                              $(4)   UOMSYS
      EXTERNAL REFERENCES:  PCT$, IALL$, PRINT$, FRSTI$, LASTI$, FRSTD$,
      LASTD$, TWAIT$, COND$, READ$, PRTCN$, SNAP$, EXIT$, ERR$, EABT$

SYS$*RLIB$.NWDA$/FOR69
                                  $(1)   011134 011701        $(2)   045260 045263
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
EXTERNAL REFERENCES:  NTAB$, IOCOD$, NHPFA$, NRSX$, NCDAF$,
NDASCD$, NFIND$, WAIT$, NIOER$, NIO1V$, NIO2V$, NIO3V$, NIO1B$,
NDT$, UNIT$, NRELD$, NDPFDL$, R$, IOW$, NDANW$, NNWDL$, NC1UL0,
NIO3$, NCSP$, NIO2$, NEXIT$, W$, NR91$, NTST0$, NFRA$, NBLNK$,
FHS10$, FHS20$, NFRJ$, NREC$, NFPC$, NSTSV$, NKLN$, NFMT$

SYS$*RLIB$.NRDA$/FOR69
                            $(1)  011702 012331      $(2)  045264 045276
    EXTERNAL REFERENCES:  NTAB$, NHPFA$, IOCOD$, NRSX$, NCDAF$,
    NDASCD$, NFIND$, WAIT$, NIOER$, NIO1V$, NIO2V$, NIO3V$, NIO1B$,
    NDT$, UNIT$, NRELD$, NDPFDL$, NDANW$, NNWDL$, NC1UL0, NIO3$,
    NIO2$, NFBY1$, NFRONF$, STREG$, NBCW$, NSTAT$, NERCT$, W$, IOW$,
    R$, NR91$, FHS1$, FHS2$, NFRH$, NREC$, NKLN$, NFRA$, NLLM$, NRTR$,
    NFMT$, NCSP$, NIIC$

SYS$*RLIB$.NDEF$/FOR69
                            $(1)  012332 013067      $(2)  045277 045376
    EXTERNAL REFERENCES:  NTAB$, NS11$, IOCOD$, NFCHK$, NERU$, NC1UL0,
    TEMP$, NC1UL1, NFPKT$, NBTOD$, CSF$, NFAF$, UNIT$, NBFMG$, R$,
    IOW$, NIOER$, W$, STREG$, PRINT$, NWALK$

SYS$*RLIB$.NINTR$/FOR69
                            $(1)  013070 013130      $(2)  045377 045415
    EXTERNAL REFERENCES:  NTPER$, STREG$, FIELD$, PRINT$, NEE$, CEND$,
    IALL$

SYS$*RLIB$.ATAN$/FOR59
                            $(1)  013131 013334      $(2)  045416 045447
    EXTERNAL REFERENCES:  NERRB$, NERRA$

SYS$*RLIB$.NIBUF$/FOR68
                            $(1)  013335 013374      $(2)  045450 045450
    EXTERNAL REFERENCES:  NTAB$, NHPFA$, NRSX$, IOCOD$, NFCHK$, NIO2$,
    NIO2V$, NR91$, FHS1$, FHS2$, NINI1$, NFMT$, NKLN$, NFRA$, NRTR$,
    NFRH$, NSTSV$, NNG90$

SYS$*RLIB$.NFINP$/FOR69
                            $(1)  013375 014022      $(2)  045451 045534
    EXTERNAL REFERENCES:  NTAB$, NHPFA$, NRSX$, IOCOD$, NFCHK$, NERU$,
    NIO1V$, NIO2V$, NIO3V$, NFTCB$, NIO1B$, STREG$, UNIT$, NSTAT$,
    NERCT$, NDT$, NDPFDL$, NFBY1$, NIO2$, NBFRL$, WAIT$, NIOER$,
    NDANW$, NNWDL$, NC1UL0, NIO3$

SYS$*RLIB$.NSTOP$/JSC
                            $(1)  014023 014070      $(2)  045535 045574
    EXTERNAL REFERENCES:  COM$, EXIT$, NRSF$, REST$, COND$, EABT$,
    IALL$, ERR$, PRINT$

SYS$*RLIB$.SINCOS$/FOR59
                            $(1)  014071 014223      $(2)  045575 045616
    EXTERNAL REFERENCES:  NERRA$, NERRB$
```

```
SYS$*RLIB$.SQRT$/FOR69
                              $(1)   014224 014264      $(2)   045617 045630
     EXTERNAL REFERENCES:   NERRA$

SYS$*RLIB$.NIER$/FOR69
                              $(1)   014265 014446      $(2)   045631 045751
     EXTERNAL REFERENCES:   NR93$, NS11$

SYS$*RLIB$.NOBUF$/FOR68
                              $(1)   014447 014507
     EXTERNAL REFERENCES:   NTAB$, NHPFA$, NR9X$, IOCOD$, NFCHK$, NERU$,
     PACKT$, NIOER$A, NTST0$, NIO2U$, NR91$, NBLNK$, FHS10$, FHS20$,
     NOTI1$, NB$, NFMT$, WAIT$

SYS$*RLIB$.NERR$/FOR69
                              $(1)   014510 015110      $(2)   045752 046142
     EXTERNAL REFERENCES:   PRINT$, NEE$, EABT$, NS11$

SYS$*RLIB$.IDL$/64
                              $(1)   015111 015157
     EXTERNAL REFERENCES:   SLT$, LOAD$

EX42-00002*TEKLIB.PARCLT
                              $(1)   015160 015264      $(0)   046143 046147
                                                        $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TEKLIB.PCLIPT
                              $(1)   015265 015324      $(0)   046150 046153
                              $(3)   TKTRNX             $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TEKLIB.WINCOT
                              $(1)   015325 015427      $(0)   046154 046174
                              $(3)   TKTRNX             $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   XYCNVT, NERR3$

EX42-00002*TEKLIB.CLIPT
                              $(1)   015430 016017      $(0)   046175 046220
                              $(3)   TKTRNX             $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   PARCLT, NERR3$

EX42-00002*TEKLIB.REVCOT
                              $(1)   016020 016110      $(0)   046221 046236
                              $(3)   TKTRNX             $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   PCLIPT, NERR3$

EX42-00002*TEKLIB.IOWAIT
                              $(1)   016111 016133      $(0)   046237 046243
                              $(3)   TKTRNX             $(2)   BLANK$COMMON
```

```
        EXTERNAL REFERENCES:   TDELAY, NERR3$

EX42-00002*TEKLIB.PNTMOD
                              $(1)   016134 016162       $(0)   046244 046254
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   TOUTPT, NERR3$

EX42-00002*TEKLIB.V2ST
                              $(1)   016163 016274       $(0)   046255 046272
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   CLIPT, VECMOD, WINCOT, PCLIPT, NERR3$

EX42-00002*TEKLIB.LVLCHT
                              $(1)   016275 016323       $(0)   046273 046276
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   REVCOT, NERR3$

EX42-00002*TEKLIB.ANMODE
                              $(1)   016324 016347       $(0)   046277 046303
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   TOUTPT, NERR3$

EX42-00002*TEKLIB.POINTA
                              $(1)   016350 016401       $(0)   046304 046310
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   LVLCHT, PNTMOD, V2ST, NERR3$

EX42-00002*TEKLIB.NEWPAG
                              $(1)   016402 016456       $(0)   046311 046322
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   ANMODE, TOUTPT, CHSYNC, IOWAIT, MOVABS,
        NERR3$

EX42-00002*TEKLIB.XYCNVT
                              $(1)   016457 016640       $(0)   046323 046340
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   TOUTPT, NERR3$

EX42-00002*TEKLIB.VECMOD
                              $(1)   016641 016700       $(0)   046341 046352
                              $(3)   TKTRNX              $(2)   BLANK$COMMON
        EXTERNAL REFERENCES:   TOUTPT, NERR3$

EX42-00002*TEKLIB.RESTAT
                              $(1)   016701 017010       $(2)   046353 046362
                              $(3)   TKTRNX
        EXTERNAL REFERENCES:   ANMODE, MOVABS, HCGTE$, VECMOD, PNTMOD,
        NERFN$
ASCTBL(COMMONBLOCK)                                             046363 046562

EX42-00002*TEKLIB.CHARSH
```

```
     $(1)   [illegible] 020065       $(0)   [illegible] 046073
                                     $(2)   046074 047000
                                     $0121ASCTBL
  EXTERNAL REFERENCES:  ERRPKT, W$, APRINT$, ATREAD$, AREAD$, ERR$,
  CSF$, ION$, EXIT, TWAIT$, AIOIB$

EX42-00002*TEKLIB.SUSTAT
     $(1)   020066 020115       $(0)   047001 047013
     $(3)   TKTRNX              $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  NERR3$

EX42-00002*TEKLIB.MOVABS
     $(1)   020116 020141       $(0)   047014 047017
     $(3)   TKTRNX              $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  VECMOD, XYCNVT, NERR3$

EX42-00002*TEKLIB.MOVEA
     $(1)   020142 020167       $(0)   047020 047024
                                $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  LVLCHT, VECMOD, V2ST, NERR3$

EX42-00002*TEKLIB.DRAWA
     $(1)   020170 020232       $(0)   047025 047031
     $(3)   TKTRNX              $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  LVLCHT, VECMOD, XYCNVT, V2ST, NERR3$

EX42-00002*TEKLIB.INITT
     $(1)   020233 020343       $(0)   047032 047053
     $(3)   TKTRNX              $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  NOGRAF, ANMODE, NEWPAG, NERR3$

EX42-00002*TEKLIB.PLOT
     $(1)   020344 020407       $(0)   047054 047057
                                $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  POINTA, MOVEA, DRAWA, NERR3$

EX42-00002*TEKLIB.ERASE
     $(1)   020410 020430       $(0)   047060 047157
                                $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  SVSTAT, NEWPAG, RESTAT, NERR3$

EX42-00002*TEKLIB.BELL
     $(1)   020431 020446       $(0)   047160 047164
                                $(2)   BLANK$COMMON
  EXTERNAL REFERENCES:  TOUTPT, CHSYNC, NERR3$

EX42-00002*TEKLIB.DMPBUF
     $(1)   020447 020502       $(2)   047165 047262
  EXTERNAL REFERENCES:  SVSTAT, CHSYNC, RESTAT, NERFN$

EX42-00002*TPF$.REDBF1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                        $(1)  020503 020624      $(0)  047263 047310
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  UPACK7, PACK7, NRDA$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.RITBF1
                        $(1)  020625 020715      $(0)  047311 047334
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  UPACK7, PACK7, NWDA$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.NWBLK1
                        $(1)  020716 021046      $(0)  047335 047344
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  PACK7, LOCBUF, NXTAD, NERR3$

EX42-00002*TPF$.LOCBUF
                        $(1)  021047 021357      $(0)  047345 047407
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  UPACK7, RITBF1, PACK7, REDBF1, NERR3$

EX42-00002*TPF$.PACK7
                        $(1)  021360 021502      $(0)  047410 047434
                                                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.UPACK7
                        $(1)  021503 021617      $(0)  047435 047461
                                                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.CLEAR
                        $(1)  021620 021712      $(0)  047462 047510
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  UPACK7, RITBF1, NERR3$

EX42-00002*TPF$.NXTAD
                        $(1)  021713 021761      $(0)  047511 047516
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  NDEF$, NERR3$

EX42-00002*TPF$.BUFSET
                        $(1)  021762 022024      $(0)  047517 047532
                        $(3)  MS                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.INTGR
                        $(1)  022025 022046      $(0)  047533 047540
                                                 $(2)  BLANK$COMMON
    EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.LOCPTR
                        $(1)  022047 022324      $(0)  047541 047612
```

```
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   DMAN1, NERR3$

EX42-00002*TPF$.TITST
                              $(1)   022325 022431       $(0)   047613 047651
                              $(3)   SYSTEM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NWDU$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.POINTR
                              $(1)   022432 022471       $(0)   047652 047664
                              $(3)   MS                  $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NXTAD, PACK7, NERR3$

EX42-00002*TPF$.SETIT
                              $(1)   022472 022561       $(0)   047665 047705
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.UNPKRG
                              $(1)   022562 022711       $(0)   047706 047737
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.ZERO
                              $(1)   022712 022740       $(0)   047740 047751
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.PACKWG
                              $(1)   022741 023074       $(0)   047752 050007
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.SETVAL
                              $(1)   023075 023132       $(0)   050010 050023
                                                         $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.DMAN1
                              $(1)   023133 025075       $(0)   050024 050166
                              $(3)   MS                  $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NXTAD, CLEAR, UPACK7, PACK7, LOCBUF, NWBLK1,
     NDEF$, NRDA$, NIO1$, NIO2$, NWDA$, NWDU$, NERR3$
EXCIGT(COMMONBLOCK)                                             050167 050167
MS(COMMONBLOCK)                                                 050170 050174
ROTCON(COMMONBLOCK)                                             050175 050213
UNITS(COMMONBLOCK)                                              050214 050634
SYSTEM(COMMONBLOCK)                                             050635 050637
GTNOPS(COMMONBLOCK)                                             050640 050657
BLANK$COMMON(COMMONBLOCK)
```

```
      EXTERNAL REFERENCES:  VECMOD, XYCNVT, NERR3$

EX42-00002*TPF$.CROSS
                              $(1)   026103 026164       $(0)   057157 057166
                                                         $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:  SQRT, NERR3$

EX42-00002*TPF$.RFRESH
                              $(1)   026165 026272       $(0)   057167 057205
                              $(3)   PLTPRM              $(2)   BLANK$COMMON
                              $(5)   UNITS               $(4)   GTMBUF
      EXTERNAL REFERENCES:  ERASE, DMPBUF, DMAN1, UNPKRG, DRWABS,
      MOVABS, BELL, NERR3$

EX42-00002*TPF$.VAMP
                              $(1)   026273 027252       $(0)   057206 057625
                              $(3)   TREEV               $(2)   BLANK$COMMON
                              $(5)   SYSTEM              $(4)   GTMBUF
                                                         $(6)   UNITS
      EXTERNAL REFERENCES:  SETVAL, XFORM, GETPAN, MXV, CROSS, MXM3X3,
      XPOSE, PAT, TOT, LOAD, MNCOD$, SQRT, NWDU$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.COPLST
                              $(1)   027253 027777       $(0)   057626 057706
                              $(3)   TREEV               $(2)   BLANK$COMMON
                              $(5)   SYSTEM              $(4)   GTMBUF
                                                         $(6)   UNITS
      EXTERNAL REFERENCES:  SETVAL, POINTR, LSTDAT, TREEVH, TREECM,
      DMAN1, TREESC, NERR2$, NWDU$, NIO2$, NERR3$

EX42-00002*TPF$.LSTDAT
                              $(1)   030000 030462       $(0)   057707 057731
                                                         $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:  SETVAL, DMAN1, ZERO, NERR2$, NERR3$
WINDO(COMMONBLOCK)                                              057732 057740
PLTPRM(COMMONBLOCK)                                             057741 060000

EX42-00002*TPF$.GTMSED
                              $(1)   030463 031672       $(0)   060001 063033
                              $(3)   TREEV               $(2)   BLANK$COMMON
                              $(5)   SYSTEM              $(4)   GTMBUF
                              $(7)   ROTCON              $(6)   UNITS
                              $(011)WINDO                $(010)PLTPRM
      EXTERNAL REFERENCES:  ZERO, PLINIT, INPUT, INTERO, GETLNG, LANG,
      BLDMNU, BLDSTR, BLDSTP, COMROT, SETIT, BLDCSG, BLDACV, BLDACC,
      BLDACS, BLDCSC, DISPLY, BLDLSC, BLDLSG, BLDEXT, DMAN1, BLDVEH,
      BLDCOM, BCOMLS, ADDSEG, ADDSEC, ADDCOM, ADDVEH, ISRTSC, ISRTCM,
      ISRTVH, DELSEC, DELCOM, DELVEH, LISTCM, LISTVH, COPYCM, COPYVH,
      BLOCSG, BLOCSC, BLOCCM, BLOCVH, SETVAL, VAMP, BLDSEC, LISTAV,
      COPYTH, RFRESH, NWDU$, NIO2$, NSTOP$, NERR2$, ATAN2, NERR3$
```

```
EX42-00002*TPF$.GTM
                                  $(1)  025076 025210       $(0)  050660 050677
                                  $(3)  SYSTEM              $(2)  BLANK$COMMON
                                  $(5)  UNITS               $(4)  GIMOPS
                                  $(7)  EXCIGT              $(6)  MS
                                                            $(010)GTMBUF
      EXTERNAL REFERENCES:  BUFSET, BLDHLG, GTMINP, GTMSED, SEGEDT,
      INITAL, CALCUL, NINTR$, NERR2$, NWDU$, NIO2$, NSTOP$
TKTRNX(COMMONBLOCK)                                               050700 050765
TREEV(COMMONBLOCK)                                                050766 052044
GTMBUF(COMMONBLOCK)                                               052045 057072

             SEGMENT AA*                    025211 031672         057073 063033
             FOLLOWS SEGMENT MAIN


SYS$*RLIB$.NOSYM$/FOR69
                                  $(1)  025211 025453       $(2)  057073 057074
      EXTERNAL REFERENCES:  NTAB$, NCA$, NHPFB$, ARRE, OUTCNT, ENDEC$,
      NCHAR$, NHPFA$, IOCOD$, NRSX$, PACKT$, NTST0$, NIO2U$, NR91$,
      NBLNK$, FHS10$, FHS20$, NFRJ$, NREC$, NFPC$, NSTSU$, NIO1$, NERU$,
      NPU$, NKLN$, NETF$, NPR$, NFRA$, NFMT$, FPC00, ARRN

EX42-00002*ODIN-PETE.OP
                                  $(1)  025454 025511       $(0)  057075 057110
                                                            $(2)  BLANK$COMMON
      EXTERNAL REFERENCES:  NERR3$

EX42-00002*ODIN-PETE.TOT
                                  $(1)  025512 025551       $(0)  057111 057124
                                                            $(2)  BLANK$COMMON
      EXTERNAL REFERENCES:  NERR3$

EX42-00002*ODIN-PETE.PAT
                                  $(1)  025552 025637       $(0)  057125 057135
                                                            $(2)  BLANK$COMMON
      EXTERNAL REFERENCES:  NERR3$

EX42-00002*ODIN-PETE.XPOSE
                                  $(1)  025640 025677       $(0)  057136 057143
                                                            $(2)  BLANK$COMMON
      EXTERNAL REFERENCES:  NERR3$

EX42-00002*ODIN-PETE.MXM3X3
                                  $(1)  025700 026041       $(0)  057144 057152
                                                            $(2)  BLANK$COMMON
      EXTERNAL REFERENCES:  NERR3$

EX42-00002*TEKLIB.DRWABS
                                  $(1)  026042 026102       $(0)  057153 057156
                                  $(3)  TKTRNX              $(2)  BLANK$COMMON
```

```
          SEGMENT BB*                025211 025604      057073 062132
          FOLLOWS SEGMENT MAIN


EX42-00002*TPF$.BLDHLG
                          $(1)   025211 025604      $(0)   057073 062132
                          $(3)   GTMOPS             $(2)   BLANK$COMMON
                          $(5)   UNITS              $(4)   SYSTEM
                          $(7)   TREEV              $(6)   EXCIGT
                                                    $(010)GTMBUF
     EXTERNAL REFERENCES:  INPUT, INTERO, GETLNG, LANG, BLDMNU, DMAN1,
     SETIT, USEOPS, EXTERM, PLOT, NWDU$, NIO2$, NSTOP$, NERR2$, NERR3$

          SEGMENT CC*                025211 026752      057073 062424
          FOLLOWS SEGMENT MAIN


EX42-00002*TPF$.GTMINP
                          $(1)   025211 026752      $(0)   057073 062424
                          $(3)   TREEV              $(2)   BLANK$COMMON
                          $(5)   SYSTEM             $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  INPUT, INTERO, GETLNG, LANG, SETIT, DMAN1,
     UNPKRG, INTBCD, POINTR, SETVAL, TREEVH, TREECM, TREESC, BLDMNU,
     NWDU$, NIO2$, NSTOP$, NERR2$, NIO1$, NRBU$, NRDU$, NERR3$

          SEGMENT DD*                025211 026305      057073 062135
          FOLLOWS SEGMENT MAIN


EX42-00002*TPF$.BLDAPT
                          $(1)   025211 025315      $(0)   057073 057106
                          $(3)   TREEV              $(2)   BLANK$COMMON
                          $(5)   SYSTEM             $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  TITST, DMAN1, ZERO, NERR3$

EX42-00002*TPF$.SEGEDT
                          $(1)   025316 026305      $(0)   057107 062135
                          $(3)   TREEV              $(2)   BLANK$COMMON
                          $(5)   SYSTEM             $(4)   GTMBUF
                          $(7)   ROTCON             $(6)   UNITS
     EXTERNAL REFERENCES:  ZERO, INPUT, INTERO, GETLNG, LANG, BLDMNU,
     BLDFPT, INTGR, BLDSTR, BLDSTP, BLDEOR, BLDAPT, BLDRPT, BLDDPT,
     BLDIPT, SETIT, BLDCSG, SCMVRT, SEGDIS, BLDLSG, BLDEXT, DMAN1,
     SETVAL, CUTSEG, BLDACV, BLDACC, BLDACS, BLDCSC, NWDU$, NIO2$,
     NSTOP$, NERR2$, ATAN2, NERR3$

          SEGMENT FF*                025211 026554      057073 065216
          FOLLOWS SEGMENT MAIN
```

```
EX42-00002*TPF$.LANGST
                                   $(1)   025211 025425        $(0)   057073 057146
                                                               $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   RANDAC, PACKWG, UNPKRG, NERR3$

EX42-00002*TPF$.LGPREP
                                   $(1)   025426 025612        $(0)   057147 057214
                                                               $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   RANDAC, NWDU$, NIO2$, NERR3$

EX42-00002*TPF$.INTTAL
                                   $(1)   025613 026554        $(0)   057215 065216
                                   $(3)   TREEV                $(2)   BLANK$COMMON
                                   $(5)   SYSTEM               $(4)   GTMBUF
                                                               $(6)   UNITS
      EXTERNAL REFERENCES:   DMAN1, LGPREP, LANGST, INPUT, INTERO, LANG,
      SETIT, PACKWG, INTGR, NWDU$, NIO2$, NRDU$, NIO1$, NIO3$, NERR2$,
      NERR3$

              SEGMENT GG*              025211 026063       057073 062173
              FOLLOWS SEGMENT MAIN


SYS$*RLIB$.TANCOTAN$/FOR59
                                   $(1)   025211 025406        $(2)   057073 057113
      EXTERNAL REFERENCES:   NERRB$, NERRA$, NERRC$

SYS$*RLIB$.NEXP6$/FOR68
                                   $(1)   025407 025603        $(2)   057114 057165
      EXTERNAL REFERENCES:   NERRA$, NERRB$, NERRC$

EX42-00002*TPF$.CALCUL
                                   $(1)   025604 026063        $(0)   057166 062173
                                   $(3)   TREEV                $(2)   BLANK$COMMON
                                   $(5)   SYSTEM               $(4)   GTMBUF
                                                               $(6)   UNITS
      EXTERNAL REFERENCES:   INPUT, INTERO, GETLNG, LANG, BLDMNU, NWDU$,
      NIO2$, NSTOP$, NERR2$, XPRR, SIN, COS, TAN, NERR3$

              SEGMENT CCC*             031673 035400       065217 066006
              FOLLOWS SEGMENT AA (LONGEST IBANK)
                      SEGMENT BB
                      SEGMENT CC
                      SEGMENT DD
                      SEGMENT FF (LONGEST DBANK)
                      SEGMENT GG


SYS$*RLIB$.NEXP5$/FOR68
```

```
                       $(1)   031673 031760      $(2)   065217 065226
    EXTERNAL REFERENCES:   NERRA$, NERRB$, NERRC$

EX42-00002*TPF$.INPUT
                       $(1)   031761 032523      $(0)   065227 065314
                       $(3)   GTMOPS             $(2)   BLANK$COMMON
                                                 $(4)   SYSTEM
    EXTERNAL REFERENCES:   DMAN1, UNPKRG, PUSHDN, SETIT, NWDU$, NIO1$,
    NIO2$, NERR2$, NRDU$, NSTOP$, NERR3$

EX42-00002*TPF$.STRING
                       $(1)   032524 032742      $(0)   065315 065350
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.RANDAC
                       $(1)   032743 033605      $(0)   065351 065514
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   KEYF, NERR2$, NWDU$, NIO2$, NIO1$, NERR3$

EX42-00002*TPF$.KEYF
                       $(1)   033606 033623      $(0)   065515 065522
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.INTER3
                       $(1)   033624 033674      $(0)   065523 065537
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   PACKNG, NERR3$

EX42-00002*TPF$.INTER2
                       $(1)   033675 033732      $(0)   065540 065546
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.LANG
                       $(1)   033733 034177      $(0)   065547 065663
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   RANDAC, PACKNG, NERR3$

EX42-00002*TPF$.GETLNG
                       $(1)   034200 034322      $(0)   065664 065712
                       $(3)   UNITS              $(2)   BLANK$COMMON
                                                 $(4)   SYSTEM
    EXTERNAL REFERENCES:   DMAN1, NWDU$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.INTERO
                       $(1)   034323 035400      $(0)   065713 066006
                                                 $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:   STRING, INTER2, INTER3, XPRI, NERR3$
```

```
               SEGMENT DDA*            026306 027600      062136 062310
               FOLLOWS SEGMENT DD

EX42-00002*TEKLIB.SWINDO
                         $(1)  026306 026326      $(0)  062136 062141
                         $(3)  TKTRNK             $(2)  BLANK$COMMON
     EXTERNAL REFERENCES:  NERR3$

EX42-00002*TEKLIB.FINITT
                         $(1)  026327 026353      $(0)  062142 062146
                                                  $(2)  BLANK$COMMON
     EXTERNAL REFERENCES:  MOVABS, ANMODE, NSTOP$, NERR3$

EX42-00002*TEKLIB.VWINDO
                         $(1)  026354 026374      $(0)  062147 062152
                         $(3)  TKTRNK             $(2)  BLANK$COMMON
     EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.SCRENE
                         $(1)  026375 026463      $(0)  062153 062166
                                                  $(2)  BLANK$COMMON
     EXTERNAL REFERENCES:  SWINDO, NERR3$
TEKCOM(COMMONBLOCK)                                     062167 062173

EX42-00002*TPF$.PLOTS
                         $(1)  026464 026707      $(0)  062174 062217
                         $(3)  TEKCOM             $(2)  BLANK$COMMON
     EXTERNAL REFERENCES:  INITT, DMPBUF, BELL, ERASE, DRAWA, MOVEA,
     FINITT, NWDU$, NIO2$, NRDU$, NERR3$

EX42-00002*TPF$.SEGDIS
                         $(1)  026710 027600      $(0)  062220 062310
                         $(3)  GTMOPS             $(2)  BLANK$COMMON
                         $(5)  GTMBUF             $(4)  TREEV
                         $(7)  UNITS              $(6)  SYSTEM
     EXTERNAL REFERENCES:  PLOTS, SETIT, DMAN1, SETVAL, LOCPTR, ERASE,
     VWINDO, SCRENE, PLOT, DMPBUF, ZERO, NWDU$, NIO1$, NIO2$, NERR3$

               SEGMENT DDB*            031673 031776      063034 063053
               FOLLOWS SEGMENT DD
                    SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)

EX42-00002*TPF$.BLDMNU
                         $(1)  031673 031776      $(0)  063034 063053
                         $(3)  TREEV              $(2)  BLANK$COMMON
                         $(5)  SYSTEM             $(4)  GTMBUF
                                                  $(6)  UNITS
     EXTERNAL REFERENCES:  SETIT, DMAN1, NWDU$, NIO1$, NIO2$, NERR3$

               SEGMENT DDC*            026306 026466      062136 062213
```

```
          FOLLOWS SEGMENT DD


EX42-00002*TPF$.BLDFPT
                        $(1)   026306 026466        $(0)   062136 062213
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
    EXTERNAL REFERENCES:  TITST, LOCPTR, NWDU$, NIO2$, NIO1$, NERR3$

          SEGMENT DDD*              031673 031771        063034 063045
          FOLLOWS SEGMENT DD
                SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDSTR
                        $(1)   031673 031771        $(0)   063034 063045
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
    EXTERNAL REFERENCES:  ZERO, SETIT, NERR3$

          SEGMENT DDZ*              031673 031771        063034 063045
          FOLLOWS SEGMENT DD
                SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDSTP
                        $(1)   031673 031771        $(0)   063034 063045
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
    EXTERNAL REFERENCES:  ZERO, SETIT, NERR3$

          SEGMENT DDE*              026306 027507        062136 062356
          FOLLOWS SEGMENT DD


EX42-00002*TPF$.BLDEQR
                        $(1)   026306 026527        $(0)   062136 062171
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
    EXTERNAL REFERENCES:  SETIT, LOCPTR, SEGNAM, EQARC, TISWAP, ZERO,
    NWDU$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.EQARC
                        $(1)   026530 027201        $(0)   062172 062267
                        $(3)   SYSTEM               $(2)   BLANK$COMMON
    EXTERNAL REFERENCES:  DMAN1, SQRT, NWDU$, NIO1$, NIO2$, NERR3$
```

```
EX42-00002*TPF$.SEGNAM
                          $(1)   027202 027275       $(0)   062270 062323
                          $(3)   SYSTEM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   DMAN1, NWDU$, NIO2$, NERR3$

EX42-00002*TPF$.TISWAP
                          $(1)   027276 027507       $(0)   062324 062356
                                                     $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   DMAN1, NERR3$

          SEGMENT DDF*              026306 026560       062136 062233
          FOLLOWS SEGMENT DD

EX42-00002*TPF$.BLDRPT
                          $(1)   026306 026445       $(0)   062136 062162
                          $(3)   TREEV               $(2)   BLANK$COMMON
                          $(5)   SYSTEM              $(4)   GTMBUF
                                                     $(6)   UNITS
     EXTERNAL REFERENCES:   TITST, LOCPTR, RPCPT, ZERO, NWDU$, NIO2$,
     NERR3$

EX42-00002*TPF$.RPCPT
                          $(1)   026446 026560       $(0)   062163 062233
                          $(3)   SYSTEM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   DMAN1, NWDU$, NIO1$, NIO2$, NERR3$

          SEGMENT DDG*              026306 026563       062136 062177
          FOLLOWS SEGMENT DD

EX42-00002*TPF$.BLDDPT
                          $(1)   026306 026563       $(0)   062136 062177
                          $(3)   TREEV               $(2)   BLANK$COMMON
                          $(5)   SYSTEM              $(4)   GTMBUF
                                                     $(6)   UNITS
     EXTERNAL REFERENCES:   TITST, SETIT, LOCPTR, DMAN1, ZERO, NWDU$,
     NIO2$, NERR3$

          SEGMENT DDH*              026306 026556       062136 062171
          FOLLOWS SEGMENT DD

EX42-00002*TPF$.BLDIPT
                          $(1)   026306 026556       $(0)   062136 062171
                          $(3)   TREEV               $(2)   BLANK$COMMON
                          $(5)   SYSTEM              $(4)   GTMBUF
                                                     $(6)   UNITS
     EXTERNAL REFERENCES:   TITST, DMAN1, ZERO, NWDU$, NIO2$, NERR3$

          SEGMENT DDI*              031673 032242       063034 063100
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          FOLLOWS SEGMENT DD
                    SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)

EX42-00002*TPF$.BLDCSG
                              $(1)  031673 032242      $(0)  063034 063103
                              $(3)  TREEV              $(2)  BLANK$COMMON
                              $(5)  SYSTEM             $(4)  GTMBUF
                                                       $(6)  UNITS
     EXTERNAL REFERENCES:  SETIT, DMAN1, SETVAL, ZERO, POINTR, LOCPTR,
     NWDU$, NIO1$, NIO2$, NERR3$

          SEGMENT DDY*               026306 026627      062136 062167
          FOLLOWS SEGMENT DD

EX42-00002*TPF$.SCMVRT
                              $(1)  026306 026627      $(0)  062136 062167
                              $(3)  TREEV              $(2)  BLANK$COMMON
                              $(5)  SYSTEM             $(4)  GTMBUF
                              $(7)  ROTCOM             $(6)  UNITS
     EXTERNAL REFERENCES:  XFORM, SETIT, SETVAL, LOCPTR, DMAN1, MXV,
     ZERO, NERR2$, NERR3$

          SEGMENT DDJ*               031673 032027      063034 063072
          FOLLOWS SEGMENT DD
                    SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)

EX42-00002*TPF$.BLDLSG
                              $(1)  031673 032027      $(0)  063034 063072
                              $(3)  TREEV              $(2)  BLANK$COMMON
                              $(5)  SYSTEM             $(4)  GTMBUF
                                                       $(6)  UNITS
     EXTERNAL REFERENCES:  SETIT, TITST, DMAN1, NWDU$, NIO1$, NIO2$,
     NERR3$

          SEGMENT DDK*               031673 032112      063034 063106
          FOLLOWS SEGMENT DD
                    SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)

EX42-00002*TPF$.BLDEXT
                              $(1)  031673 032036      $(0)  063034 063070
                              $(3)  TREEV              $(2)  BLANK$COMMON
                              $(5)  SYSTEM             $(4)  GTMBUF
                                                       $(6)  UNITS
     EXTERNAL REFERENCES:  SETIT, TITST, DMAN1, USEOPS, NREW$, NWDU$,
     NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.USEOPS
```

```
                           $(1)   032037 032112        $(0)   063071 063106
                           $(3)   GTMOPS               $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   PUSHDW, NERR3$

             SEGMENT DDL*              026306 027530        062136 062305
             FOLLOWS SEGMENT DD


EX42-00002*TPF$.PIERCE
                           $(1)   026306 026643        $(0)   062136 062177
                                                       $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.CUTSEG
                           $(1)   026644 027347        $(0)   062200 062250
                           $(3)   TREEV                $(2)   BLANK$COMMON
                           $(5)   SYSTEM               $(4)   GTMBUF
                           $(7)   ROTCOM               $(6)   UNITS
     EXTERNAL REFERENCES:   DIRCOS, SETIT, SETVAL, TITST, DMAN1, PIERCE,
     IDIREC, NWDU$, NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.IDIREC
                           $(1)   027350 027417        $(0)   062251 062263
                                                       $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.DIRCOS
                           $(1)   027420 027530        $(0)   062264 062305
                                                       $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   SIN, COS, NERR3$

             SEGMENT DDM*              031673 032002        063034 063064
             FOLLOWS SEGMENT DD
                     SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDACV
                           $(1)   031673 032002        $(0)   063034 063064
                           $(3)   TREEV                $(2)   BLANK$COMMON
                           $(5)   SYSTEM               $(4)   GTMBUF
                                                       $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, TREEVH, DMAN1, ZERO, SETVAL, NWDU$,
     NIO1$, NIO2$, NERR3$

             SEGMENT DDN*              031673 032002        063034 063064
             FOLLOWS SEGMENT DD
                     SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDACC
                           $(1)   031673 032002        $(0)   063034 063064
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                        $(3)   TREEV                    $(2)   BLANK$COMMON
                        $(5)   SYSTEM                   $(4)   GTMBUF
                                                        $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, TREECM, DMAN1, ZERO, SETVAL, NWDU$,
     NIO1$, NIO2$, NERR3$

          SEGMENT DDO*              031673 032006       063034 063063
          FOLLOWS SEGMENT DD
                SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDACS
                        $(1)   031673 032006            $(0)   063034 063063
                        $(3)   TREEV                    $(2)   BLANK$COMMON
                        $(5)   SYSTEM                   $(4)   GTMBUF
                                                        $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, TREESC, DMAN1, SETVAL, NWDU$, NIO1$,
     NIO2$, NERR3$

          SEGMENT DDP*              031673 032256       063034 063102
          FOLLOWS SEGMENT DD
                SEGMENT AA (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.BLDCSC
                        $(1)   031673 032256            $(0)   063034 063102
                        $(3)   TREEV                    $(2)   BLANK$COMMON
                        $(5)   SYSTEM                   $(4)   GTMBUF
                                                        $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, TREESC, DMAN1, SETVAL, ZERO, POINTR,
     LOCPTR, NWDU$, NIO1$, NIO2$, NERR3$

          SEGMENT AAA*              031673 032363       063034 063102
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.COMROT
                        $(1)   031673 032363            $(0)   063034 063102
                        $(3)   TREEV                    $(2)   BLANK$COMMON
                        $(5)   SYSTEM                   $(4)   GTMBUF
                        $(7)   ROTCOM                   $(6)   UNITS
     EXTERNAL REFERENCES:   SETVAL, XFORM, LSTDAT, MXV, DMAN1, ZERO,
     NERR2$, NERR3$

          SEGMENT AAB*              031673 032025       063034 063103
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.ISRTVH
                        $(1)   031673 032025            $(0)   063034 063103
                        $(3)   TREEV                    $(2)   BLANK$COMMON
                        $(5)   SYSTEM                   $(4)   GTMBUF
```

```
                                                   $(6)   UNITS
     EXTERNAL REFERENCES:   SETVAL, ZERO, SETIT, DMAN1, TREEVH, NWDU$,
     NIO2$, NIO1$, NERR3$

          SEGMENT AAC*              031673 032100      063034 063115
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLDLSC
                    $(1)   031673 032100      $(0)   063034 063115
                    $(3)   TREEV              $(2)   BLANK$COMMON
                    $(5)   SYSTEM             $(4)   GTMBUF
                                              $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, TREESC, DMAN1, SETVAL, NWDU$, NIO1$,
     NIO2$, NERR3$

          SEGMENT AAD*              031673 031755      063034 063047
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.BCOMLS
                    $(1)   031673 031722      $(0)   063034 063041
                    $(3)   TREEV              $(2)   BLANK$COMMON
                    $(5)   SYSTEM             $(4)   GTMBUF
                                              $(6)   UNITS
     EXTERNAL REFERENCES:   ZERO, SETIT, SETVAL, NERR3$

EX42-00002*TPF$.BLDVEH
                    $(1)   031723 031755      $(0)   063042 063047
                    $(3)   TREEV              $(2)   BLANK$COMMON
                    $(5)   SYSTEM             $(4)   GTMBUF
                                              $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, POINTR, TREEVH, NERR3$

          SEGMENT AAE*              031673 031750      063034 063043
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLDCOM
                    $(1)   031673 031750      $(0)   063034 063043
                    $(3)   TREEV              $(2)   BLANK$COMMON
                    $(5)   SYSTEM             $(4)   GTMBUF
                                              $(6)   UNITS
     EXTERNAL REFERENCES:   SETIT, POINTR, TREECM, ZERO, SETVAL, NERR3$

          SEGMENT AAF*              031673 032072      063034 063104
          FOLLOWS SEGMENT AA


EX42-00002*TPF$.ADDSEG
                    $(1)   031673 031742      $(0)   063034 063042
```

```
                                        $(3)   TREEV                 $(2)   BLANK$COMMON
                                        $(5)   SYSTEM                $(4)   GTMBUF
                                                                     $(6)   UNITS
        EXTERNAL REFERENCES:   SETIT, TITST, TREESC, NERR3$

EX42-00002*TPF$.ADDSEC
                                        $(1)   031743 032072         $(0)   063043 063104
                                        $(3)   TREEV                 $(2)   BLANK$COMMON
                                        $(5)   SYSTEM                $(4)   GTMBUF
                                                                     $(6)   UNITS
        EXTERNAL REFERENCES:   SETIT, TREESC, DMAN1, SETVAL, NWDU$, NIO1$,
        NIO2$, NERR3$

                 SEGMENT AAG*               031673 032173         063034 063140
                 FOLLOWS SEGMENT AA


EX42-00002*TPF$.ADDCOM
                                        $(1)   031673 032022         $(0)   063034 063076
                                        $(3)   TREEV                 $(2)   BLANK$COMMON
                                        $(5)   SYSTEM                $(4)   GTMBUF
                                                                     $(6)   UNITS
        EXTERNAL REFERENCES:   SETIT, TREECM, DMAN1, SETVAL, NWDU$, NIO1$,
        NIO2$, NERR3$

EX42-00002*TPF$.ADDVEH
                                        $(1)   032023 032173         $(0)   063077 063140
                                        $(3)   TREEV                 $(2)   BLANK$COMMON
                                        $(5)   SYSTEM                $(4)   GTMBUF
                                                                     $(6)   UNITS
        EXTERNAL REFERENCES:   SETIT, TREEVH, DMAN1, SETVAL, TREECM, NWDU$,
        NIO1$, NIO2$, NERR3$

                 SEGMENT AAH*               031673 032041         063034 063105
                 FOLLOWS SEGMENT AA


EX42-00002*TPF$.ISRTSC
                                        $(1)   031673 032041         $(0)   063034 063105
                                        $(3)   TREEV                 $(2)   BLANK$COMMON
                                        $(5)   SYSTEM                $(4)   GTMBUF
                                                                     $(6)   UNITS
        EXTERNAL REFERENCES:   SETVAL, ZERO, SETIT, DMAN1, TREESC, NWDU$,
        NIO2$, NIO1$, NERR3$

                 SEGMENT AAI*               031673 032041         063034 063106
                 FOLLOWS SEGMENT AA


EX42-00002*TPF$.ISRTCM
                                        $(1)   031673 032041         $(0)   063034 063106
```

```
                       $(3)   TREEV                $(2)   BLANK$COMMON
                       $(5)   SYSTEM               $(4)   GTMBUF
                                                   $(6)   UNITS
       EXTERNAL REFERENCES:  SETVAL, ZERO, SETIT, DMAN1, TREECM, NWDU$,
       NIO2$, NIO1$, NERR3$

            SEGMENT AAJ*             031673 032031      063034 063070
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.DELSEC
                       $(1)   031673 032031        $(0)   063034 063070
                       $(3)   TREEV                $(2)   BLANK$COMMON
                       $(5)   SYSTEM               $(4)   GTMBUF
                                                   $(6)   UNITS
       EXTERNAL REFERENCES:  SETIT, DMAN1, TREESC, SETVAL, ZERO, NWDU$,
       NIO1$, NIO2$, NERR3$

            SEGMENT AAL*             031673 032031      063034 063071
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.DELCOM
                       $(1)   031673 032031        $(0)   063034 063071
                       $(3)   TREEV                $(2)   BLANK$COMMON
                       $(5)   SYSTEM               $(4)   GTMBUF
                                                   $(6)   UNITS
       EXTERNAL REFERENCES:  SETIT, DMAN1, TREECM, SETVAL, ZERO, NWDU$,
       NIO1$, NIO2$, NERR3$

            SEGMENT AAM*             031673 032027      063034 063070
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.DELVEH
                       $(1)   031673 032027        $(0)   063034 063070
                       $(3)   TREEV                $(2)   BLANK$COMMON
                       $(5)   SYSTEM               $(4)   GTMBUF
                                                   $(6)   UNITS
       EXTERNAL REFERENCES:  SETIT, TREEVH, DMAN1, SETVAL, ZERO, NWDU$,
       NIO1$, NIO2$, NERR3$

            SEGMENT AAN*             031673 032076      063034 063110
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.LISTCM
                       $(1)   031673 032076        $(0)   063034 063110
                       $(3)   TREEV                $(2)   BLANK$COMMON
                       $(5)   SYSTEM               $(4)   GTMBUF
                                                   $(6)   UNITS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        EXTERNAL REFERENCES:  SETVAL, SETIT, TREECM, DMAN1, NWDU$, MIO1$,
        MIO2$, MERR3$

               SEGMENT AAO*             031673 032137      063034 063112
               FOLLOWS SEGMENT AA


EX42-00002*TPF$.LISTVH
                                $(1)  031673 032137     $(0)  063034 063112
                                $(3)  TREEV             $(2)  BLANK$COMMON
                                $(5)  SYSTEM            $(4)  GTMBUF
                                                        $(6)  UNITS
        EXTERNAL REFERENCES:  SETVAL, SETIT, TREEVH, DMAN1, ZERO, NWDU$,
        MIO1$, MIO2$, MERR3$

               SEGMENT AAW*             031673 032030      063034 063104
               FOLLOWS SEGMENT AA


EX42-00002*TPF$.COPYCM
                                $(1)  031673 032030     $(0)  063034 063104
                                $(3)  TREEV             $(2)  BLANK$COMMON
                                $(5)  SYSTEM            $(4)  GTMBUF
                                                        $(6)  UNITS
        EXTERNAL REFERENCES:  SETIT, TREECM, DMAN1, ZERO, COPLST, NWDU$,
        MIO2$, MIO1$, MERR3$

               SEGMENT AAX*             031673 032016      063034 063073
               FOLLOWS SEGMENT AA


EX42-00002*TPF$.COPYVH
                                $(1)  031673 032016     $(0)  063034 063073
                                $(3)  TREEV             $(2)  BLANK$COMMON
                                $(5)  SYSTEM            $(4)  GTMBUF
                                                        $(6)  UNITS
        EXTERNAL REFERENCES:  SETIT, TREEVH, DMAN1, ZERO, COPLST, NWDU$,
        MIO1$, MIO2$, MERR3$

               SEGMENT AAY*             031673 031762      063034 063062
               FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLOCSG
                                $(1)  031673 031762     $(0)  063034 063062
                                $(3)  TREEV             $(2)  BLANK$COMMON
                                $(5)  SYSTEM            $(4)  GTMBUF
                                                        $(6)  UNITS
        EXTERNAL REFERENCES:  SETIT, DMAN1, ZERO, SETVAL, NWDU$, MIO1$,
        MIO2$, MERR3$
```

```
            SEGMENT AAZ*              031673 032020        063034 063064
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLOCSC
                        $(1)   031673 032020        $(0)   063034 063064
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  SETIT, TREESC, DMAN1, ZERO, SETVAL, NWDU$,
     NIO1$, NIO2$, NERR3$

            SEGMENT ABA*              031673 032002        063034 063064
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLOCCM
                        $(1)   031673 032002        $(0)   063034 063064
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  SETIT, TREECM, DMAN1, ZERO, SETVAL, NWDU$,
     NIO1$, NIO2$, NERR3$

            SEGMENT ABB*              031673 032002        063034 063064
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLOCVH
                        $(1)   031673 032002        $(0)   063034 063064
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  SETIT, TREEVH, DMAN1, ZERO, SETVAL, NWDU$,
     NIO1$, NIO2$, NERR3$

            SEGMENT ABC*              031673 031777        063034 063060
            FOLLOWS SEGMENT AA


EX42-00002*TPF$.BLDSEC
                        $(1)   031673 031777        $(0)   063034 063060
                        $(3)   TREEV                $(2)   BLANK$COMMON
                        $(5)   SYSTEM               $(4)   GTMBUF
                                                    $(6)   UNITS
     EXTERNAL REFERENCES:  SETIT, SETVAL, POINTR, DMAN1, TREESC, NWDU$,
     NIO2$, NERR3$

            SEGMENT ABD*              031673 031760        063034 063063
            FOLLOWS SEGMENT AA
```

```
EX42-00002*TPF$.LISTAV
                                 $(1)   031673 031760       $(0)   063034 063063
                                 $(3)   TREEV               $(2)   BLANK$COMMON
                                 $(5)   SYSTEM              $(4)   GTMBUF
                                                            $(6)   UNITS
      EXTERNAL REFERENCES:  DMAN1, NWDU$, NIO2$, NIO1$, NERR3$

             SEGMENT ABE*             031673 034104       063034 063232
             FOLLOWS SEGMENT AA


SYS$*RLIB$.NBKSP$/FOR68
                                 $(1)   031673 032370       $(2)   063034 063061
      EXTERNAL REFERENCES:  NTAB$, NHPFA$, NS11$, IOCOD$, NFCHK$, WAIT$,
      UPDDA$, NBFMG$, NIOER$, R$, IOW$, MB$, DRAIN$, NBFRL$, NBFRS$,
      CFE, STREG$, PRINT$, BS1BL$, MF$, NRBFA$, RDBLK$, NWALK$

SYS$*RLIB$.NFOUT$/FOR69
                                 $(1)   032371 033007       $(2)   063062 063103
      EXTERNAL REFERENCES:  NTAB$, NHPFA$, NRSX$, IOCOD$, NFCHK$, NERU$,
      NIOER$A, NIO1V$, NIO2V$, NIO3V$, DRAIN$, MB$, MF$, IOW$, UPDDA$,
      NBFGT$, WAIT$, NIOER$, BS1BL$, NSWTC$, NBFMG$, NBFRS$, NIO1B$,
      STREG$, UNIT$, NSTAT$, NERCT$, NC1UL0, NDT$, NDPFDL$, NBFRL$, CFE,
      NIO2$, NDANW$, NWWDL$, NIO3$

EX42-00002*TPF$.COPYTH
                                 $(1)   033010 033173       $(0)   063104 063157
                                 $(3)   TREEV               $(2)   BLANK$COMMON
                                 $(5)   SYSTEM              $(4)   GTMBUF
                                                            $(6)   UNITS
      EXTERNAL REFERENCES:  SETIT, TREEVH, DMAN1, SETVAL, COPY3, NWDU$,
      NIO1$, NIO2$, NERR3$

EX42-00002*TPF$.COPY3
                                 $(1)   033174 034104       $(0)   063160 063232
                                                            $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:  LSTDAT, NERR2$, NWBU$, NIO1$, NIO2$, NWDU$,
      NBSP$, NRBU$, NRDU$, NERR3$

             SEGMENT BBA*             025605 026312       062133 062316
             FOLLOWS SEGMENT BB


SYS$*RLIB$.NERTRAN$/FOR68
                                 $(1)   025605 025763       $(2)   062133 062244
      EXTERNAL REFERENCES:  ABORT$, ERR$, EXIT$, CSF$, SETC$, COND$,
      DATE$, NERR$, STREG$, FIELD$, PRINT$

EX42-00002*TPF$.EXTERN
                                 $(1)   025764 026312       $(0)   062245 062316
```

```
                    $(3)   SYSTEM                $(2)   BLANK$COMMON
                                                 $(4)   QTMOPS
   EXTERNAL REFERENCES:   FUSHON, DMAN1, UNPKRG, ERTRAN, NWDU$, MIO1$,
   MIO2$, NSTOP$, NERR3$

            SEGMENT CCA*              026753 027056       062425 062450
            FOLLOWS SEGMENT CC


EX42-00002*TPF$.INTBCD
                    $(1)   026753 027056     $(0)   062425 062450
                                             $(2)   BLANK$COMMON
   EXTERNAL REFERENCES:   PACKWG, NERR3$

            SEGMENT ZZ2*              035624 036051       063763 064022
            FOLLOWS SEGMENT DDY
                    SEGMENT AAA
                    SEGMENT BBBC  (LONGEST IBANK)(LONGEST DBANK)
                    SEGMENT BBBD


EX42-00002*TPF$.XFORM
                    $(1)   035624 035760     $(0)   063763 064002
                                             $(2)   BLANK$COMMON
   EXTERNAL REFERENCES:   SIN, COS, NERR3$

EX42-00002*TPF$.MXV
                    $(1)   035761 036051     $(0)   064003 064022
                                             $(2)   BLANK$COMMON
   EXTERNAL REFERENCES:   NERR3$

            SEGMENT ZZ4*              032257 034523       063103 063413
            FOLLOWS SEGMENT AA
                    SEGMENT DDO
                    SEGMENT DDP  (LONGEST IBANK)(LONGEST DBANK)
                    SEGMENT CC
                    SEGMENT DDM
                    SEGMENT DDN


EX42-00002*TPF$.TREESC
                    $(1)   032257 032654     $(0)   063103 063154
                    $(3)   SYSTEM            $(2)   BLANK$COMMON
   EXTERNAL REFERENCES:   SETVAL, DISERC, ZERO, NWDU$, MIO2$, NERR3$

EX42-00002*TPF$.DISERC
                    $(1)   032655 033525     $(0)   063155 063271
                    $(3)   SYSTEM            $(2)   BLANK$COMMON
   EXTERNAL REFERENCES:   DMAN1, NWDU$, MIO1$, MIO2$, NERR3$

EX42-00002*TPF$.TREEVH
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                        $(1)   033526 034071       $(0)   063272 063336
                        $(3)   SYSTEM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  SETVAL, DISERO, ZERO, NERR3$

EX42-00002*TPF$.TREECM
                        $(1)   034072 034523       $(0)   063337 063413
                        $(3)   SYSTEM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  SETVAL, DISERO, ZERO, NWDU$, NIO2$, NERR3$

           SEGMENT ZZ8*            035401 035652       066007 066047
           FOLLOWS SEGMENT DDK
                SEGMENT CCC (LONGEST IBANK)(LONGEST DBANK)


EX42-00002*TPF$.PUSHDW
                        $(1)   035401 035652       $(0)   066007 066047
                        $(3)   GTMOPS              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  DMAN1, NERR3$

           SEGMENT BBB*            031673 032445       063034 063434
           FOLLOWS SEGMENT AA


EX42-00002*TPF$.NFRAME
                        $(1)   031673 031702       $(0)   063034 063037
                                                   $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  NERR3$

EX42-00002*TPF$.XNORM1
                        $(1)   031703 031776       $(0)   063040 063061
                        $(3)   PLTPRM              $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  CROSS, MXV, NERR3$

EX42-00002*TPF$.INITEK
                        $(1)   031777 032030       $(0)   063062 063105
                                                   $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:  INITT, DMPBUF, NWDU$, NIO2$, NRDU$, NERR3$

EX42-00002*TPF$.DISPLY
                        $(1)   032031 032445       $(0)   063106 063434
                        $(3)   GTMOPS              $(2)   BLANK$COMMON
                        $(5)   SYSTEM              $(4)   TREEV
                        $(7)   ROTCON              $(6)   UNITS
                        $(011)WINDO                $(010)PLTPRM
                                                   $(012)GTMBUF
     EXTERNAL REFERENCES:  INITEK, XFORM, ERASE, DMPBUF, ZOOMSC, DISCF,
     SETVAL, GETPAN, GTMPLT, XNORM1, MXV, AFILT, NFRAME, CALPLT,
     NERR3$

           SEGMENT BBBA*           031673 031737       063034 063044
           FOLLOWS SEGMENT AA
```

```
EX42-00002*TPF$.PLINIT
                                 $(1)   031673 031737        $(0)   063304 063434
                                 $(3)   PLTPRM               $(2)   BLANK$COMMON
                                                             $(4)   WINDO
      EXTERNAL REFERENCES:   NERR3$

                SEGMENT 33BC*            032446 035623        063435 063762
                FOLLOWS SEGMENT 33B


EX42-00002*TPF$.ZOOMSC
                                 $(1)   032446 032501        $(0)   063435 063445
                                 $(3)   PLTPRM               $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.GETPAN
                                 $(1)   032502 032717        $(0)   063446 063475
                                                             $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   PTPAIR, SETVAL, NERR3$

EX42-00002*TPF$.PTPAIR
                                 $(1)   032720 033602        $(0)   063476 063520
                                                             $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   LSTDAT, SETVAL, NERR2$, NERR3$

EX42-00002*TPF$.XNORM
                                 $(1)   033603 033773        $(0)   063521 063557
                                                             $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   SQRT, SIN, COS, NERR3$

EX42-00002*TPF$.GTMPLT
                                 $(1)   033774 034212        $(0)   063560 063617
                                 $(3)   PLTPRM               $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   PLTGTM, SETVAL, NERR3$

EX42-00002*TPF$.AFILT
                                 $(1)   034213 034552        $(0)   063620 063655
                                                             $(2)   BLANK$COMMON
      EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.PLTGTM
                                 $(1)   034553 035231        $(0)   063656 063711
                                 $(3)   WINDO                $(2)   BLANK$COMMON
                                                             $(4)   SYSTEM
      EXTERNAL REFERENCES:   CALPLT, IGOOD, NERR3$

EX42-00002*TPF$.IGOOD
                                 $(1)   035232 035442        $(0)   063712 063726
                                 $(3)   WINDO                $(2)   BLANK$COMMON
```

```
     EXTERNAL REFERENCES:   NERR3$

EX42-00002*TPF$.CALPLT
                              $(1)   035443 035623        $(0)   063727 063762
                              $(3)   PLTPRM               $(2)   BLANK$COMMON
                              $(5)   UNITS                $(4)   GTMBUF
     EXTERNAL REFERENCES:   PACKWG, DMAN1, DMPBUF, MOVABS, BELL, DRWABS,
     NWDU$, NIO2$, NERR3$

            SEGMENT BBBD*            032446 034214        063435 063615
            FOLLOWS SEGMENT BBB


EX42-00002*TPF$.MAXMIN
                              $(1)   032446 032601        $(0)   063435 063470
                                                          $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   SETVAL, DMAN1, NERR3$

EX42-00002*TPF$.LSTDA1
                              $(1)   032602 033264        $(0)   063471 063513
                                                          $(2)   BLANK$COMMON
     EXTERNAL REFERENCES:   SETVAL, DMAN1, ZERO, NERR2$, NERR3$

EX42-00002*TPF$.DISCF
                              $(1)   033265 033521        $(0)   063514 063550
                              $(3)   ROTCOM               $(2)   BLANK$COMMON
                                                          $(4)   PLTPRM
     EXTERNAL REFERENCES:   MAXVAL, NNJ, NERR3$

EX42-00002*TPF$.MAXVAL
                              $(1)   033522 034214        $(0)   063551 063615
                              $(3)   TREEV                $(2)   BLANK$COMMON
                              $(5)   UNITS                $(4)   SYSTEM
                              $(7)   PLTPRM               $(6)   ROTCOM
                                                          $(010)GTMBUF
     EXTERNAL REFERENCES:   DMAN1, SETVAL, LSTDA1, MAXMIN, NERR2$,
     NERR3$


IBANK DRAWN TO SCALE:     300 WORDS DECIMAL PER DASH

MAIN (10377)
-----------------------------------
                                   AA* (2354)
                                   -------
                                          BBBA* (37)
                                          -
                                   DD* (573)
                                   --
                                          TAK* (144)
```

BB* (252)

CC* (366)

FF* (740)

GG* (427)

CCC* (1362)

ZZ3* (170)

DDO* (76)

DDF* (244)

DDM* (72)

DDN* (72)

ZZ4* (1189)

DDY* (210)

AAA* (313)

BBB* (363)

BBBC* (1646)

BBBD* (871)

ZZ2* (150)

CCA* (60)

DBA* (326)

ABE* (1162)

ABD* (54)

ABC* (69)

ABB* (72)

ABA* (72)

AAZ* (86)

ORIGINAL PAGE IS
OF POOR QUALITY

- AAY* (56)
- AAX* (34)
- AAW* (94)
- AAO* (165)
- AAN* (132)
- AAM* (93)
- AAL* (95)
- AAJ* (95)
- AAI* (183)
- AAH* (183)
- AAG* (193)
- AAF* (128)
- AAE* (46)
- AAD* (51)
- AAC* (134)
- AAB* (91)

DDL* (659)

- DDJ* (93)
- DDI* (232)

DDH* (169)

DDG* (174)

DDF* (171)

DDE* (642)

- DDZ* (63)
- DDD* (63)

DDC* (113)

DDB* (68)

DDA* (690)

DBANK DRAWN TO SCALE: 200 WORDS DECIMAL PER DASH

MAIN (7147)

AA* (2017)

BBBA* (9)

DD* (1571)

DDK* (43)

BB* (1568)

CC* (1734)

FF* (3156)

GG* (1601)

CCC* (376)

ZZ3* (33)

DDD* (24)

DDP* (39)

DDN* (25)

DDM* (25)

ZZ4* (201)

DDY* (26)

AAA* (39)

BBB* (257)

BBBC* (214)

BBBD* (113)

ZZ2* (32)

CCA* (20)

ZBA* (116)

ABE* (127)

ABD* (24)

ABC* (21)

ABB* (25)

ABA* (25)

AAZ* (25)

AAY* (23)

AAX* (32)

AAW* (41)

AAO* (47)

AAN* (45)

AAM* (29)

AAL* (30)

AAJ* (29)

AAI* (43)

AAH* (42)

AAG* (69)

AAF* (41)

AAE* (8)

AAD* (12)

AAC* (50)

AAB* (40)

DDL* (104)

[illegible]

```
[illegible]
DDI* (40)
DDH* (28)
DDG* (34)
DDF* (62)
DDE* (145)
DDZ* (10)
DDD* (10)
DDC* (46)
DDB* (16)
DDA* (107)
```

REFERENCES

1. Gentry, A.: Hypersonic Arbitrary Body Aerodynamic Program. Douglas Report. DAC56080. June 1967.

APPENDIX A -

FLOWCHARTS OF SELECTED SUBROUTINES

This appendix contains automatically generated flowcharts of selected subroutines from GTM. A computer program FLOGEN was used to generate the flowcharts. Additional flowcharts may be generated using the following procedure:

```
@RUN -----

@COPY,S GTM.,TPF$.

@MAP,N FM9*FLOWGEN.FLOWGM,TPF$.

@XQT

 INPUT

 IREAD=1, ISPEC=2

 $END
```

The above procedure will produce microfilm output of all subroutines in the GTM.

```
SUBROUTINE    DMAN (IPP,IT,MK,IDATA,IBUF,IV,IUNDAT)
```

```
C $NOTE(  CALLING PARAMETERS)
```

```
DIMENSION IT(1),IDATA(1),IBUF(1)
DIMENSION IUNDAT(1)
DIMENSION IBF(7),LBF(7)
COMMON/MS/NT,KEYWRD,NREC,LENGTH,INCLEN
EQUIVALENCE (KW,IBF(3))
IOP=IPP
M=MK
```

```
C  TEMPORARY DEFINITON OF  IUN **********0*0*0*0*0*0*0*
```

```
IRTN=0
```

```
C       IUN--DISC UNIT DEDICATED TO MS STORAGE
C       IOP --OPERATION CODE
C           =5HCLEAR--THIS IS USED AFTER A FILE HAS BEEN COMPLETED
C                     USING CODES 20,21,30,31.
C           =5HCLOSE--THIS IS USED TO CLOSE THE LIBRARY SO THAT IT
C                     MAY BE PICKED UP BY A SUBSEQUENT JOB STEP
C           =10HPURGE--REMOVE THIS FILE FORM THE LIST OF RETRIEVABLE
C                      DATA FILES
```

```
C           =+N --WRITE
C           =-N --READ
C           =10HTAPEINPUT
C           =10HTAPEOUTPUT
C                   PERMANENT STORAGE OF MS DATA--SEE INSTRUCTIONS
C                   BELOW
C             WRITE CODES
C               N=10  DATA IS COMPLETE IN IDATA(MATRIX STORE)
```

```
C               N=20  WRITE A PARTIAL FILE--FIXED LENGTH RECORDS
C               N=21  WRITE A PARTIAL FILE--VARIABLE LENGTH RECORDS
C               N=30  EXTEND A FILE--FIXED LENGTH RECORDS
C               N=31  EXTEND A FILE--VARIABLE LENGTH RECORDS
C       N--THE NUMBER OF WORDS IN THE DATA TITLE
C       IT--AN ARRAY CONTAINING THE TITLE--IT MUST BE DIMENSIONED N+1
C       M--THE NUMBER OF WORDS IN THE DATA RECORD STORED IN IDATA
C       IDATA--AN ARRAY CONTAINING THE DATA RECORD
```

CONT. ON PG 2

```
C       IBUF --THE BUFFER TO USE FOR THIS FILE
C       NBUF --THE LENGTH OF THE BUFFER
C  - - -- - - - PERMANENT STORAGE OF MS DATA
C        N=TAPE UNIT ON WHICK TO WRITE TAPE
C       IT= A WORKING ARRAY LARGE ENOUGH FOR THE LONGEST TITLE
C             IN THE STORED MS DATA
C       IDATA= A WORKING ARRAY LARGE ENOUGH TO ACCOMODATE THE LARGEST
C             BUFFER USED TO WRITE THE MS TAPE.
```

```
C                 NOTE--CLOSE MUST BE EXECUTED PRIOR TO WRITING A
C                       TAPE
```

```
IFLG=0
```

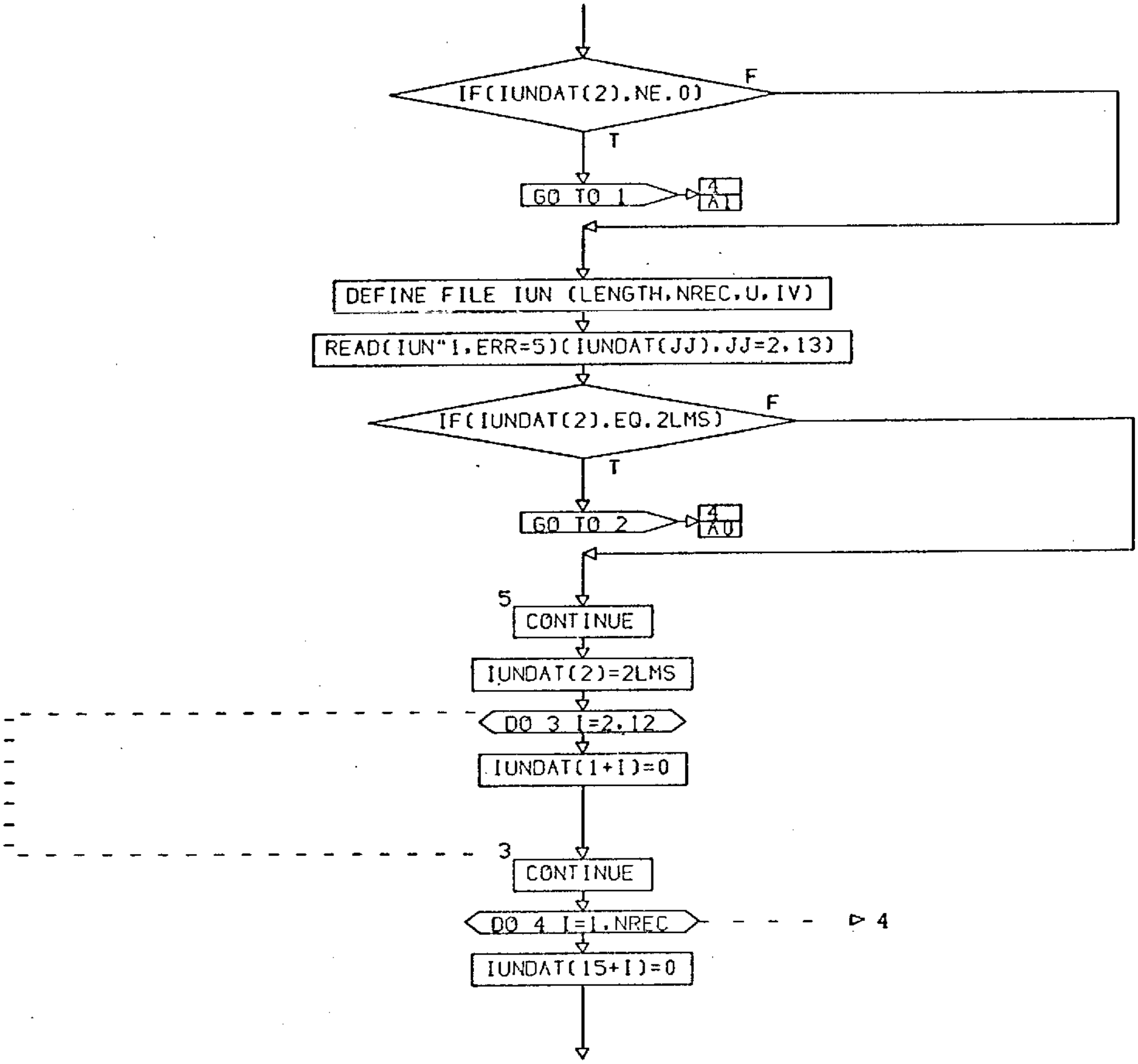
```
C        FIRST  OPERATION --OPEN MS AND ESTABLISH INDEXES
C       THIS SECTION IS TEMPORARY AND WILL BE MOVED TO A NEW PLACE IN
C       THE GAC
C   IUNDAT(1)=IUN-------UNIT NUMBER
C   IUNDAT(2)
C     TO
C   IUNDAT(12)-----KDX ARRAY
C   IUNDAT(13)=MDX
```

```
C   IUNDAT(14)=KCR
C   IUNDAT(15)=KFLG
C   IUNDAT(16)
C     TO
C   IUNDAT(NREC+15)----INDX ARRAY
C   NOTE-- INITIALIZE IUNDAT TO 0
C            NT TO 3
C            KEYWRD TO 2
```

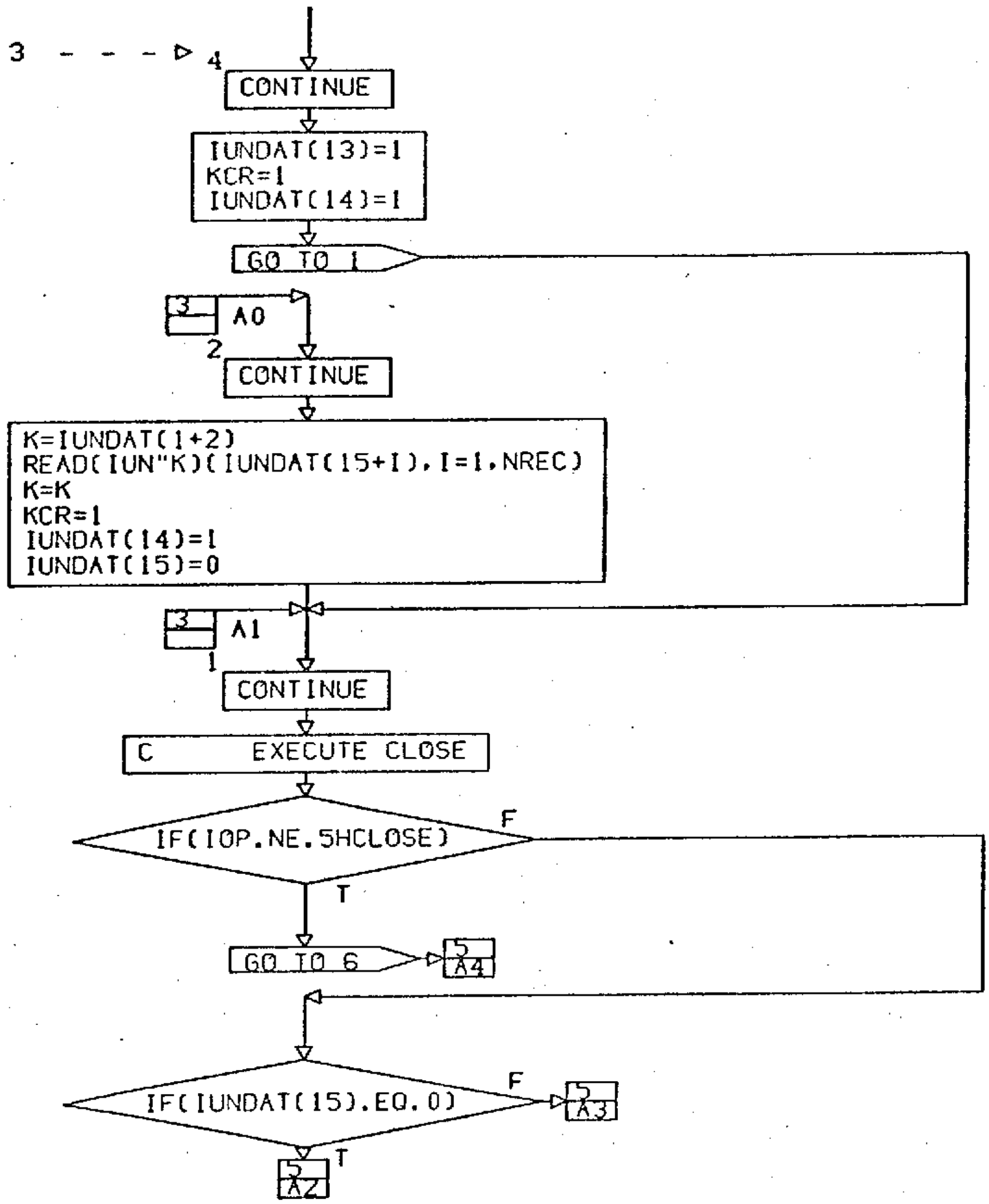
```
C           NREC TO 256
C           LENGTH TO +   (200)
C           INCLEN TO 50
```

```
INEW=0
IUN=IUNDAT(1)
KCR=IUNDAT(14)
```

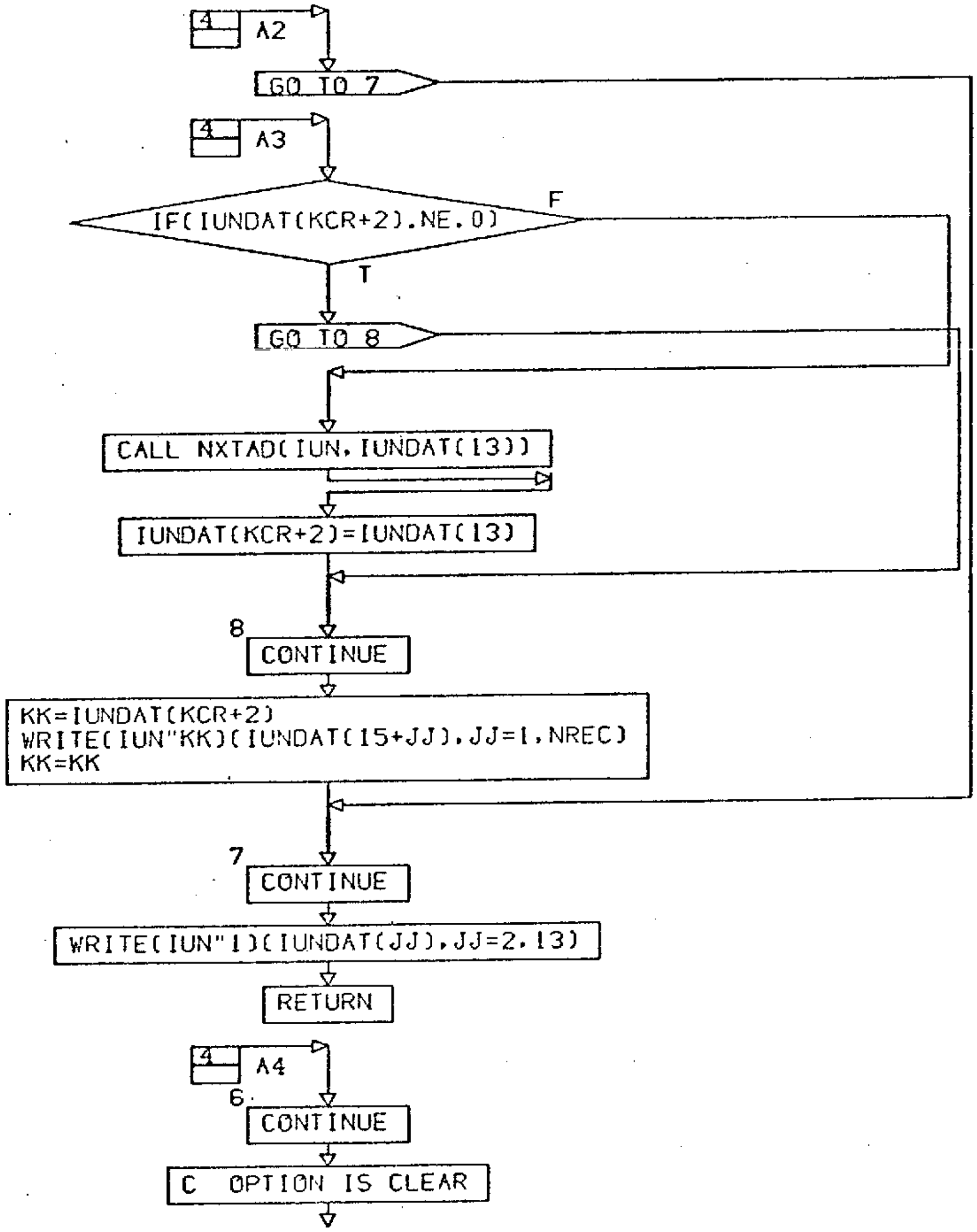
CONT. ON PG 3


IF(IUNDAT(2).NE.0)
F
T
GO TO 1
4
A1
DEFINE FILE IUN (LENGTH,NREC,U,IV)
READ(IUN'1,ERR=5)(IUNDAT(JJ),JJ=2,13)
IF(IUNDAT(2).EQ.2LMS)
F
T
GO TO 2
4
A0
5
CONTINUE
IUNDAT(2)=2LMS
DO 3 I=2,12
IUNDAT(1+I)=0
3
CONTINUE
DO 4 I=1,NREC
4
IUNDAT(15+I)=0

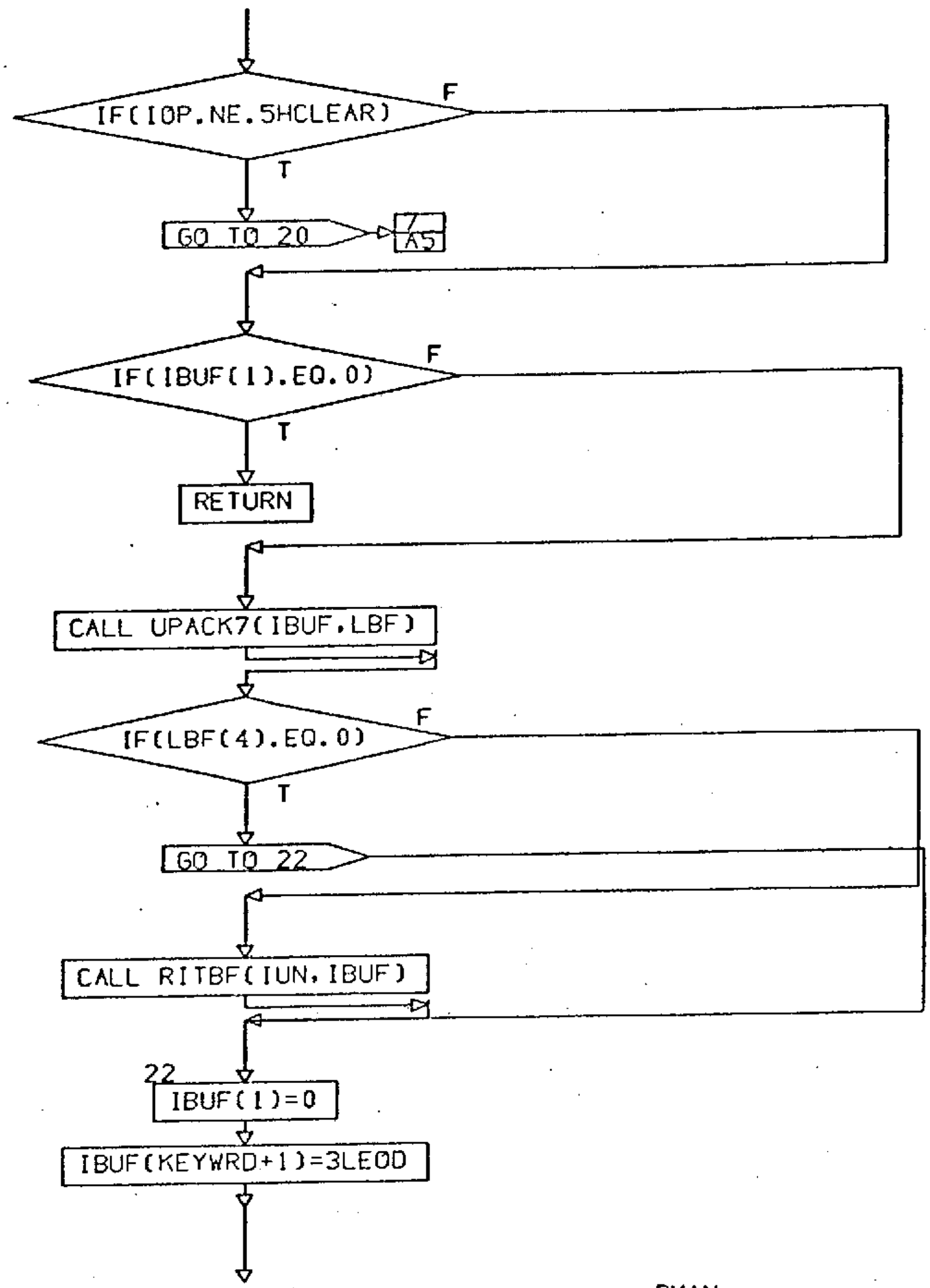
CONT. ON PG 4


3
4
CONTINUE
IUNDAT(13)=1
KCR=1
IUNDAT(14)=1
GO TO 1
3
A0
2
CONTINUE
K=IUNDAT(1+2)
READ(IUN"K)(IUNDAT(15+I),I=1,NREC)
K=K
KCR=1
IUNDAT(14)=1
IUNDAT(15)=0
3
A1
1
CONTINUE
C EXECUTE CLOSE
IF(IOP.NE.5HCLOSE)
F
T
GO TO 6
5
A4
IF(IUNDAT(15).EQ.0)
F
5
A3
T
5
A2

CONT. ON PG 5

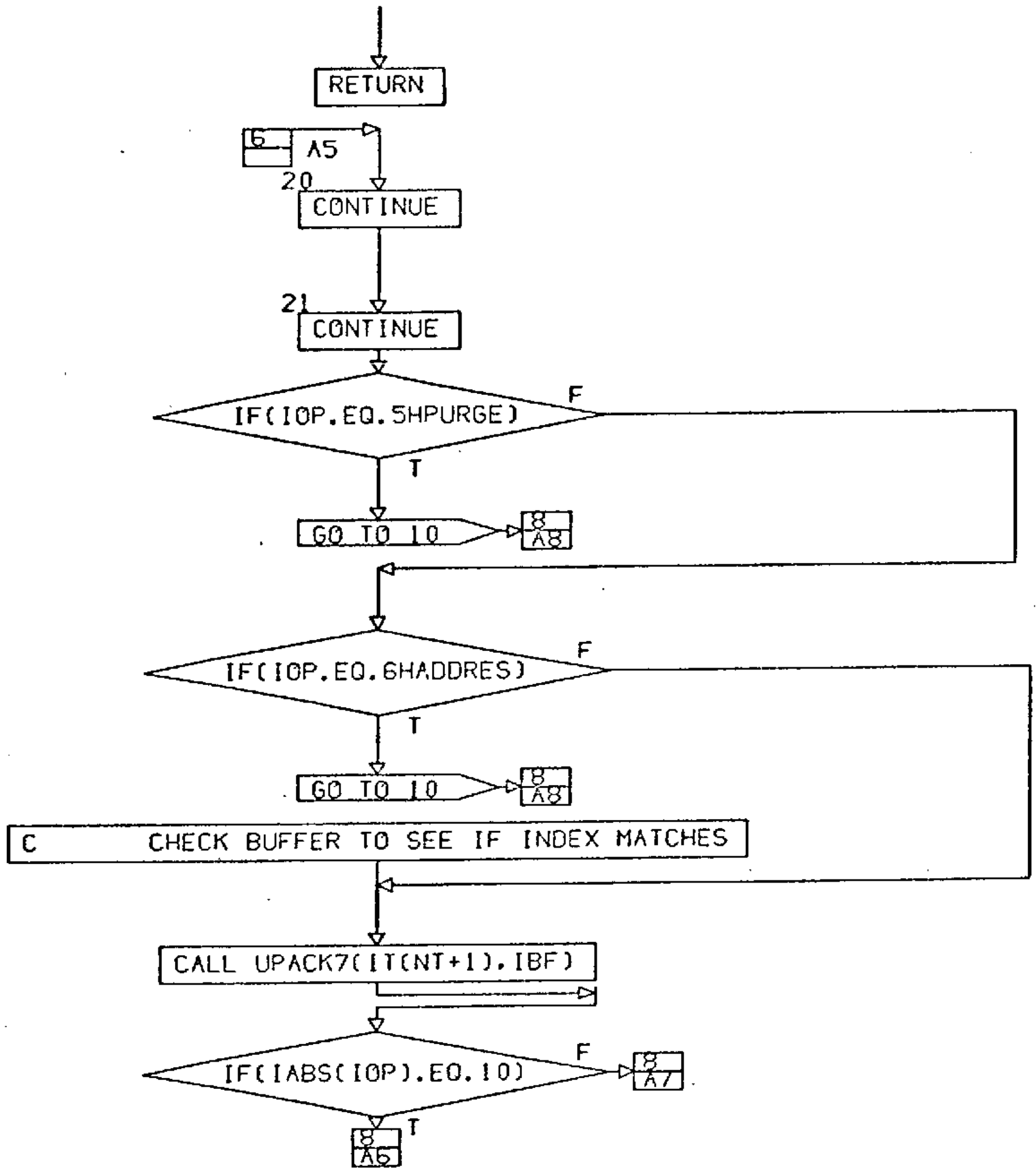

A2
GO TO 7
A3
IF(IUNDAT(KCR+2).NE.0)
F
T
GO TO 8
CALL NXTAD(IUN,IUNDAT(13))
IUNDAT(KCR+2)=IUNDAT(13)
8
CONTINUE
KK=IUNDAT(KCR+2)
WRITE(IUN"KK)(IUNDAT(15+JJ),JJ=1,NREC)
KK=KK
7
CONTINUE
WRITE(IUN"1)(IUNDAT(JJ),JJ=2,13)
RETURN
A4
6
CONTINUE
C OPTION IS CLEAR

CONT. ON PG 6



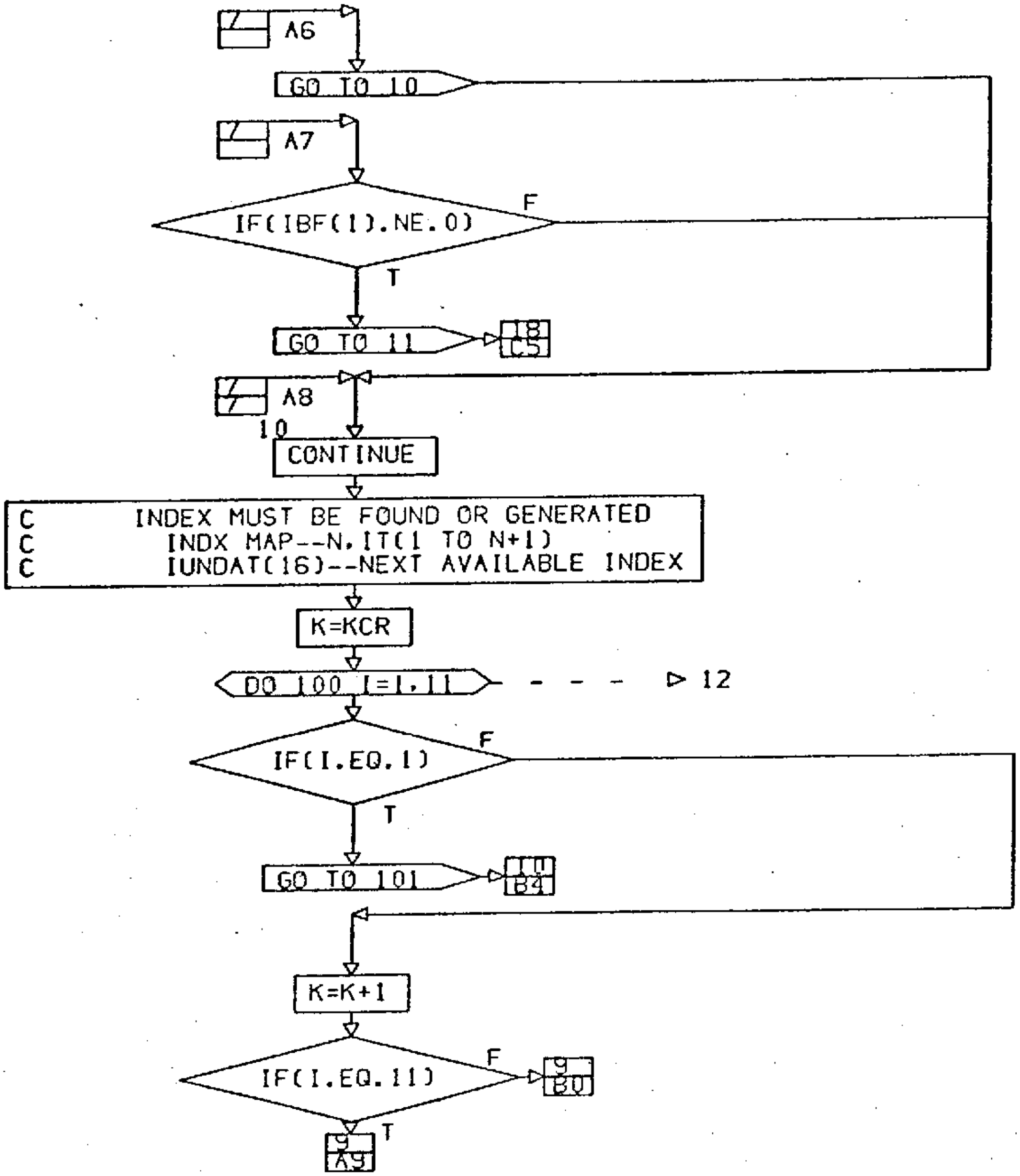
IF(IOP.NE.5HCLEAR)
F
T
GO TO 20
7
A5
IF(IBUF(1).EQ.0)
F
T
RETURN
CALL UPACK7(IBUF,LBF)
IF(LBF(4).EQ.0)
F
T
GO TO 22
CALL RITBF(IUN,IBUF)
22
IBUF(1)=0
IBUF(KEYWRD+1)=3LEOD

CONT. ON PG 7

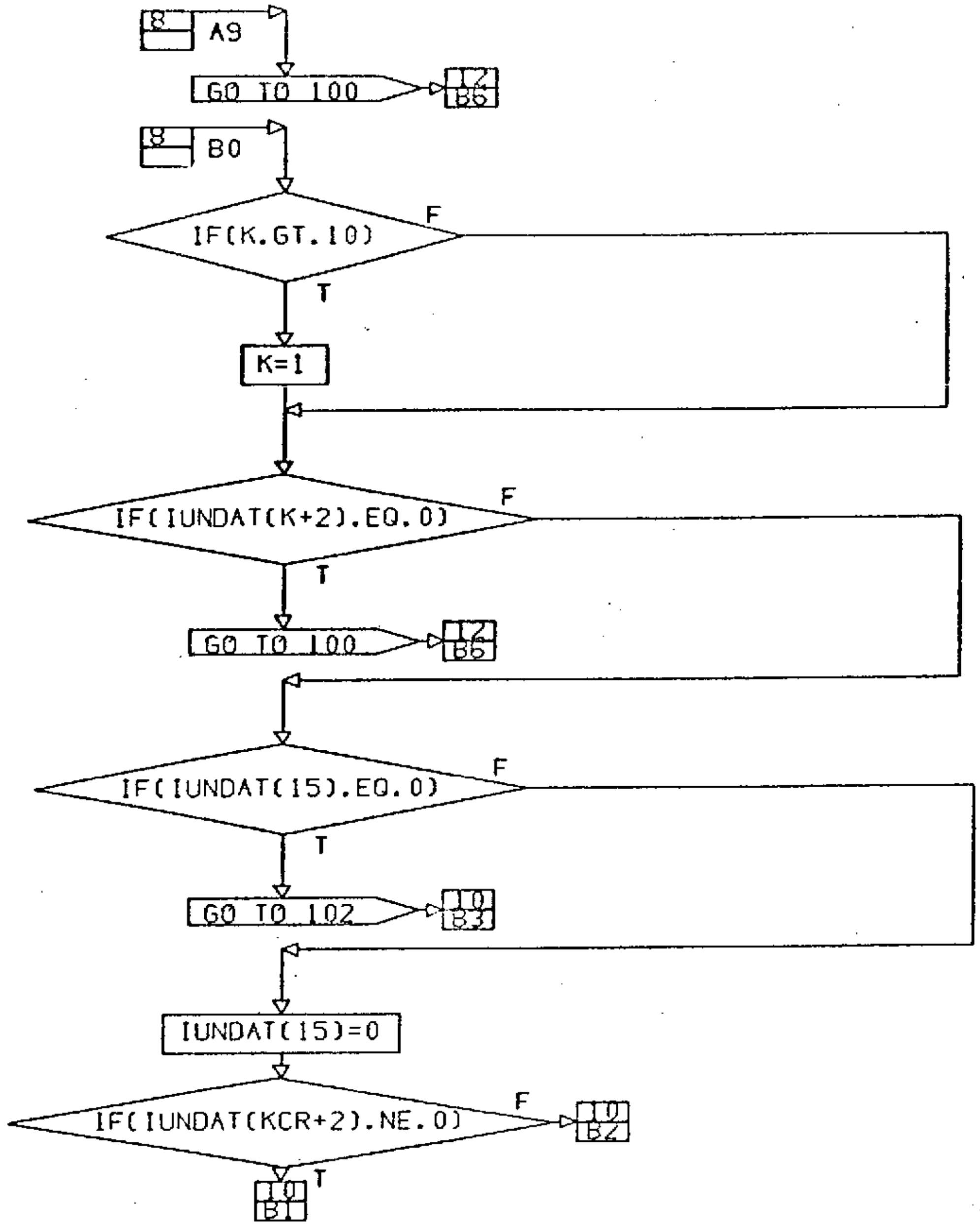


RETURN
6
A5
20
CONTINUE
21
CONTINUE
IF(IOP.EQ.5HPURGE)
F
T
GO TO 10
8
A8
IF(IOP.EQ.6HADDRES)
F
T
GO TO 10
8
A8
C CHECK BUFFER TO SEE IF INDEX MATCHES
CALL UPACK7(IT(NT+1),IBF)
IF(IABS(IOP).EQ.10)
F
8
A7
T
8
A6

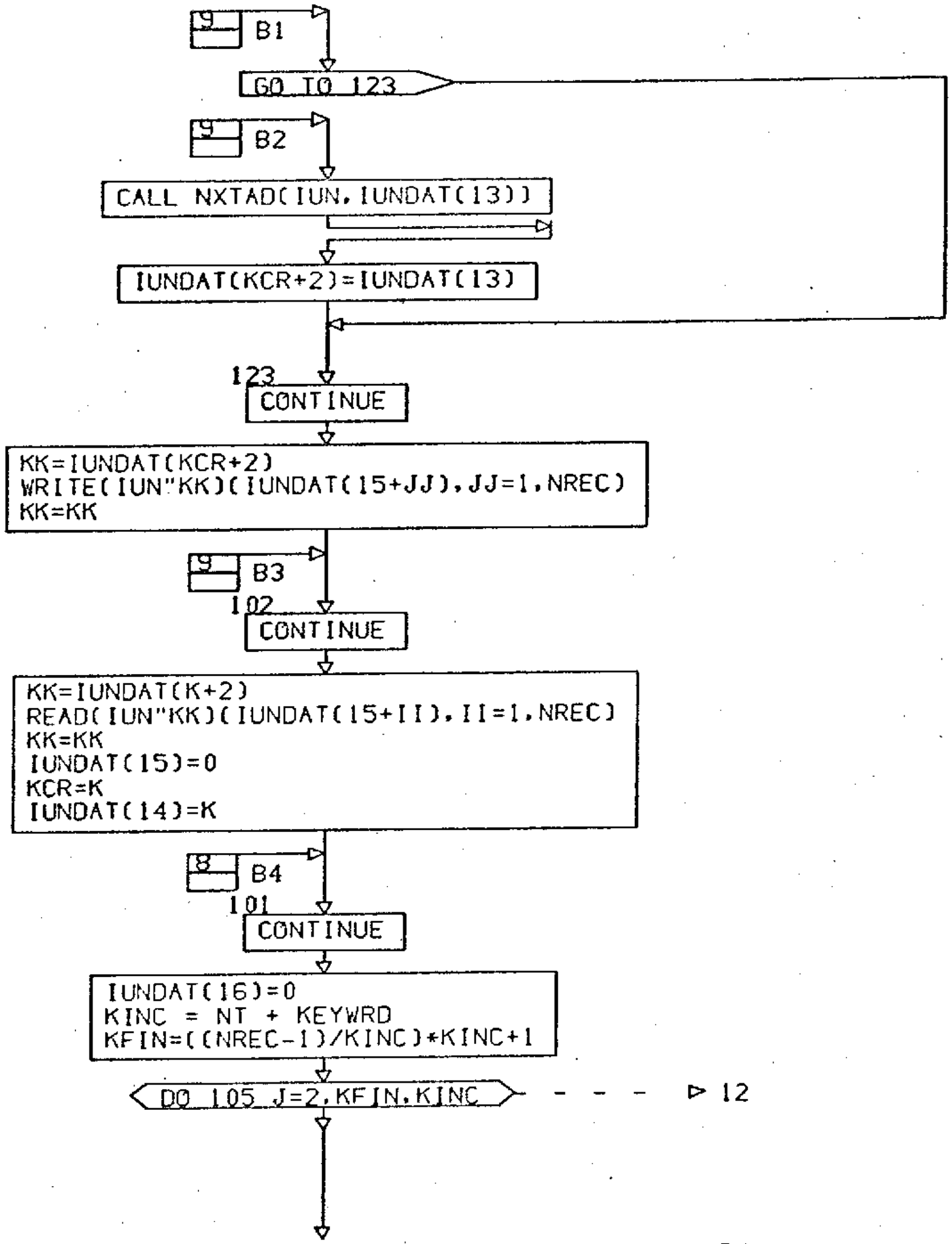
CONT. ON PG 8


A6
GO TO 10
A7
IF(IBF(1).NE.0)
F
T
GO TO 11
18
C5
A8
10
CONTINUE
C INDEX MUST BE FOUND OR GENERATED
C INDX MAP--N,IT(1 TO N+1)
C IUNDAT(16)--NEXT AVAILABLE INDEX
K=KCR
DO 100 I=1,11
12
IF(I.EQ.1)
F
T
GO TO 101
B4
K=K+1
IF(I.EQ.11)
F
9
B0
T
9
A9

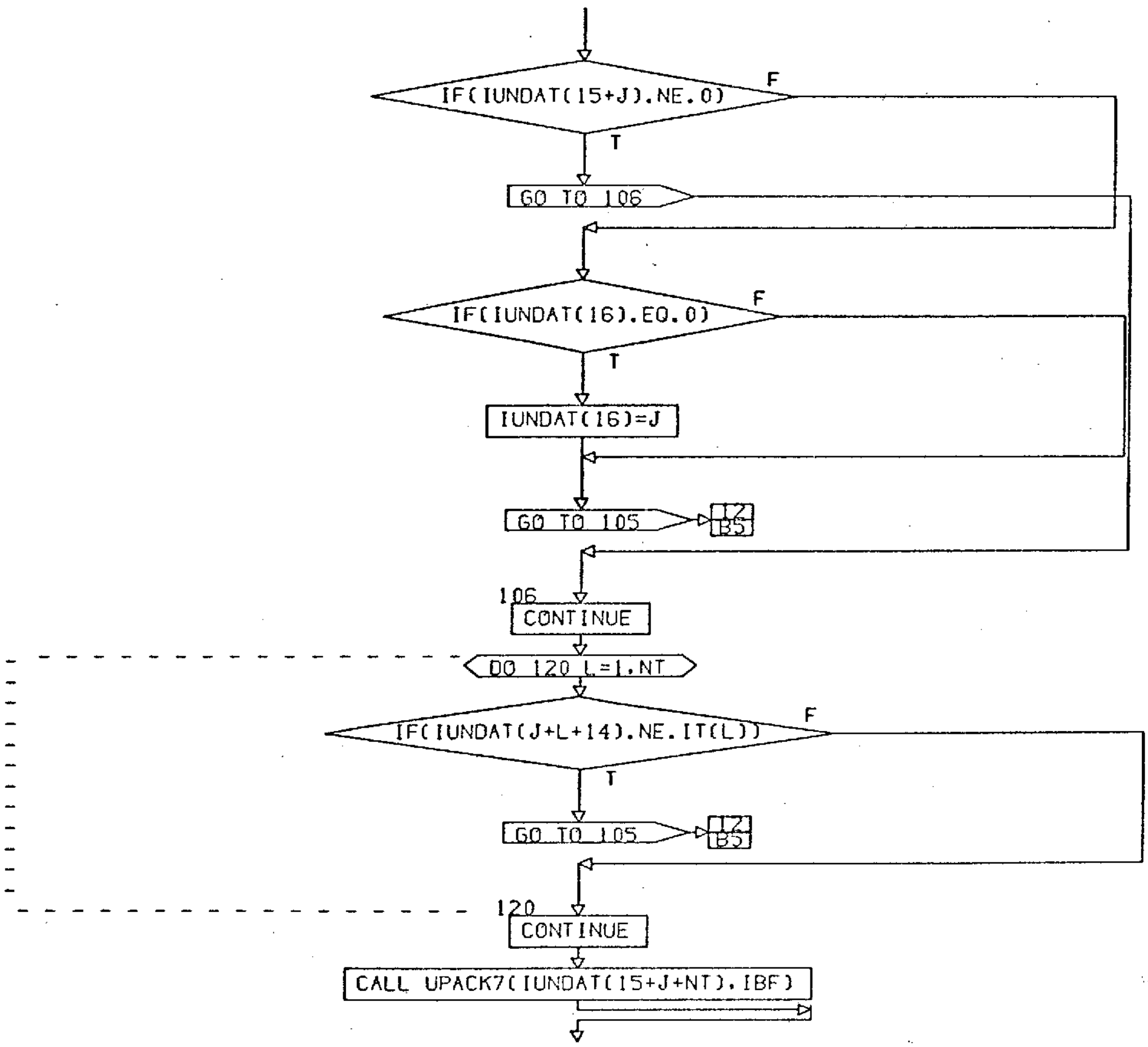
CONT. ON PG 9


8
A9
GO TO 100
12
B6
8
B0
IF(K.GT.10)
F
T
K=1
IF(IUNDAT(K+2).EQ.0)
F
T
GO TO 100
12
B6
IF(IUNDAT(15).EQ.0)
F
T
GO TO 102
10
B3
IUNDAT(15)=0
IF(IUNDAT(KCR+2).NE.0)
F
10
B2
T
10
B1

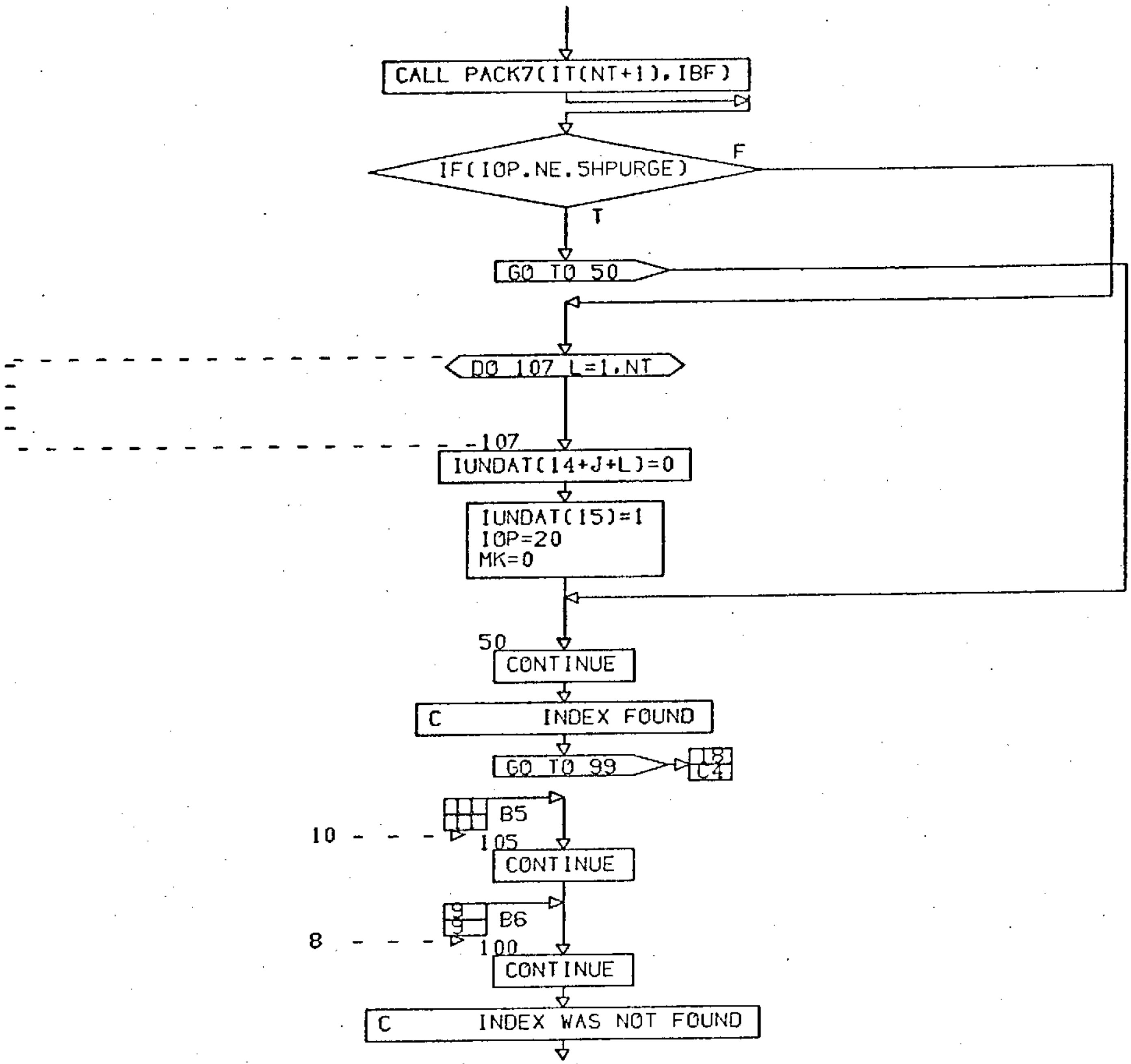
CONT. ON PG 10


9 B1
GO TO 123
9 B2
CALL NXTAD(IUN,IUNDAT(13))
IUNDAT(KCR+2)=IUNDAT(13)
123 CONTINUE
KK=IUNDAT(KCR+2)
WRITE(IUN"KK)(IUNDAT(15+JJ),JJ=1,NREC)
KK=KK
9 B3
102 CONTINUE
KK=IUNDAT(K+2)
READ(IUN"KK)(IUNDAT(15+II),II=1,NREC)
KK=KK
IUNDAT(15)=0
KCR=K
IUNDAT(14)=K
8 B4
101 CONTINUE
IUNDAT(16)=0
KINC = NT + KEYWRD
KFIN=((NREC-1)/KINC)*KINC+1
DO 105 J=2,KFIN,KINC
▷ 12

CONT. ON PG 11

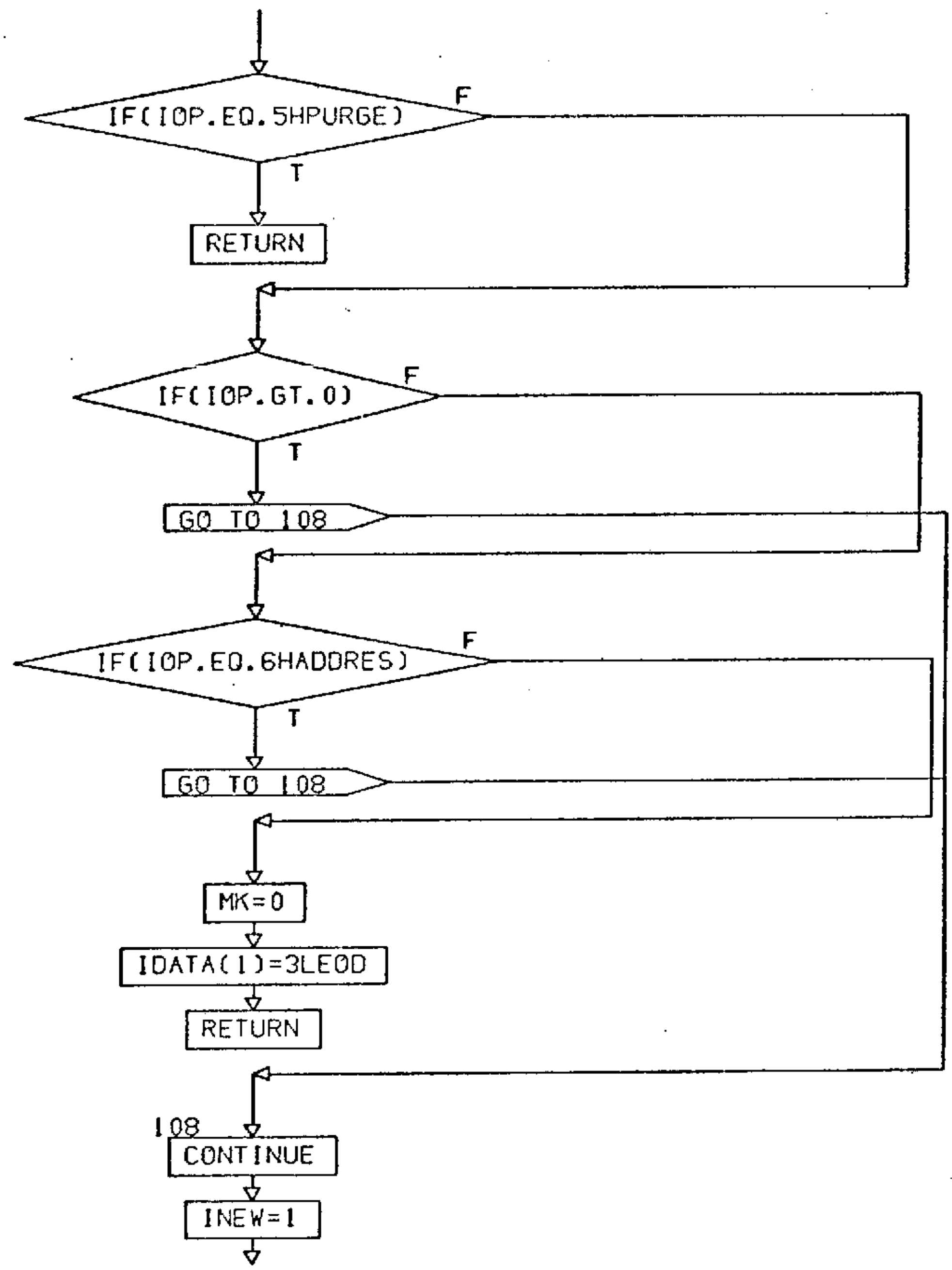

IF(IUNDAT(15+J).NE.0)
F
T
GO TO 106
IF(IUNDAT(16).EQ.0)
F
T
IUNDAT(16)=J
GO TO 105
12
B5
106
CONTINUE
DO 120 L=1,NT
IF(IUNDAT(J+L+14).NE.IT(L))
F
T
GO TO 105
12
B5
120
CONTINUE
CALL UPACK7(IUNDAT(15+J+NT),IBF)

CONT. ON PG 12

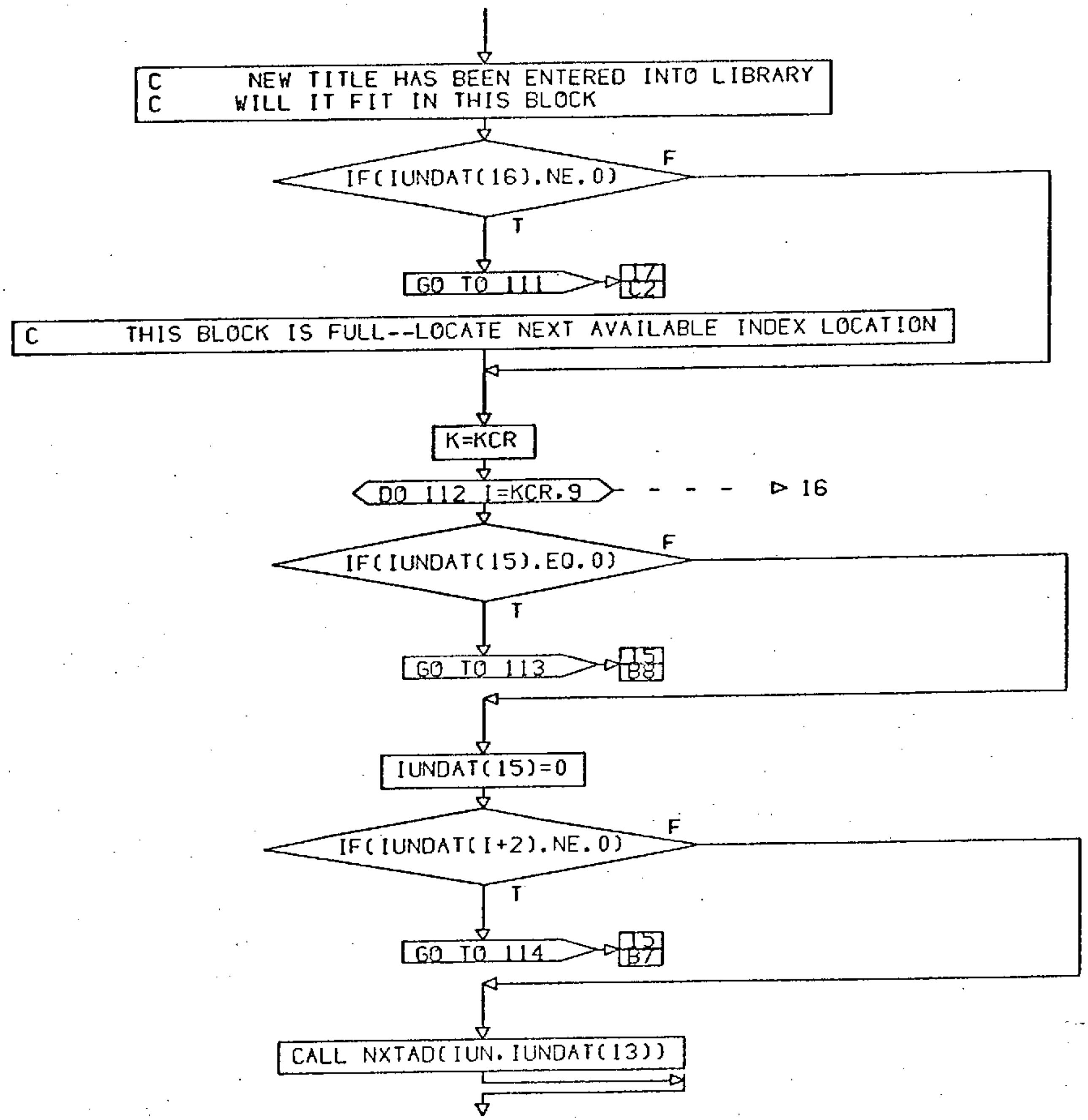


CALL PACK7(IT(NT+1),IBF)
IF(IOP.NE.5HPURGE)
F
T
GO TO 50
DO 107 L=1,NT
107
IUNDAT(14+J+L)=0
IUNDAT(15)=1
IOP=20
MK=0
50
CONTINUE
C INDEX FOUND
GO TO 99
C4
B5
10
105
CONTINUE
B6
8
100
CONTINUE
C INDEX WAS NOT FOUND

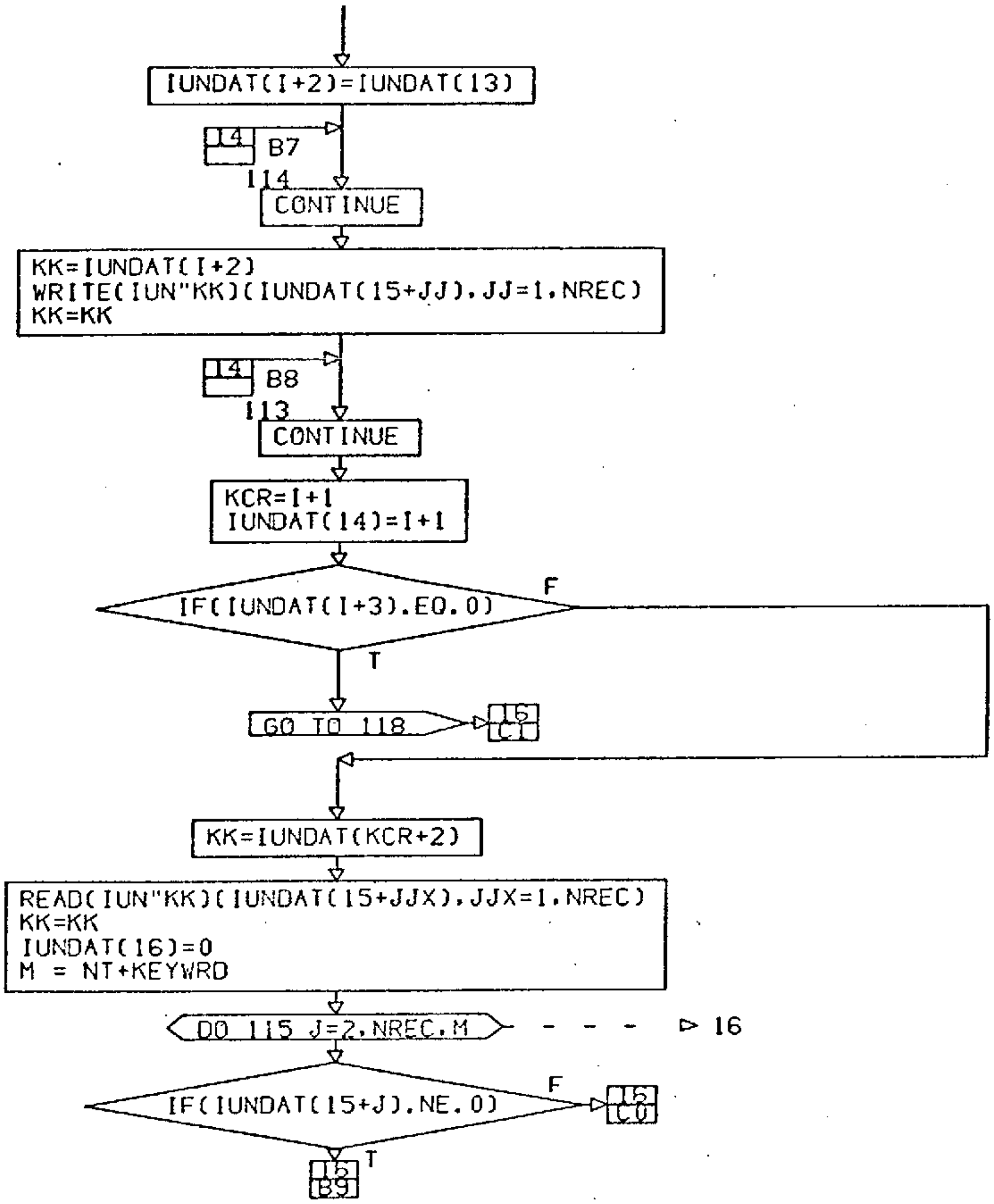
CONT. ON PG 13


IF(IOP.EQ.5HPURGE)
F
T
RETURN
IF(IOP.GT.0)
F
T
GO TO 108
IF(IOP.EQ.6HADDRES)
F
T
GO TO 108
MK=0
IDATA(1)=3LEOD
RETURN
108
CONTINUE
INEW=1

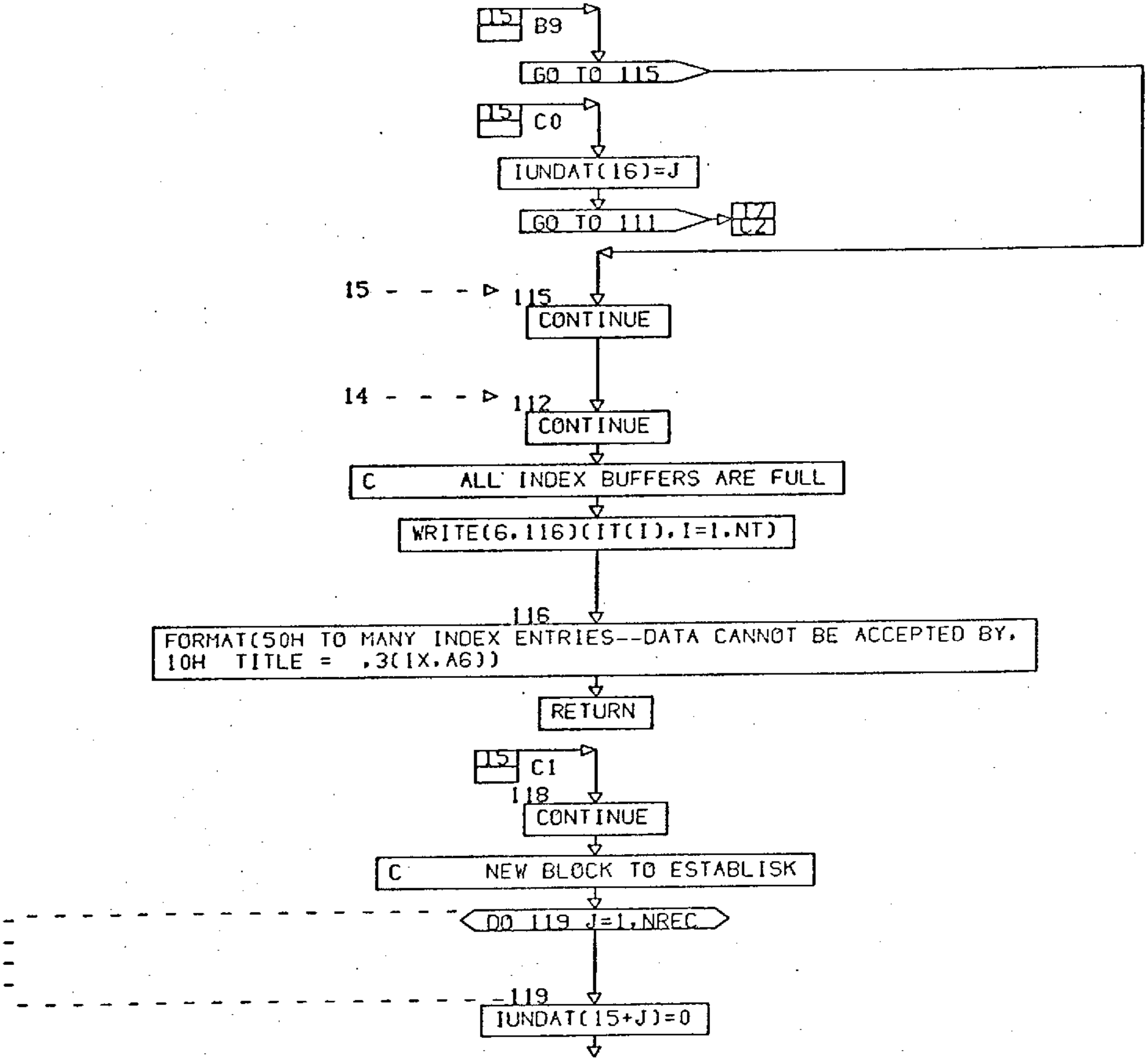
CONT. ON PG 14


C NEW TITLE HAS BEEN ENTERED INTO LIBRARY
C WILL IT FIT IN THIS BLOCK
IF(IUNDAT(16).NE.0)
F
T
GO TO 111
17
C2
C THIS BLOCK IS FULL--LOCATE NEXT AVAILABLE INDEX LOCATION
K=KCR
DO 112 I=KCR,9
16
IF(IUNDAT(15).EQ.0)
F
T
GO TO 113
15
B8
IUNDAT(15)=0
IF(IUNDAT(I+2).NE.0)
F
T
GO TO 114
15
B7
CALL NXTAD(IUN,IUNDAT(13))

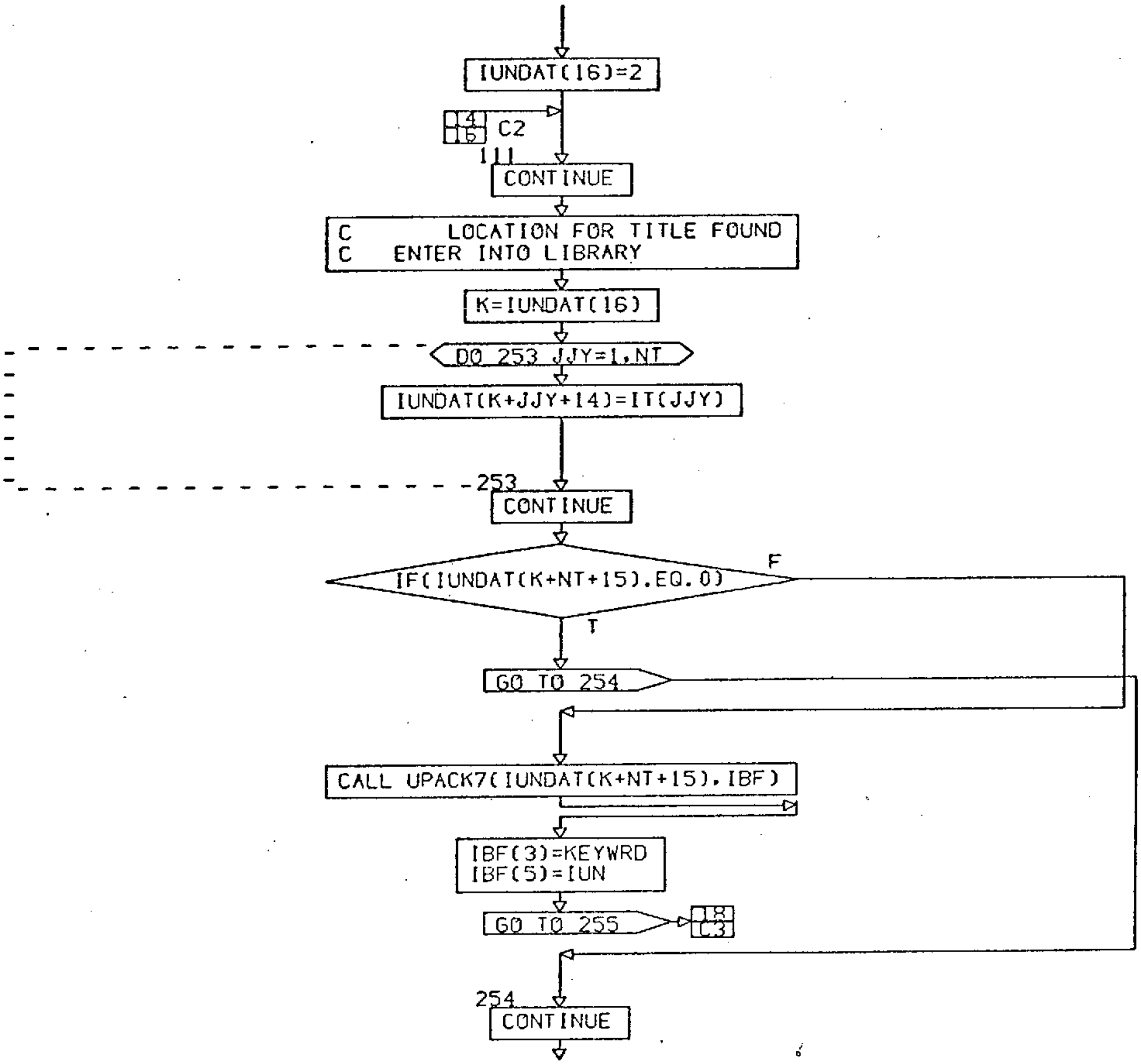
CONT. ON PG 15


IUNDAT(I+2)=IUNDAT(13)
14 B7
114
CONTINUE
KK=IUNDAT(I+2)
WRITE(IUN"KK)(IUNDAT(15+JJ),JJ=1,NREC)
KK=KK
14 B8
113
CONTINUE
KCR=I+1
IUNDAT(14)=I+1
IF(IUNDAT(I+3).EQ.0)
F
T
GO TO 118
16 C1
KK=IUNDAT(KCR+2)
READ(IUN"KK)(IUNDAT(15+JJX),JJX=1,NREC)
KK=KK
IUNDAT(16)=0
M = NT+KEYWRD
DO 115 J=2,NREC,M
16
IF(IUNDAT(15+J).NE.0)
F
16 C0
T
16 B9

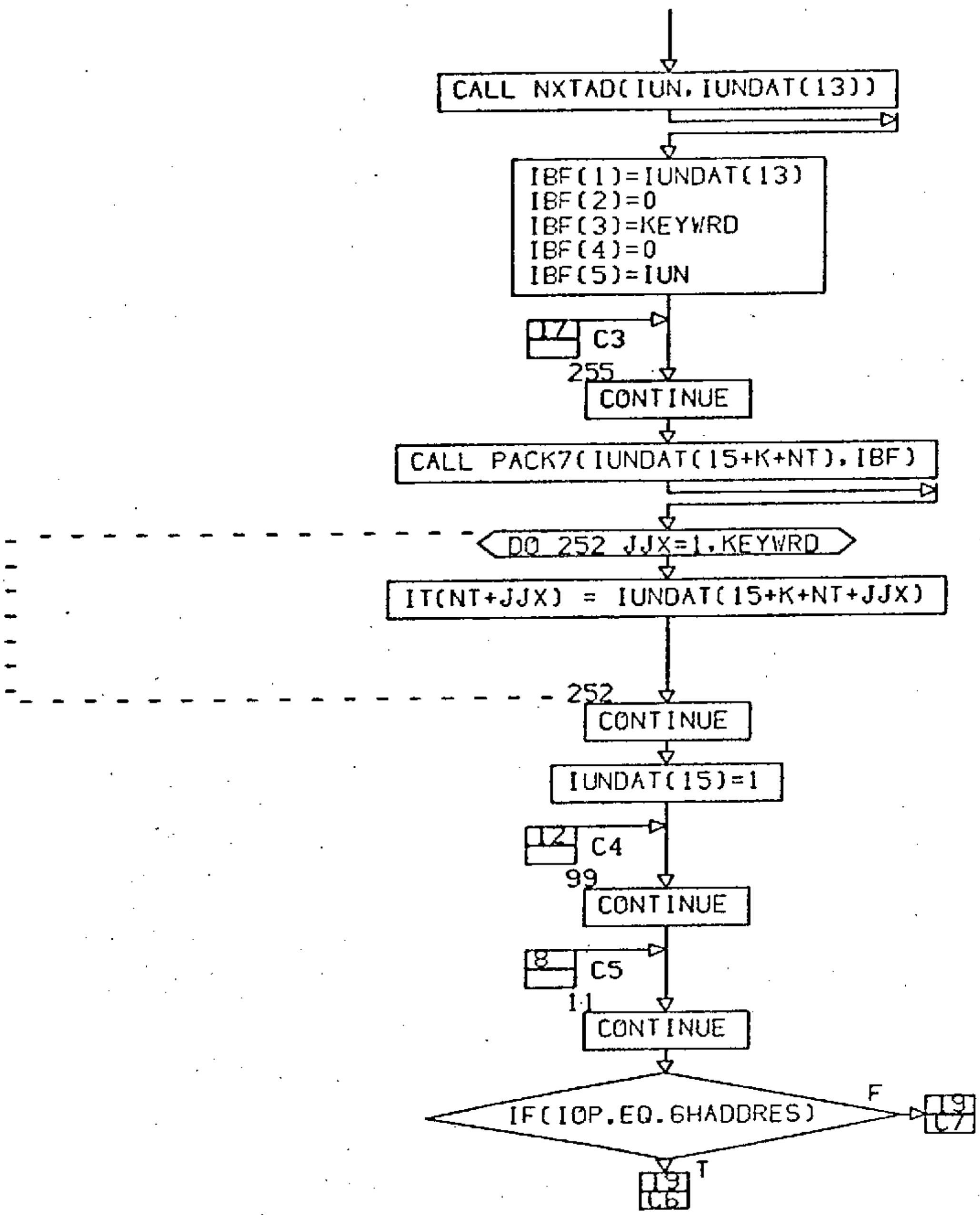
CONT. ON PG 16


15 B9
GO TO 115
15 C0
IUNDAT(16)=J
GO TO 111
17 C2
15
115
CONTINUE
14
112
CONTINUE
C ALL INDEX BUFFERS ARE FULL
WRITE(6,116)(IT(I),I=1,NT)
116
FORMAT(50H TO MANY INDEX ENTRIES--DATA CANNOT BE ACCEPTED BY,
10H TITLE = ,3(1X,A6))
RETURN
15 C1
118
CONTINUE
C NEW BLOCK TO ESTABLISK
DO 119 J=1,NREC
119
IUNDAT(15+J)=0

CONT. ON PG 17

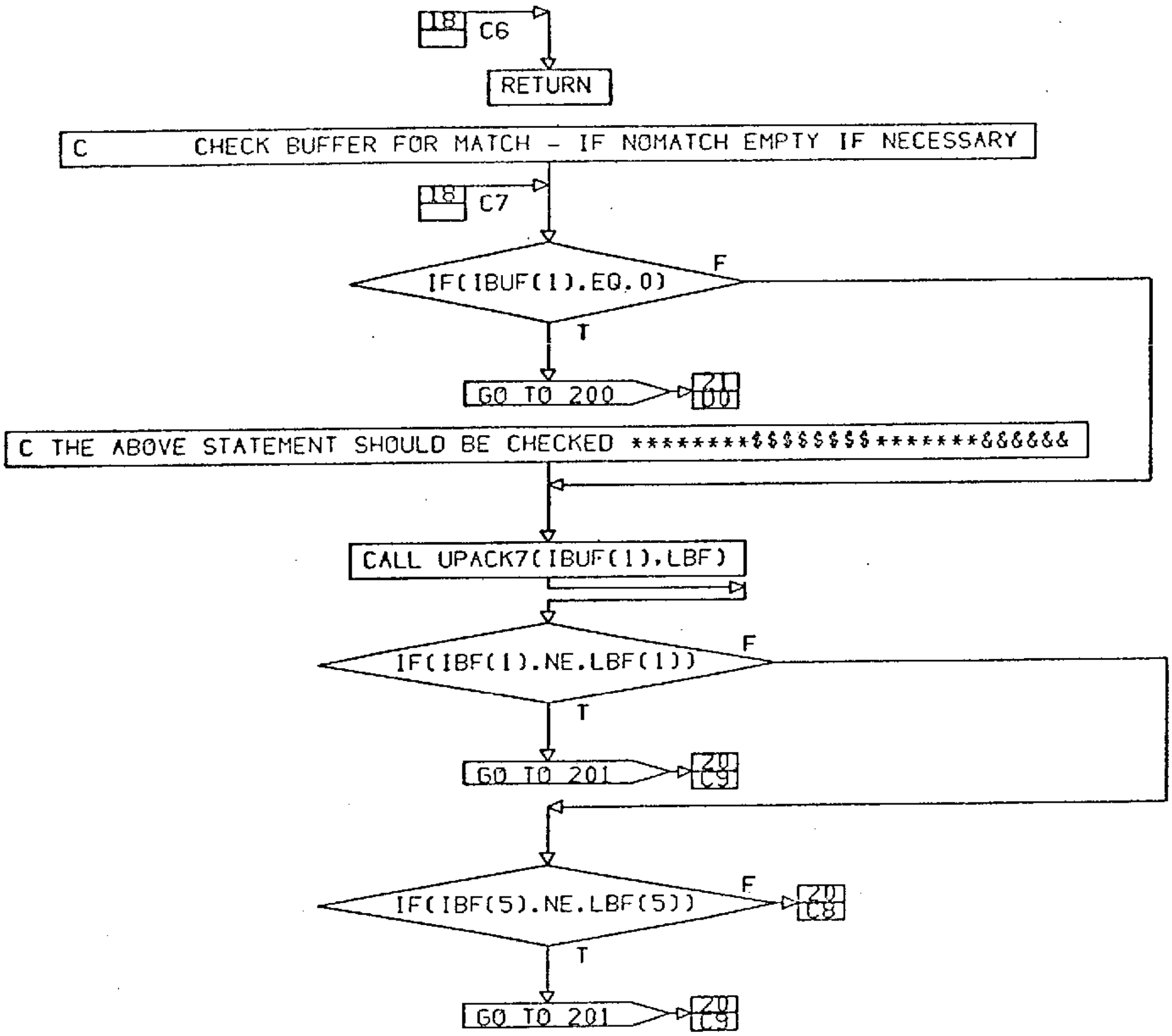

IUNDAT(16)=2
4
6
C2
111
CONTINUE
C LOCATION FOR TITLE FOUND
C ENTER INTO LIBRARY
K=IUNDAT(16)
DO 253 JJY=1,NT
IUNDAT(K+JJY+14)=IT(JJY)
253
CONTINUE
IF(IUNDAT(K+NT+15).EQ.0)
F
T
GO TO 254
CALL UPACK7(IUNDAT(K+NT+15),IBF)
IBF(3)=KEYWRD
IBF(5)=IUN
GO TO 255
18
C3
254
CONTINUE

CONT. ON PG 18

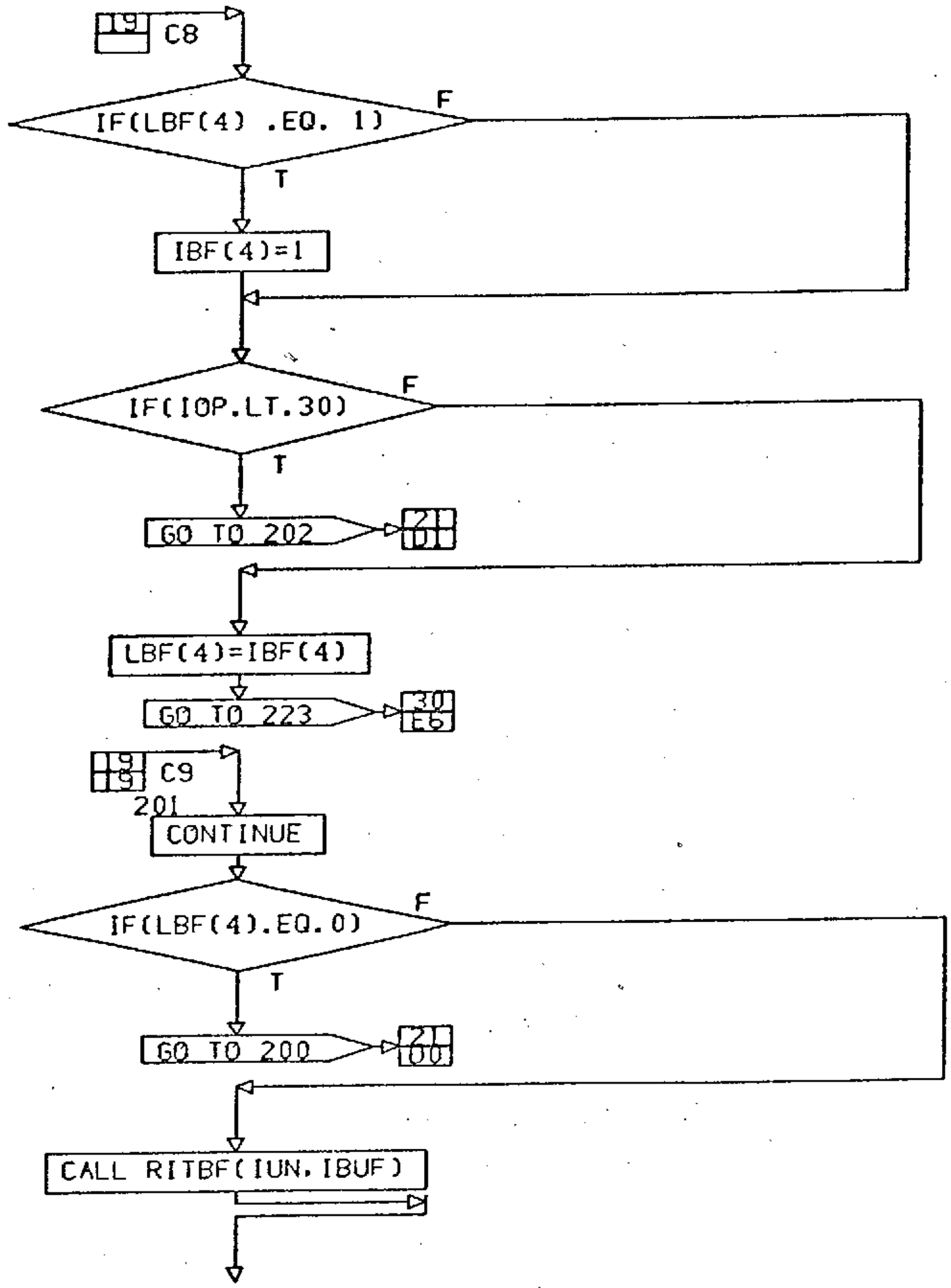


CALL NXTAD(IUN,IUNDAT(13))
IBF(1)=IUNDAT(13)
IBF(2)=0
IBF(3)=KEYWRD
IBF(4)=0
IBF(5)=IUN
17
C3
255
CONTINUE
CALL PACK7(IUNDAT(15+K+NT),IBF)
DO 252 JJX=1,KEYWRD
IT(NT+JJX) = IUNDAT(15+K+NT+JJX)
252
CONTINUE
IUNDAT(15)=1
12
C4
99
CONTINUE
8
C5
11
CONTINUE
IF(IOP.EQ.6HADDRES)
F
19
C7
T
19
C6

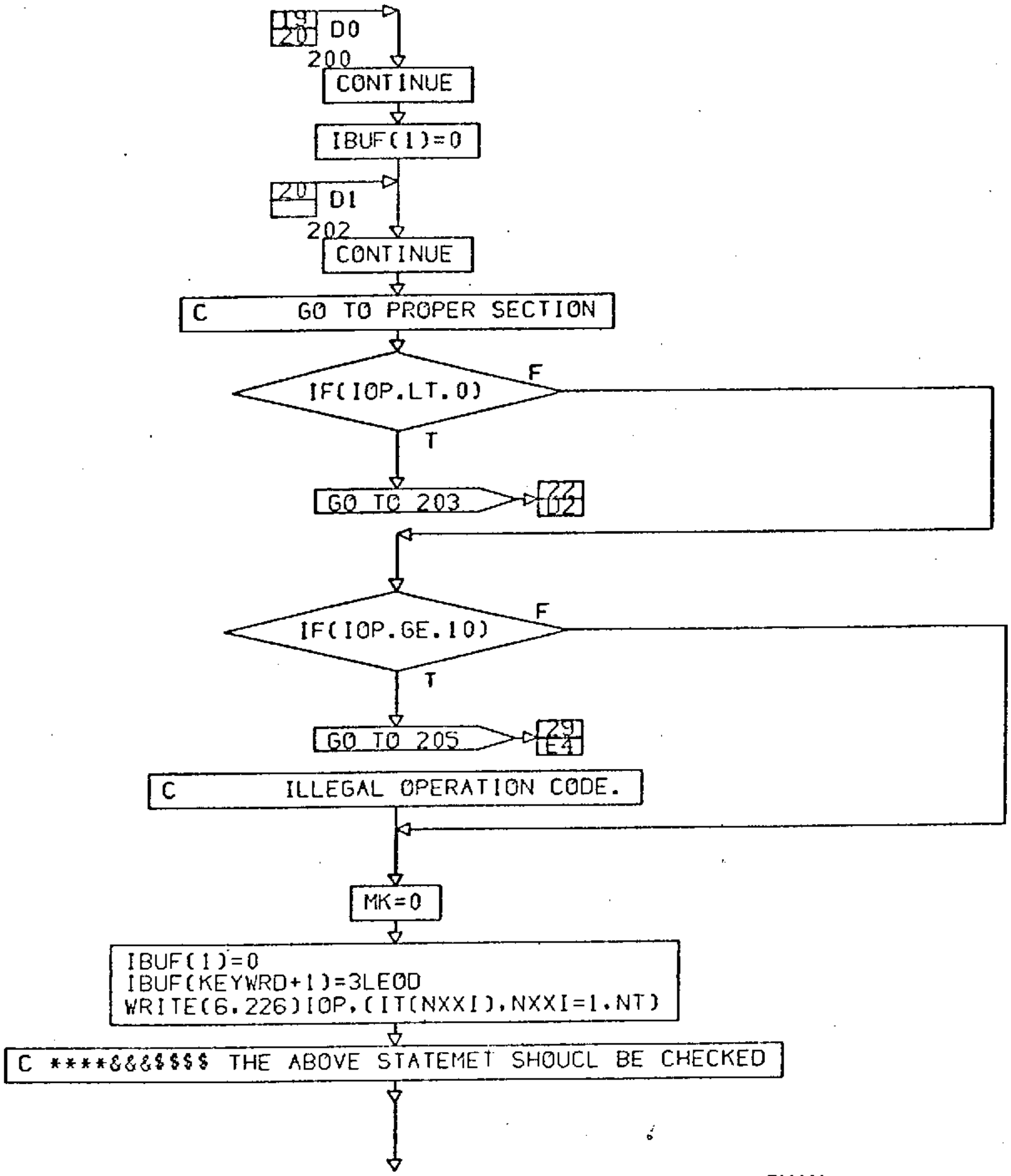
CONT. ON PG 19


18 C6
RETURN
C CHECK BUFFER FOR MATCH - IF NOMATCH EMPTY IF NECESSARY
18 C7
IF(IBUF(1).EQ.0)
F
T
GO TO 200
21 D0
C THE ABOVE STATEMENT SHOULD BE CHECKED ********$$$$$$$$*******&&&&&&
CALL UPACK7(IBUF(1),LBF)
IF(IBF(1).NE.LBF(1))
F
T
GO TO 201
20 C9
IF(IBF(5).NE.LBF(5))
F
20 C8
T
GO TO 201
20 C9

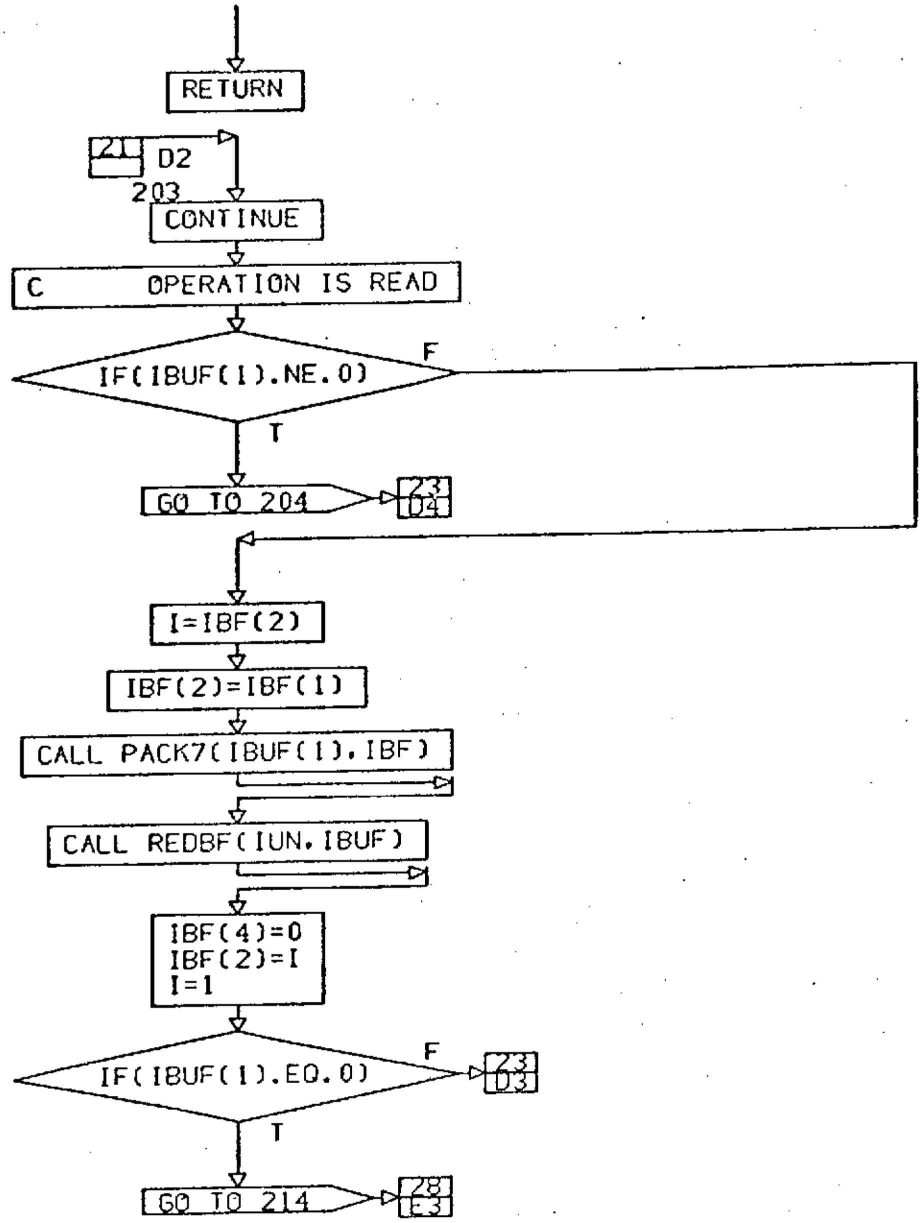
CONT. ON PG 20


19
C8
IF(LBF(4) .EQ. 1)
F
T
IBF(4)=1
IF(IOP.LT.30)
F
T
GO TO 202
21
D1
LBF(4)=IBF(4)
GO TO 223
30
E6
19
9
C9
201
CONTINUE
IF(LBF(4).EQ.0)
F
T
GO TO 200
21
D0
CALL RITBF(IUN,IBUF)

CONT. ON PG 21

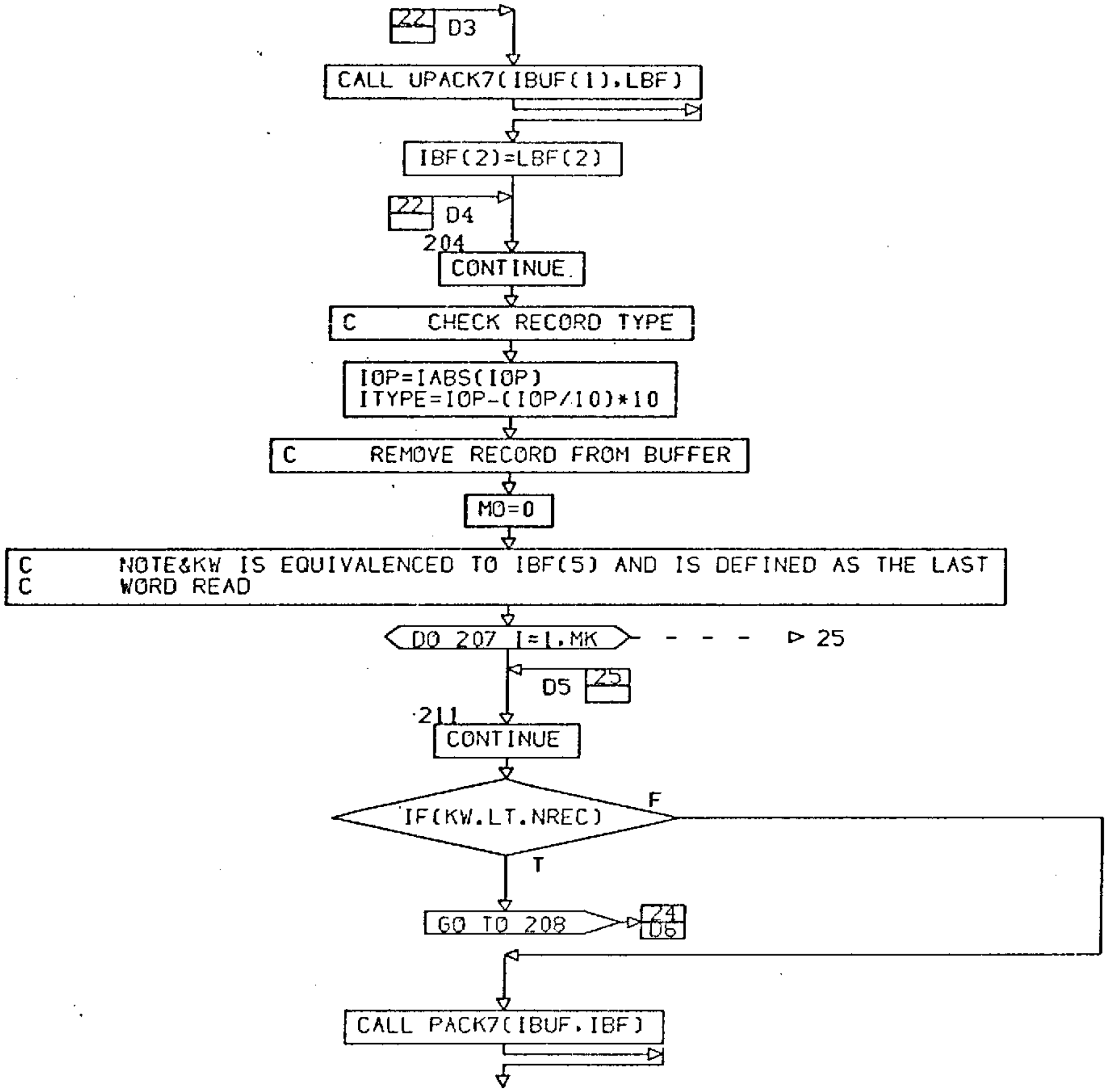

19
20
DO
200
CONTINUE
IBUF(1)=0
20
D1
202
CONTINUE
C GO TO PROPER SECTION
IF(IOP.LT.0)
F
T
GO TO 203
22
D2
IF(IOP.GE.10)
F
T
GO TO 205
29
E4
C ILLEGAL OPERATION CODE.
MK=0
IBUF(1)=0
IBUF(KEYWRD+1)=3LEOD
WRITE(6,226)IOP,(IT(NXXI),NXXI=1,NT)
C ****&&&$$$$ THE ABOVE STATEMET SHOUCL BE CHECKED

CONT. ON PG 22

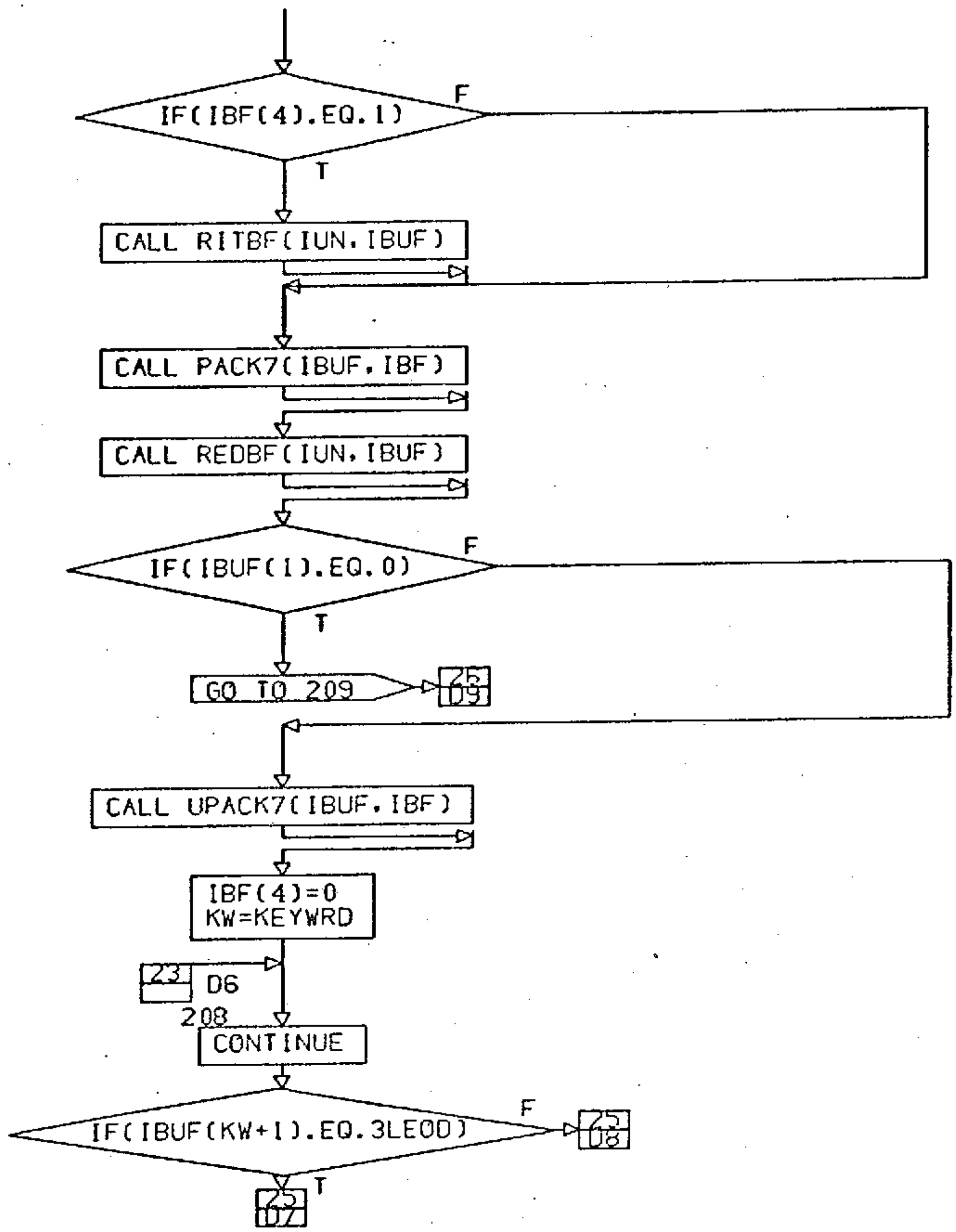


RETURN
21
D2
203
CONTINUE
C OPERATION IS READ
IF(IBUF(1).NE.0)
F
T
GO TO 204
23
D4
I=IBF(2)
IBF(2)=IBF(1)
CALL PACK7(IBUF(1),IBF)
CALL REDBF(IUN,IBUF)
IBF(4)=0
IBF(2)=I
I=1
IF(IBUF(1).EQ.0)
F
23
D3
T
GO TO 214
28
E3

CONT. ON PG 23

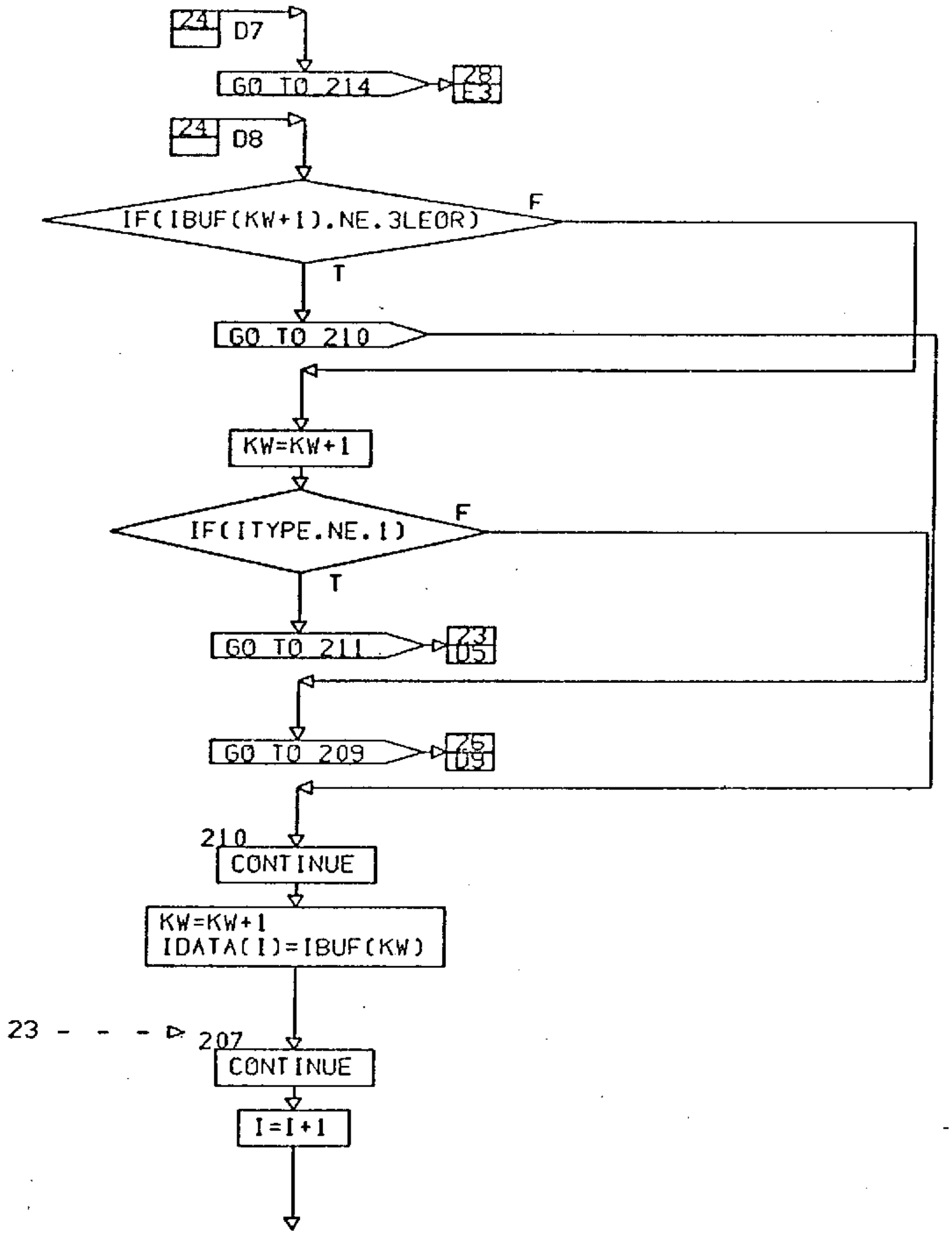

22 D3
CALL UPACK7(IBUF(1),LBF)
IBF(2)=LBF(2)
22 D4
204
CONTINUE
C CHECK RECORD TYPE
IOP=IABS(IOP)
ITYPE=IOP-(IOP/10)*10
C REMOVE RECORD FROM BUFFER
MO=0
C NOTE&KW IS EQUIVALENCED TO IBF(5) AND IS DEFINED AS THE LAST
C WORD READ
DO 207 I=1,MK
25
D5 25
211
CONTINUE
IF(KW.LT.NREC)
F
T
GO TO 208
24 D6
CALL PACK7(IBUF,IBF)

CONT. ON PG 24



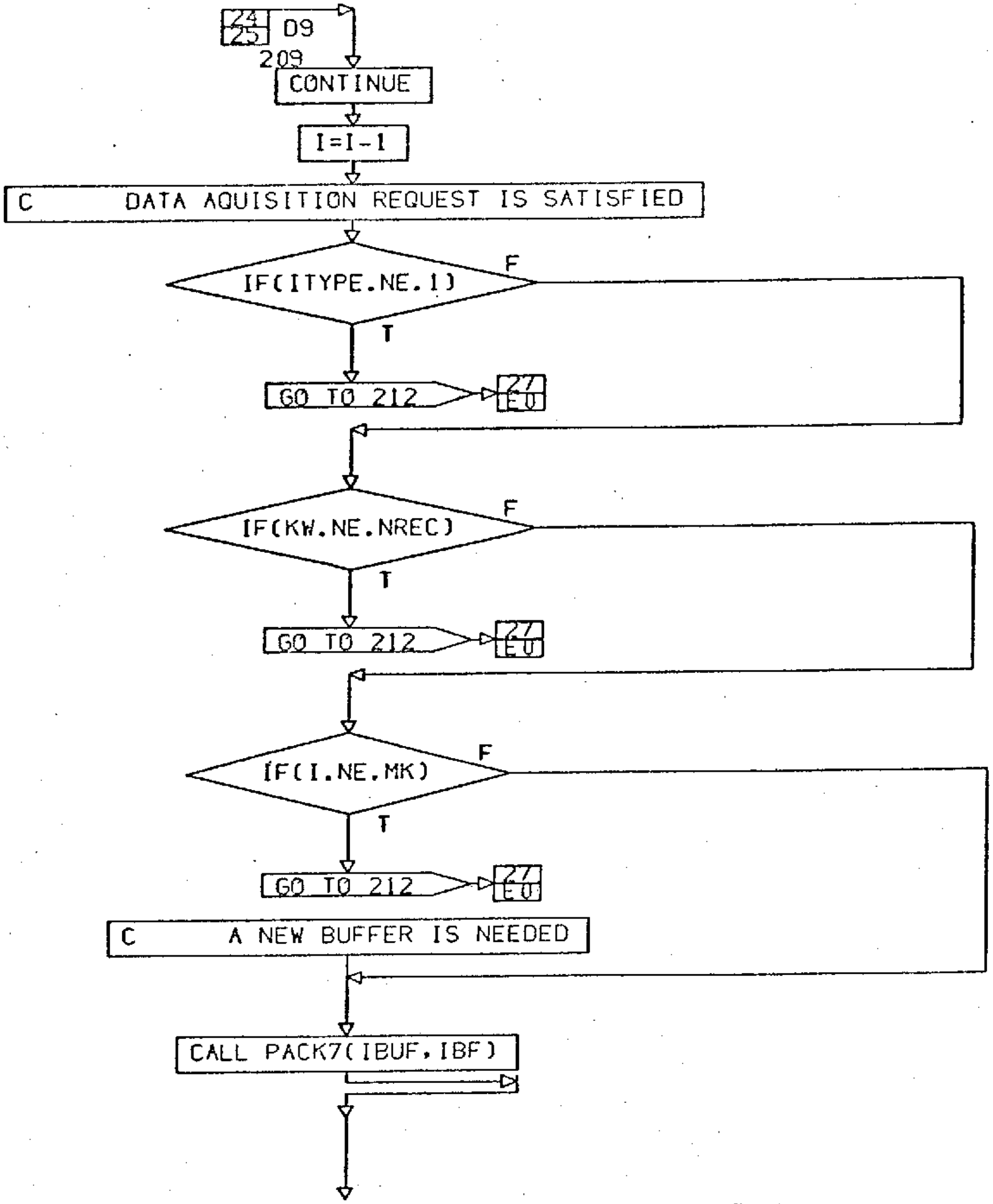
CONT. ON PG 25

DMAN
PG 24 OF 36



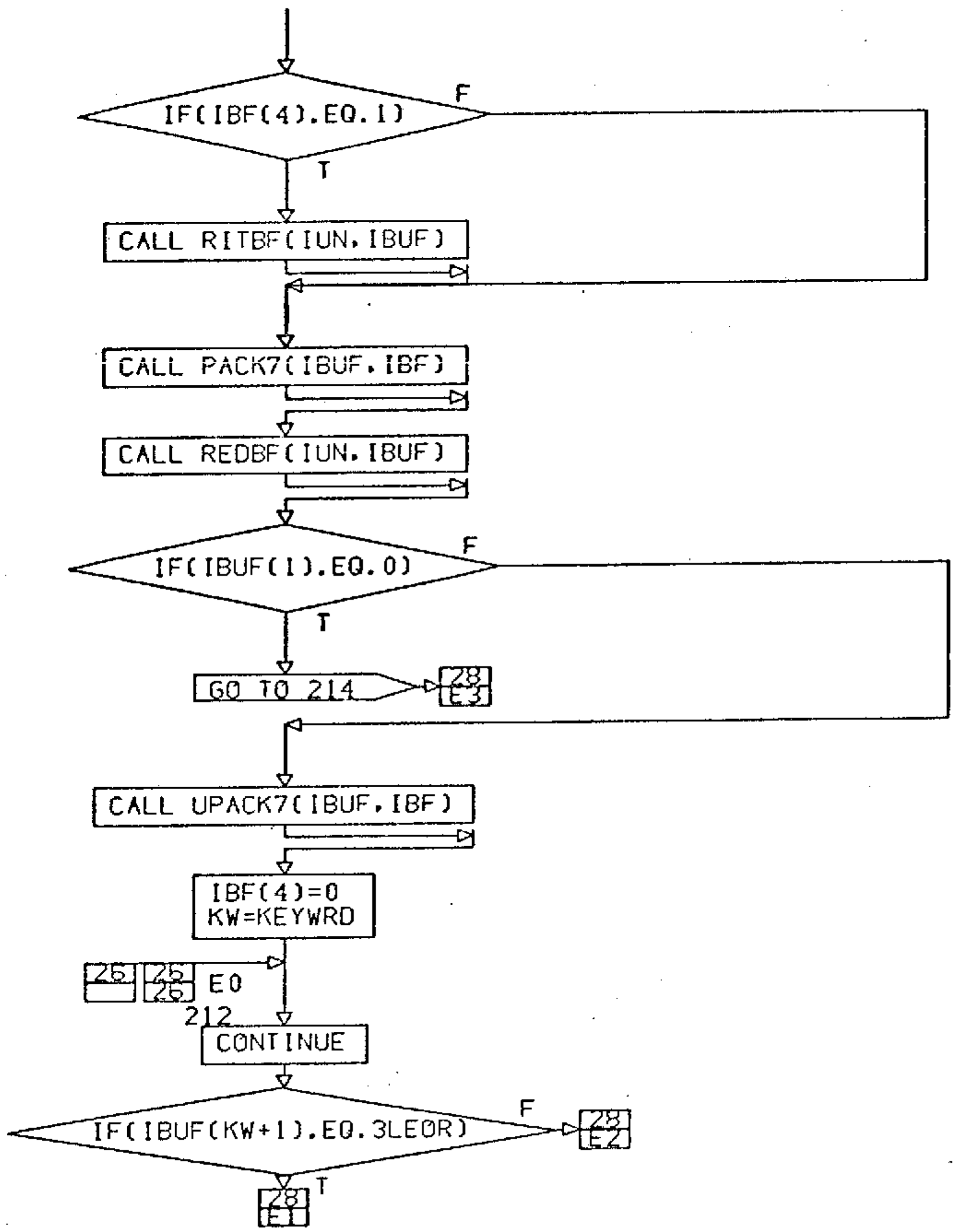
CONT. ON PG 26

DMAN
PG 25 OF 36

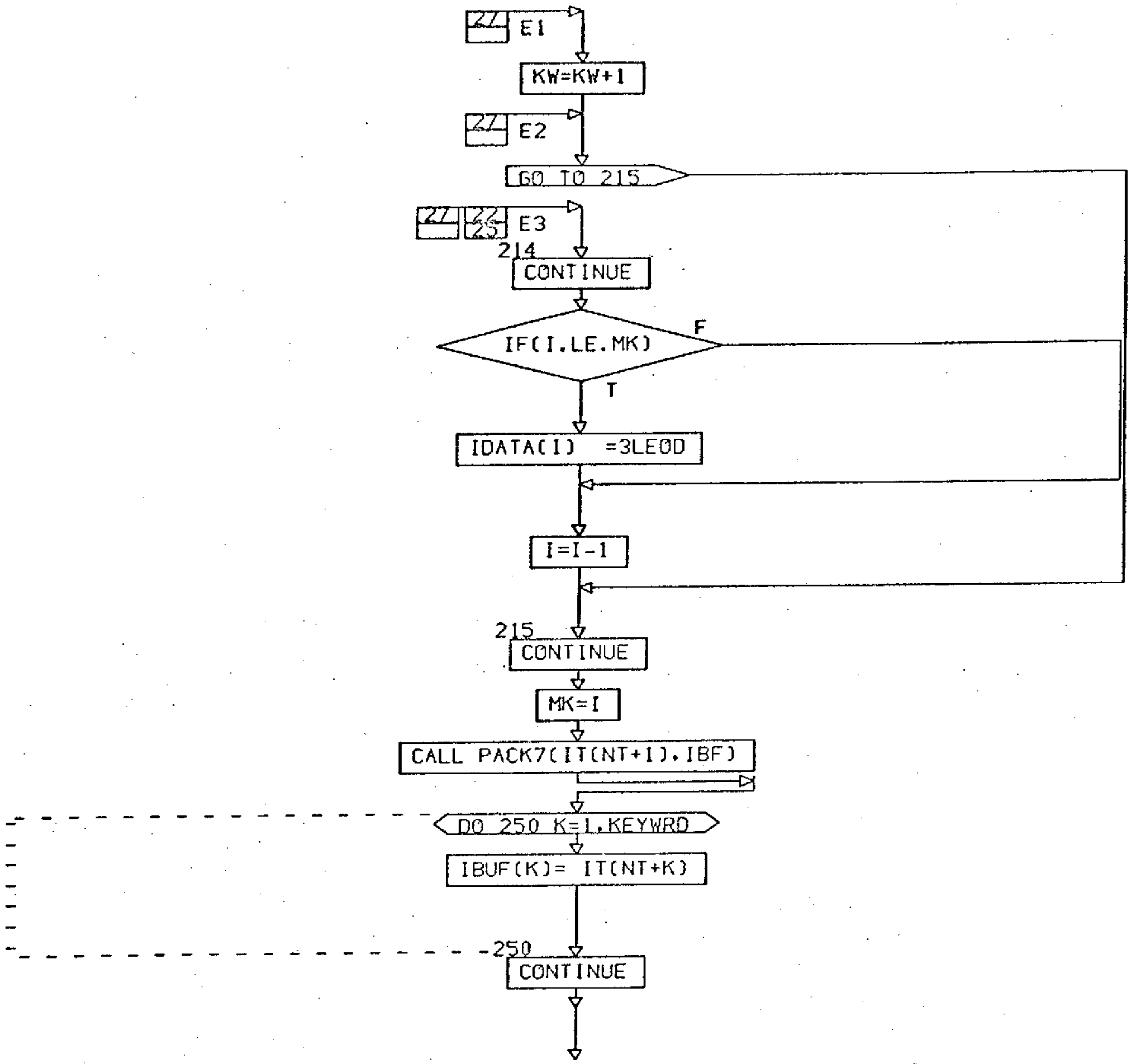


24
25
D9
209
CONTINUE
I=I-1
C DATA AQUISITION REQUEST IS SATISFIED
IF(ITYPE.NE.1)
F
T
GO TO 212
27
E0
IF(KW.NE.NREC)
F
T
GO TO 212
27
E0
IF(I.NE.MK)
F
T
GO TO 212
27
E0
C A NEW BUFFER IS NEEDED
CALL PACK7(IBUF,IBF)

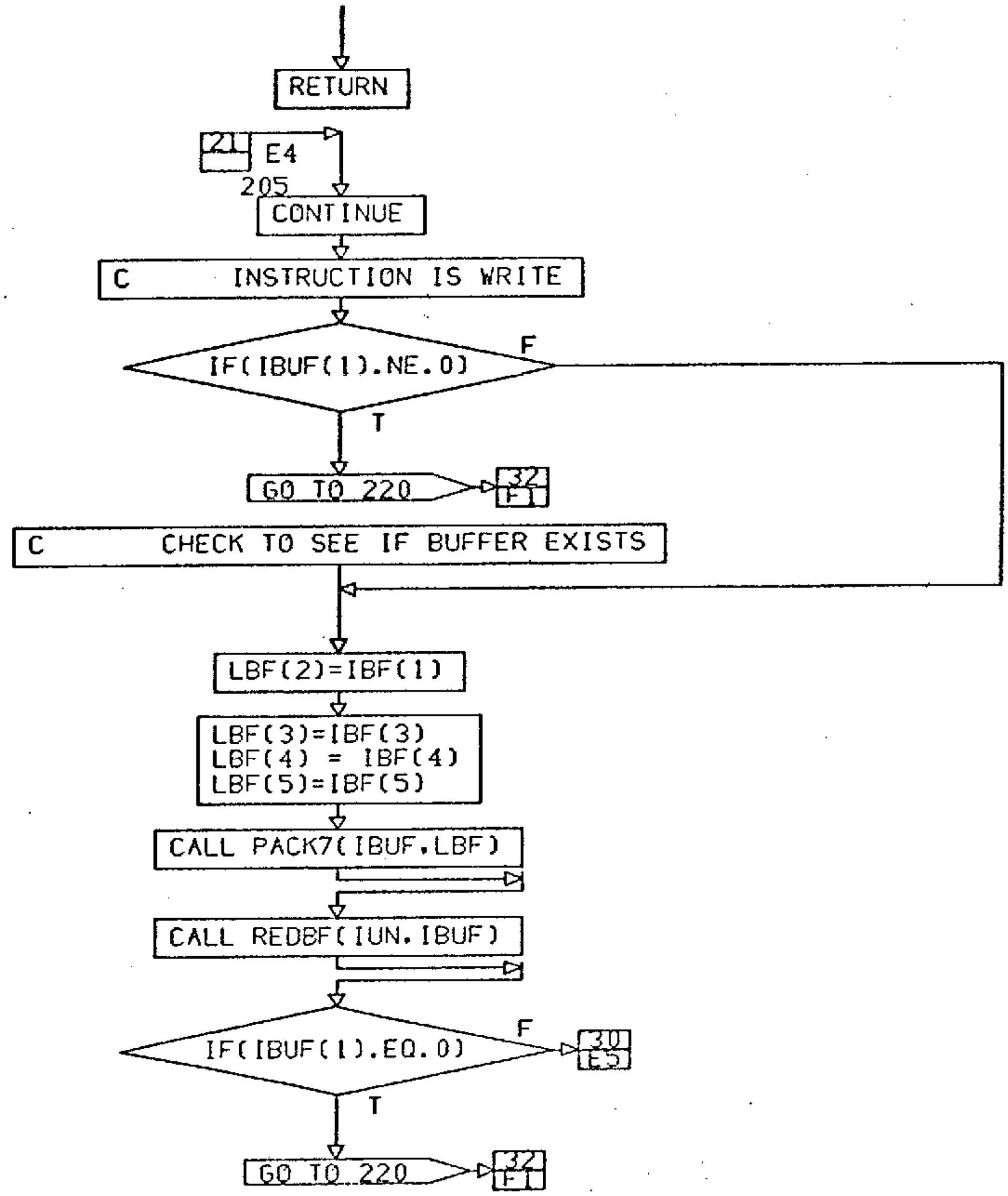
CONT. ON PG 27


IF(IBF(4).EQ.1)
F
T
CALL RITBF(IUN,IBUF)
CALL PACK7(IBUF,IBF)
CALL REDBF(IUN,IBUF)
IF(IBUF(1).EQ.0)
F
T
GO TO 214
28
E3
CALL UPACK7(IBUF,IBF)
IBF(4)=0
KW=KEYWRD
25
25
26
E0
212
CONTINUE
IF(IBUF(KW+1).EQ.3LEOR)
F
28
E2
T
28
E1

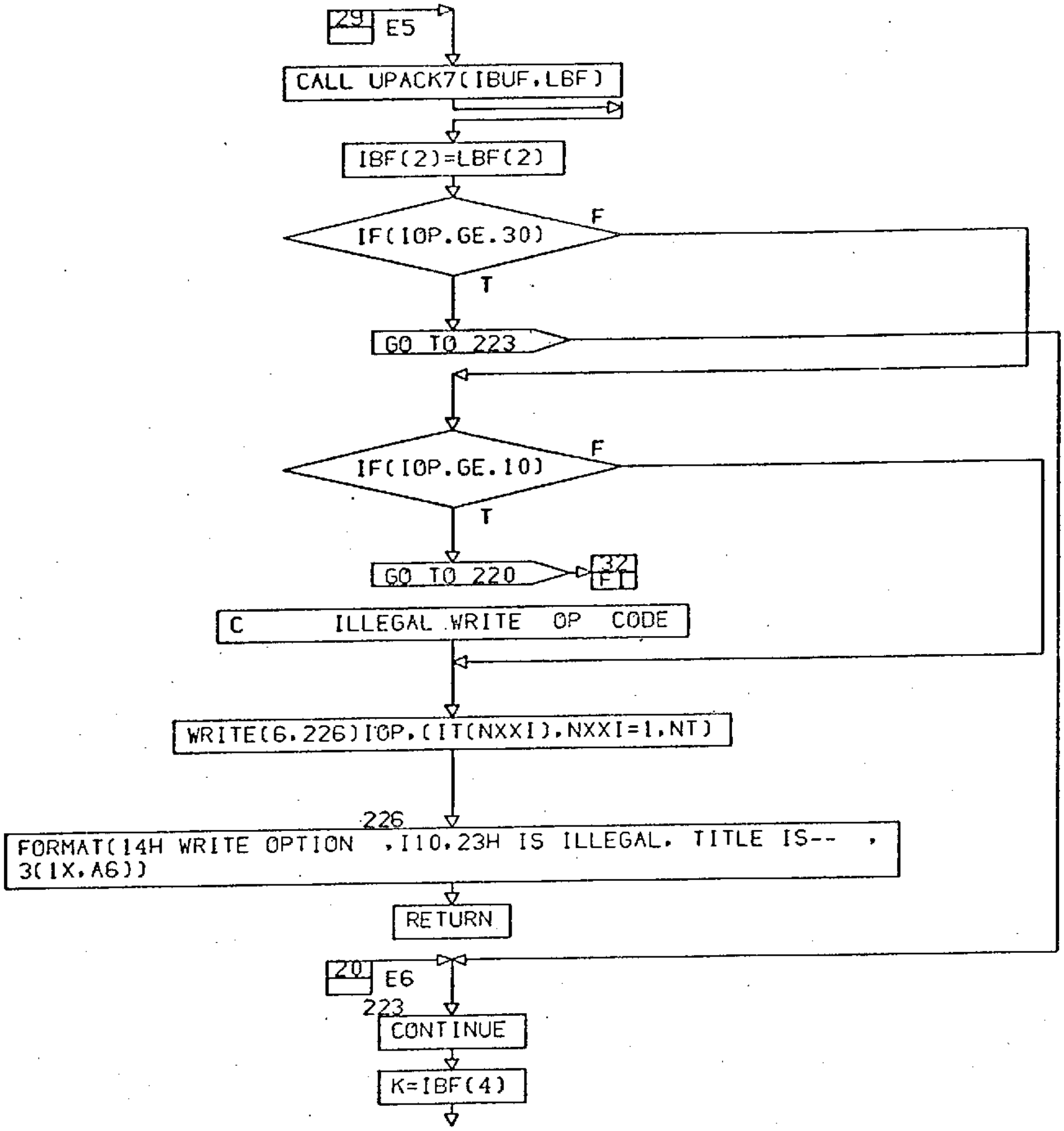
CONT. ON PG 28


27
E1
KW=KW+1
27
E2
GO TO 215
27
22
25
E3
214
CONTINUE
IF(I.LE.MK)
F
T
IDATA(I) =3LEOD
I=I-1
215
CONTINUE
MK=I
CALL PACK7(IT(NT+1),IBF)
DO 250 K=1,KEYWRD
IBUF(K)= IT(NT+K)
250
CONTINUE

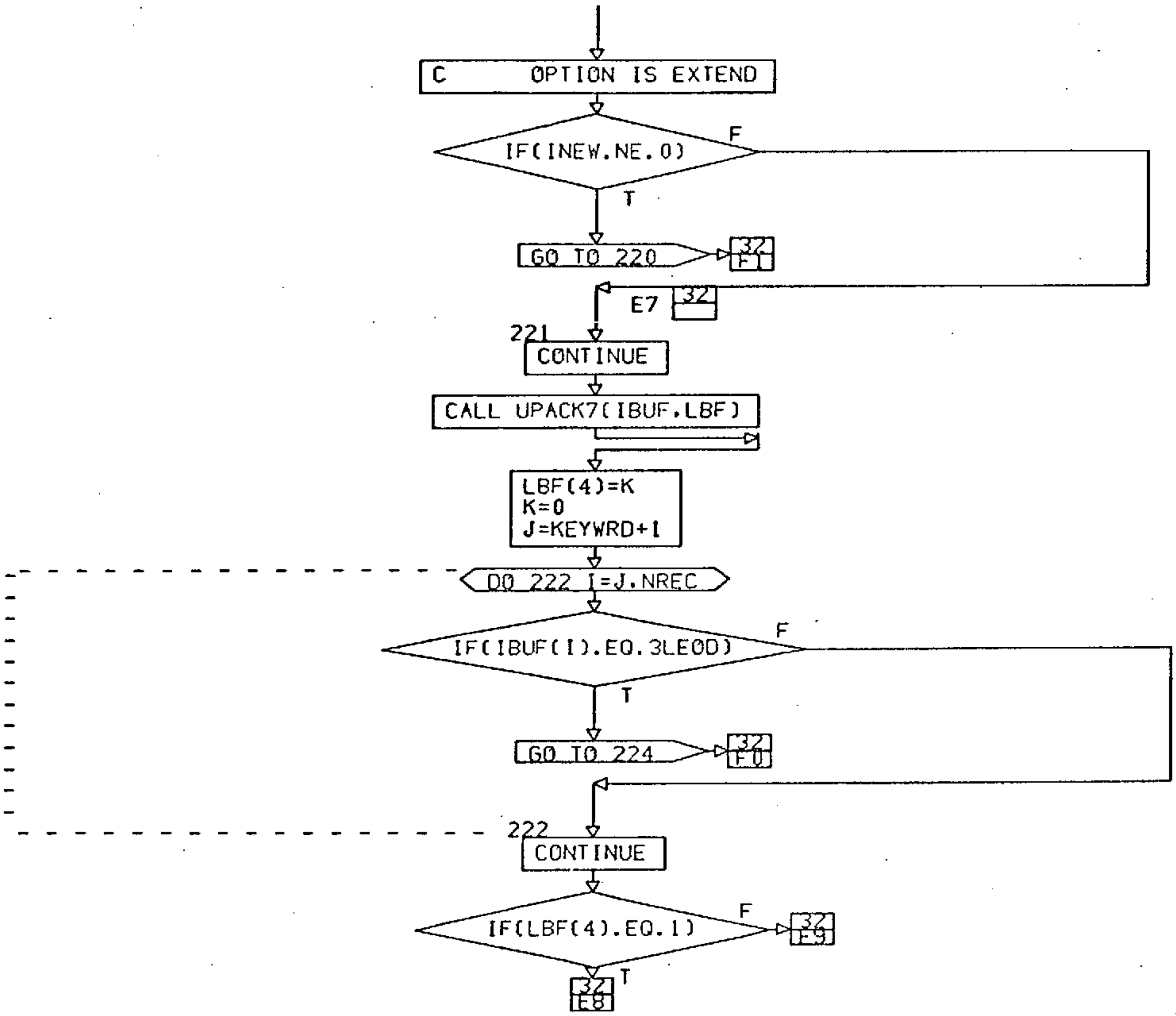
CONT. ON PG 29


RETURN
21
E4
205
CONTINUE
C INSTRUCTION IS WRITE
IF(IBUF(1).NE.0)
F
T
GO TO 220
32
F1
C CHECK TO SEE IF BUFFER EXISTS
LBF(2)=IBF(1)
LBF(3)=IBF(3)
LBF(4) = IBF(4)
LBF(5)=IBF(5)
CALL PACK7(IBUF,LBF)
CALL REDBF(IUN,IBUF)
IF(IBUF(1).EQ.0)
F
30
E5
T
GO TO 220
32
F1

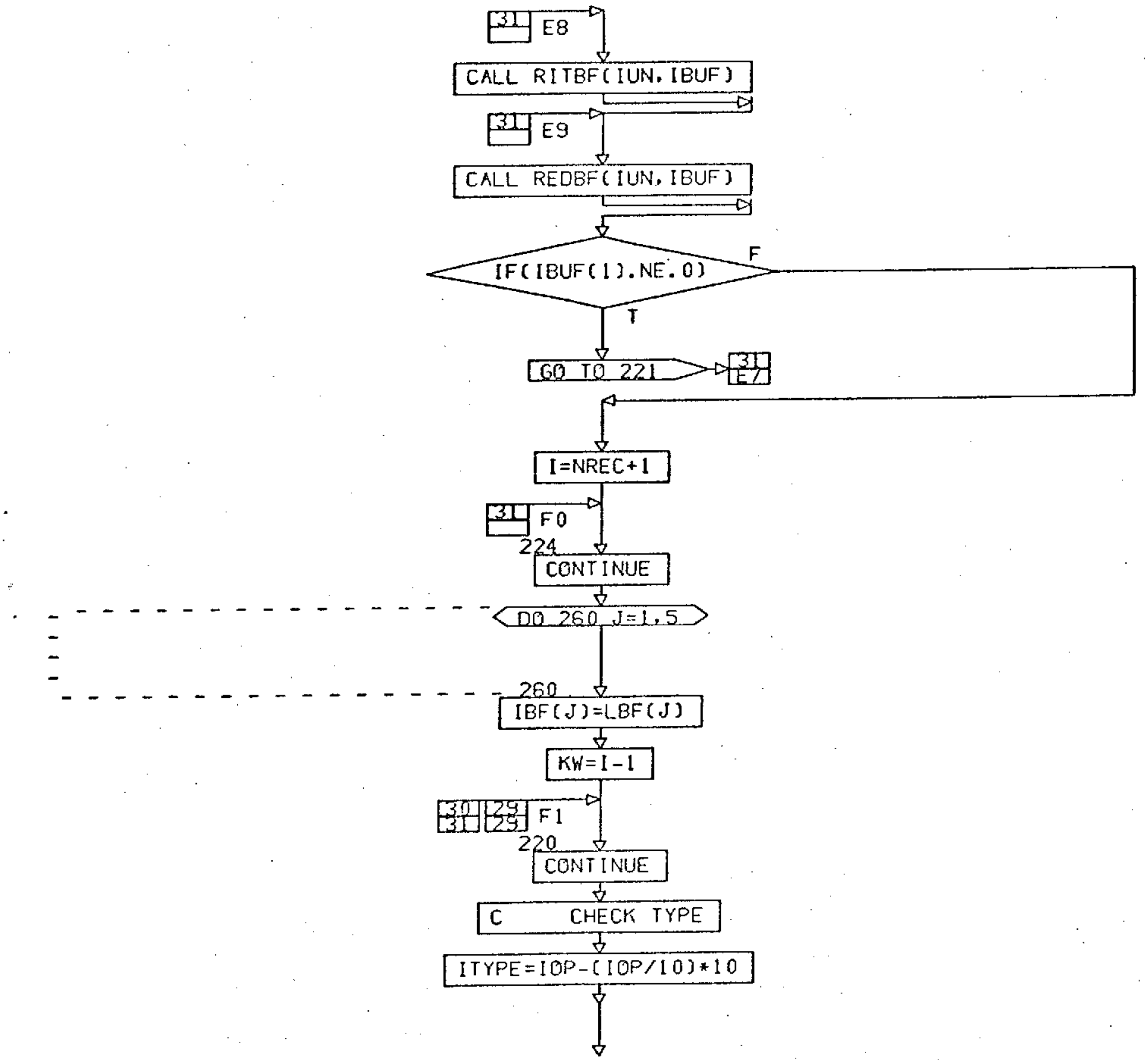
CONT. ON PG 30


29
E5
CALL UPACK7(IBUF,LBF)
IBF(2)=LBF(2)
IF(IOP.GE.30)
F
T
GO TO 223
IF(IOP.GE.10)
F
T
GO TO 220
32
F1
C ILLEGAL WRITE OP CODE
WRITE(6,226)IOP,(IT(NXXI),NXXI=1,NT)
226
FORMAT(14H WRITE OPTION ,I10,23H IS ILLEGAL. TITLE IS-- ,
3(1X,A6))
RETURN
20
E6
223
CONTINUE
K=IBF(4)

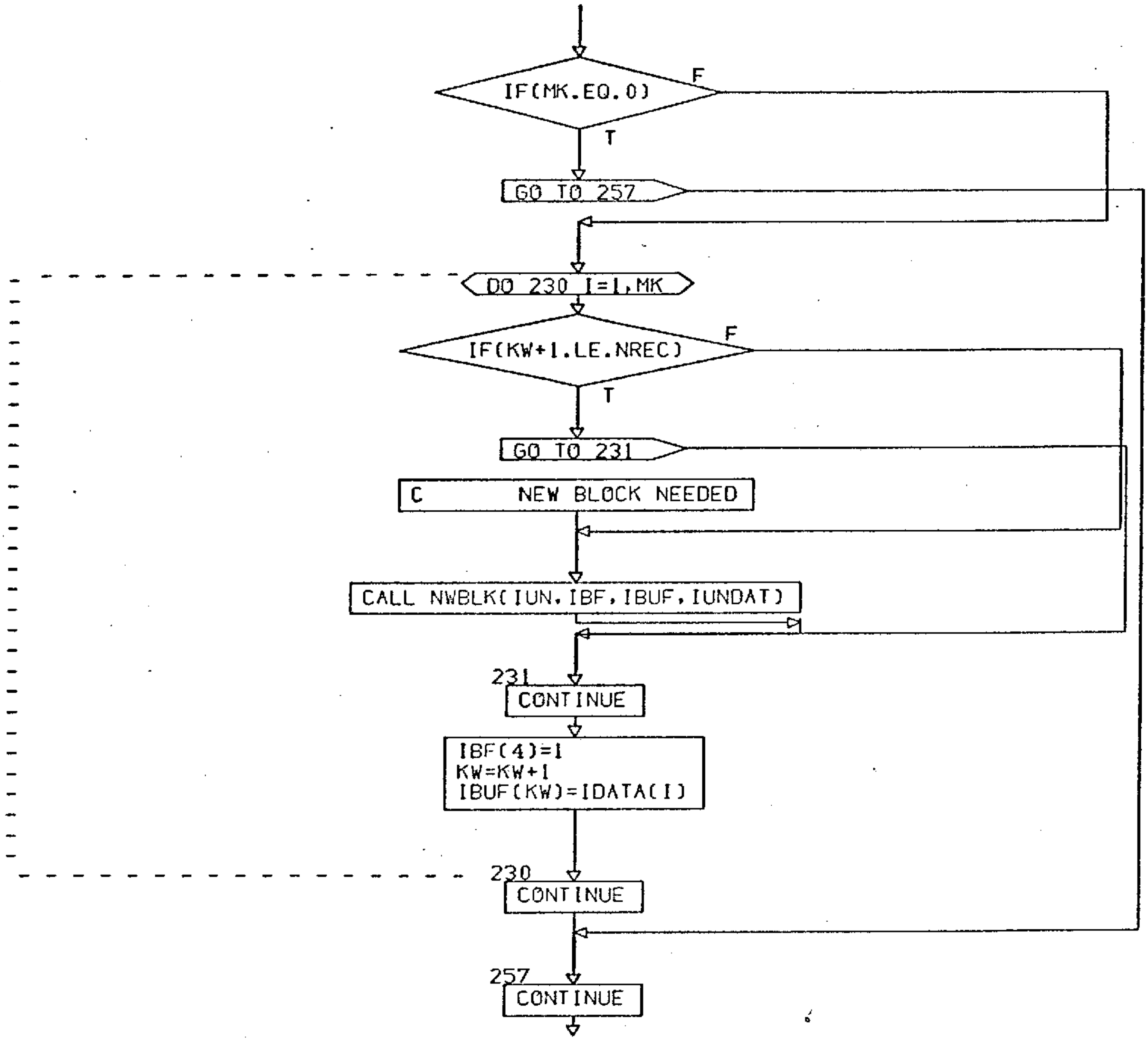
CONT. ON PG 31


C OPTION IS EXTEND
IF(INEW.NE.0)
F
T
GO TO 220
32
F1
E7
32
221
CONTINUE
CALL UPACK7(IBUF,LBF)
LBF(4)=K
K=0
J=KEYWRD+1
DO 222 I=J,NREC
IF(IBUF(I).EQ.3LEOD)
F
T
GO TO 224
32
F0
222
CONTINUE
IF(LBF(4).EQ.1)
F
32
E9
T
32
E8

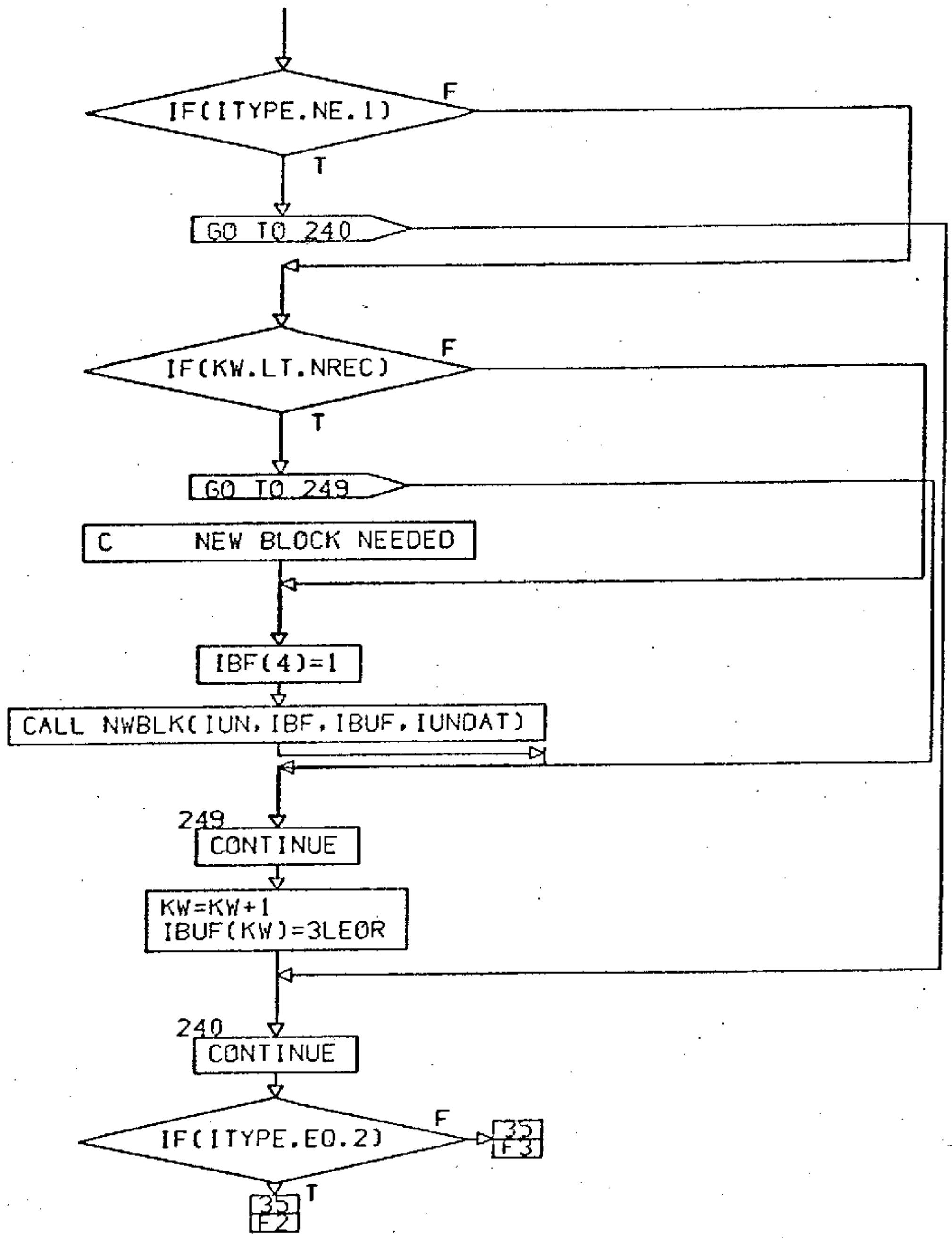
CONT. ON PG 32


31 E8
CALL RITBF(IUN,IBUF)
31 E9
CALL REDBF(IUN,IBUF)
IF(IBUF(1).NE.0)
F
T
GO TO 221
31 E7
I=NREC+1
31 F0
224
CONTINUE
DO 260 J=1,5
260
IBF(J)=LBF(J)
KW=I-1
30 29 31 25 F1
220
CONTINUE
C CHECK TYPE
ITYPE=IOP-(IOP/10)*10

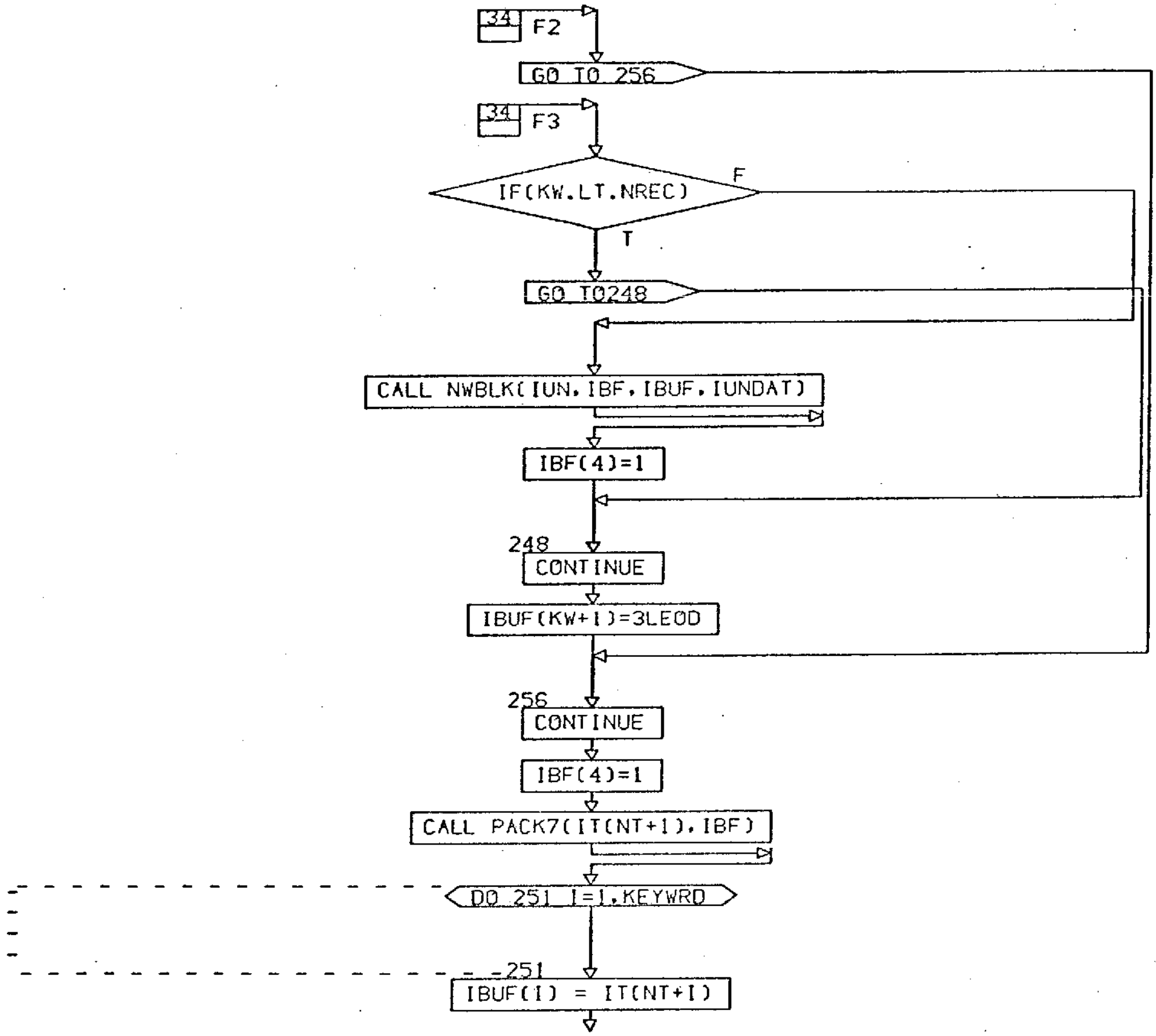
CONT. ON PG 33


IF(MK.EQ.0)
F
T
GO TO 257
DO 230 I=1,MK
IF(KW+1.LE.NREC)
F
T
GO TO 231
C NEW BLOCK NEEDED
CALL NWBLK(IUN,IBF,IBUF,IUNDAT)
231
CONTINUE
IBF(4)=1
KW=KW+1
IBUF(KW)=IDATA(I)
230
CONTINUE
257
CONTINUE

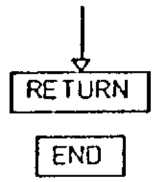
CONT. ON PG 34

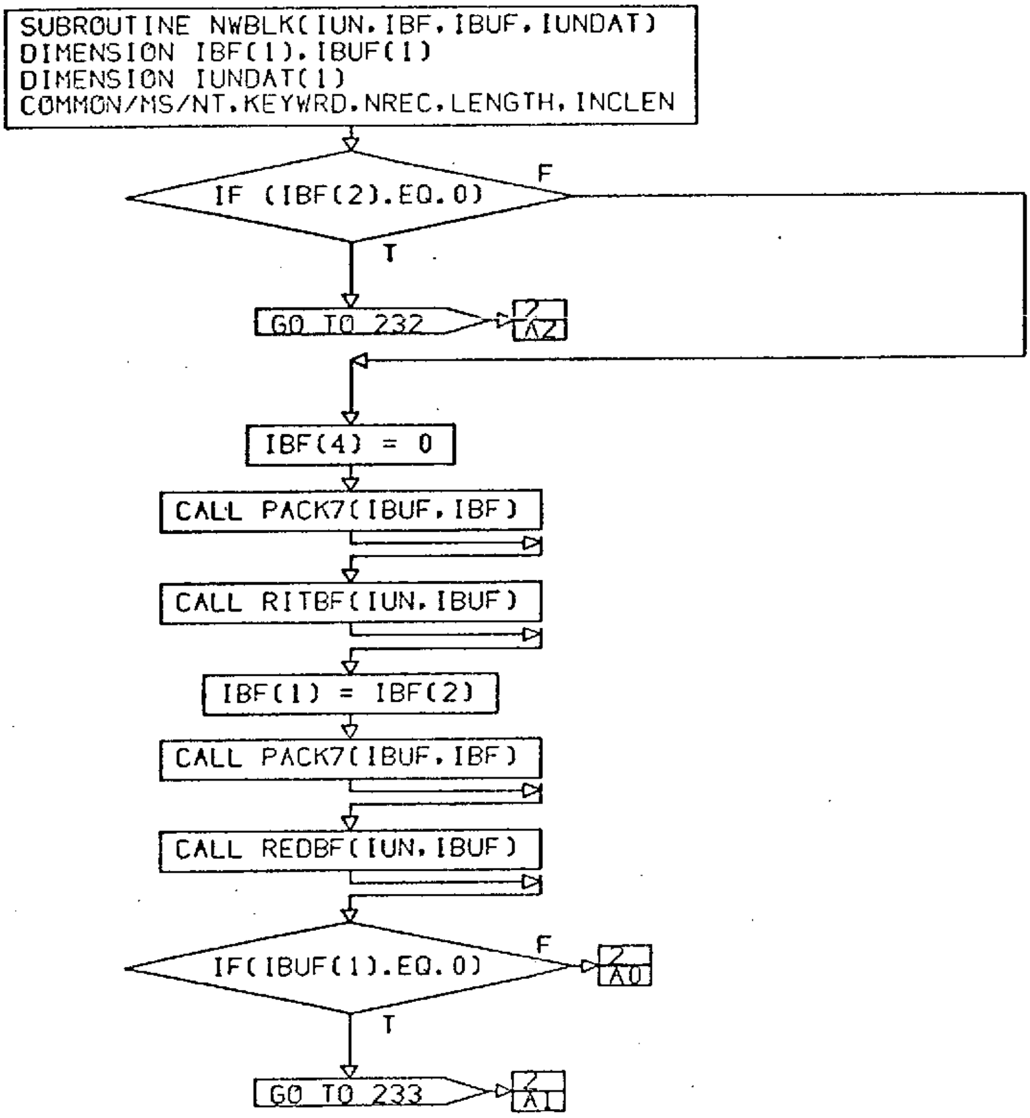

IF(ITYPE.NE.1)
F
T
GO TO 240
IF(KW.LT.NREC)
F
T
GO TO 249
C NEW BLOCK NEEDED
IBF(4)=1
CALL NWBLK(IUN,IBF,IBUF,IUNDAT)
249
CONTINUE
KW=KW+1
IBUF(KW)=3LEØR
240
CONTINUE
IF(ITYPE.EQ.2)
F
35
F3
T
35
F2

CONT. ØN PG 35

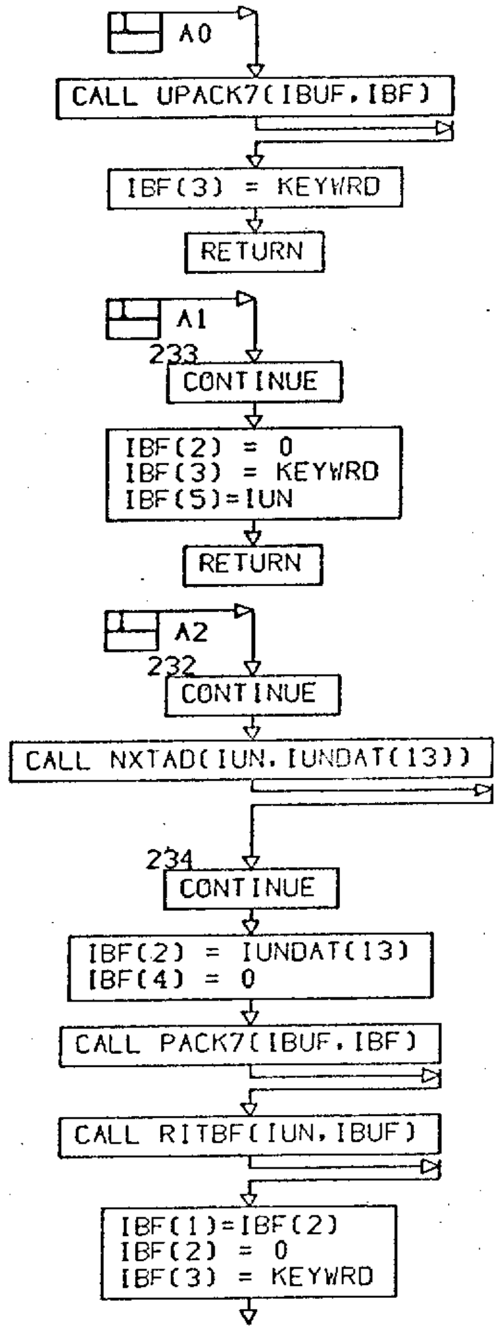

34
F2
GO TO 256
34
F3
IF(KW.LT.NREC)
F
T
GO TO248
CALL NWBLK(IUN,IBF,IBUF,IUNDAT)
IBF(4)=1
248
CONTINUE
IBUF(KW+1)=3LEOD
256
CONTINUE
IBF(4)=1
CALL PACK7(IT(NT+1),IBF)
DO 251 I=1,KEYWRD
251
IBUF(I) = IT(NT+I)

CONT. ON PG 36


RETURN
END

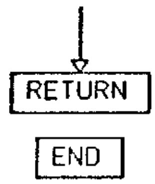

SUBROUTINE NWBLK(IUN,IBF,IBUF,IUNDAT)
DIMENSION IBF(1),IBUF(1)
DIMENSION IUNDAT(1)
COMMON/MS/NT,KEYWRD,NREC,LENGTH,INCLEN
IF (IBF(2).EQ.0)
F
T
GO TO 232
2
AZ
IBF(4) = 0
CALL PACK7(IBUF,IBF)
CALL RITBF(IUN,IBUF)
IBF(1) = IBF(2)
CALL PACK7(IBUF,IBF)
CALL REDBF(IUN,IBUF)
IF(IBUF(1).EQ.0)
F
2
A0
T
GO TO 233
2
A1

CONT. ON PG 2

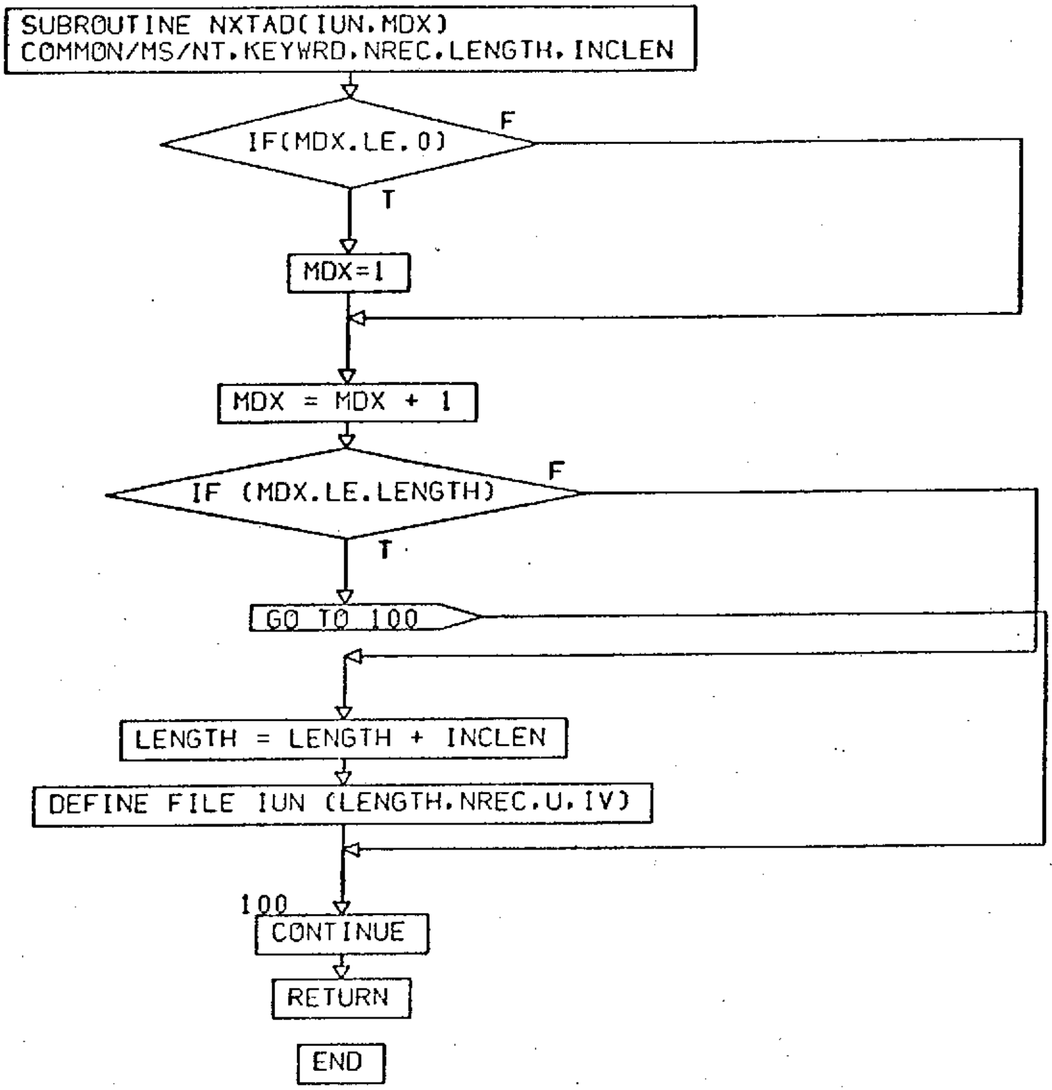


CONT. ON PG 3

NWBLK
PG 2 OF 3



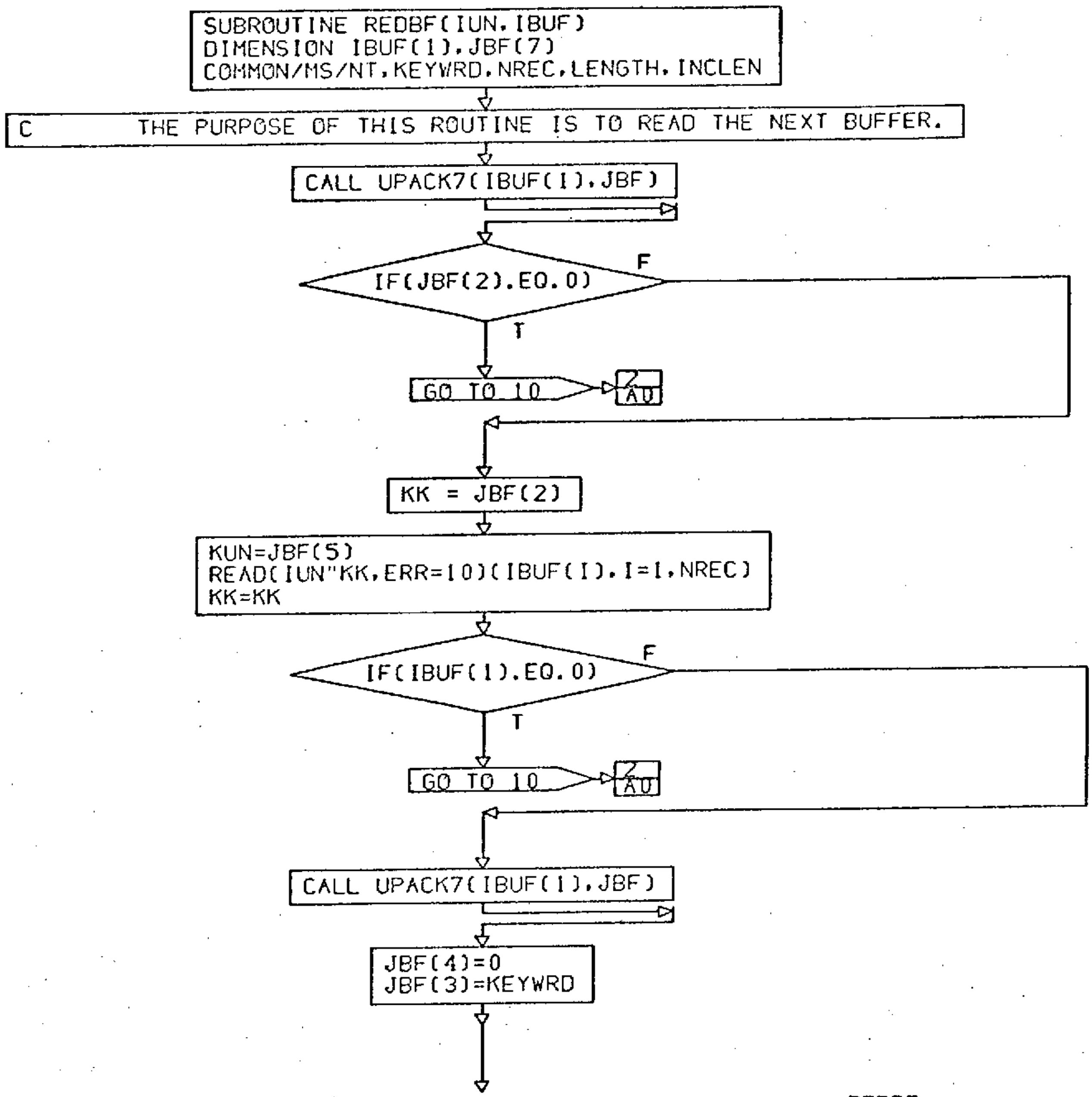
NWBLK
PG 3 FINAL



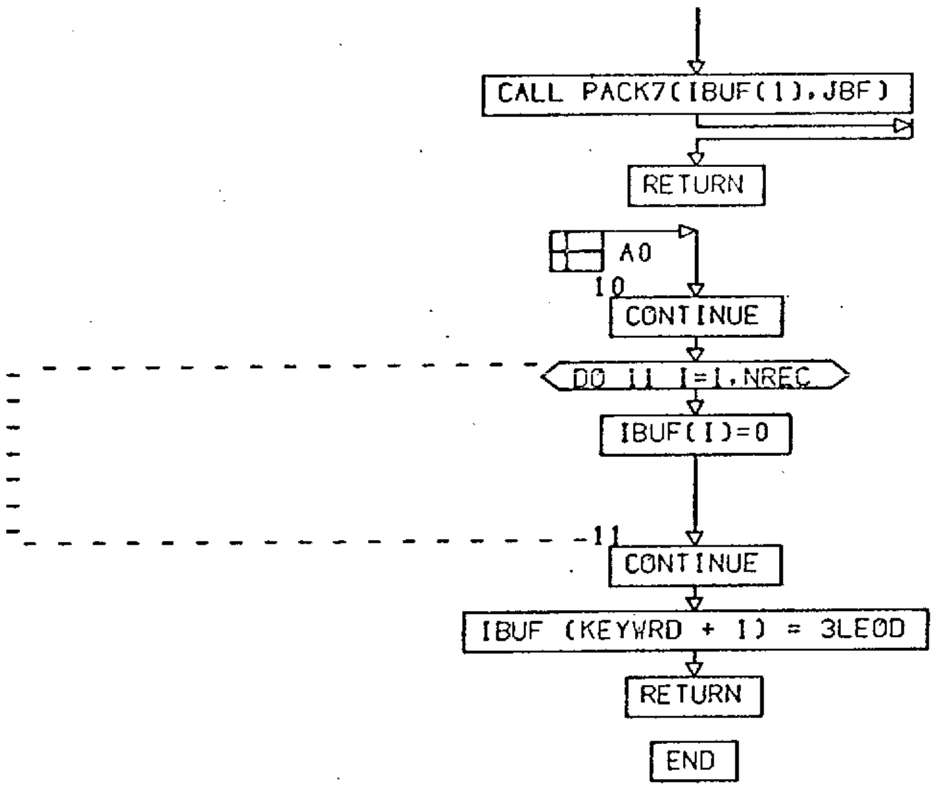
NXTAD
PG 1 FINAL

ORIGINAL PAGE IS
OF POOR QUALITY

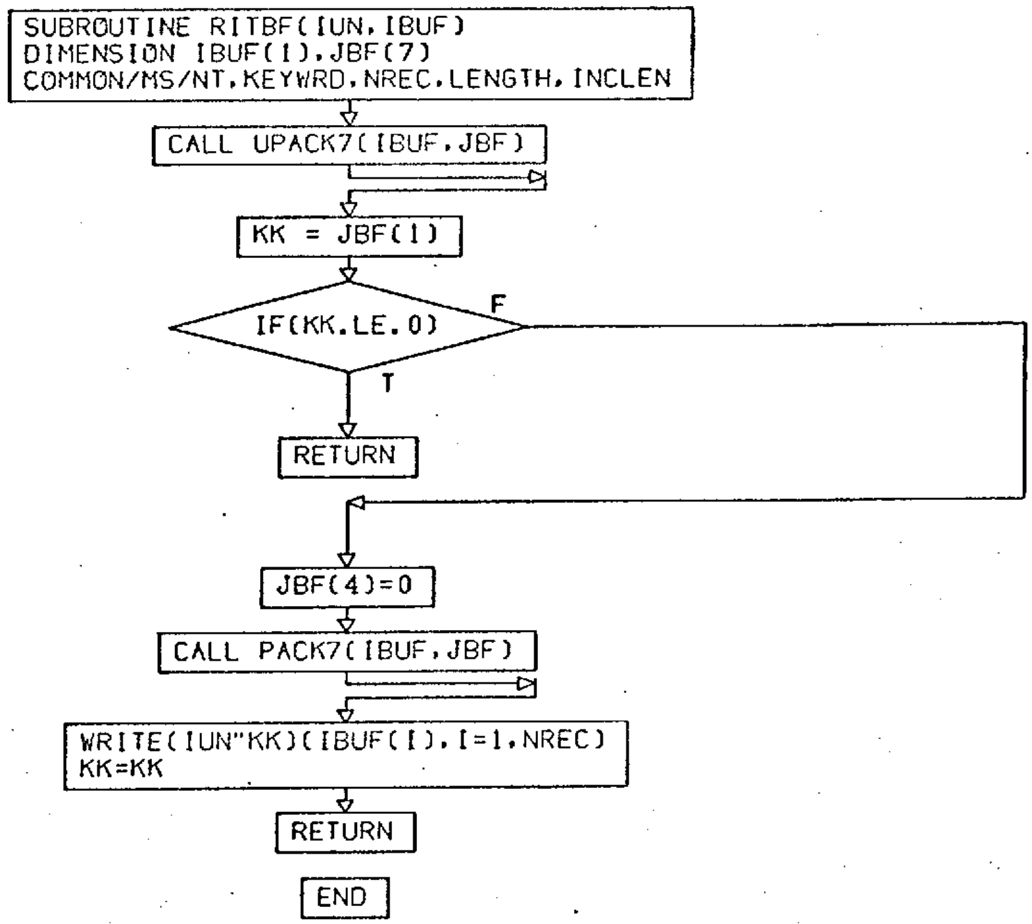
```
SUBROUTINE PACK7(IWORD,IARRAY)
INTEGER IWORD(1), IARRAY(1)
IXRY1 = IARRAY(1)
IXRY2 = IARRAY(2)
IXRY3 = IARRAY(3)
IXRY4 = IARRAY(4)
IXRY5 = IARRAY(5)
IWXD1=0

IWXD2=0
FLD(0,20,IWXD1)=FLD(16,20,IXRY1)
FLD(20,16,IWXD1)=FLD(16,16,IXRY2)
FLD(0,4,IWXD2)=FLD(32,4,IXRY2)
FLD(4,10,IWXD2)=FLD(26,10,IXRY3)
FLD(14,10,IWXD2)=FLD(26,10,IXRY4)
FLD(24,10,IWXD2)=FLD(26,10,IXRY5)
IWORD(1) = IWXD1

IWORD(2) = IWXD2

RETURN

END
```


SUBROUTINE REDBF(IUN,IBUF)
DIMENSION IBUF(1),JBF(7)
COMMON/MS/NT,KEYWRD,NREC,LENGTH,INCLEN
C THE PURPOSE OF THIS ROUTINE IS TO READ THE NEXT BUFFER.
CALL UPACK7(IBUF(1),JBF)
IF(JBF(2).EQ.0)
F
T
GO TO 10
2
AU
KK = JBF(2)
KUN=JBF(5)
READ(IUN"KK,ERR=10)(IBUF(I),I=1,NREC)
KK=KK
IF(IBUF(1).EQ.0)
F
T
GO TO 10
2
AU
CALL UPACK7(IBUF(1),JBF)
JBF(4)=0
JBF(3)=KEYWRD

CONT. ON PG 2



REDBF
PG 2 FINAL



RITBF
PG 1 FINAL

```
SUBROUTINE UPACK7(IWORD,IARRAY)
INTEGER IWORD(1), IARRAY(1)
IWXD1 = IWORD(1)
IWXD2 = IWORD(2)
DATA IXRY1/0/
DATA IXRY2/0/
DATA IXRY3/0/
DATA IXRY4/0/
```

↓

```
DATA IXRY5/0/
FLD(16,20,IXRY1)=FLD(0,20,IWXD1)
FLD(16,16,IXRY2)=FLD(20,16,IWXD1)
FLD(32,4,IXRY2)=FLD(0,4,IWXD2)
FLD(26,10,IXRY3)=FLD(4,10,IWXD2)
FLD(26,10,IXRY4)=FLD(14,10,IWXD2)
FLD(26,10,IXRY5)=FLD(24,10,IWXD2)
IARRAY(1)=IXRY1
```

↓

```
IARRAY(2)=IXRY2
IARRAY(3)=IXRY3
IARRAY(4)=IXRY4
IARRAY(5)=IXRY5
```

↓

```
RETURN
```

```
END
```

UPACK7
PG 1 FINAL