IMPLEMENTATION OF A SINGLE STACK SYMBOL STREAM PROCESSOR

by 4528

THERESA ADEL HELLNER

B.A. in Ed., Arizona State University, 1968

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

Approved by:

K. E. Kemp
Major Professor

ACKNOWLEDGEMENT

I wish to acknowledge Mr. Charles S. Walker, Assistant Professor of Computer Science at Kansas State University. Although not a member of my committee, Mr. Walker provided invaluable assistance in the preparation of this report for which I am deeply indebted. I also wish to acknowledge Mr. Walter Moore, Manager of Data Processing at Montgomery Ward and Co. in Kansas City, Mo. for allowing the use of their special Xerox machine for reproducing the program listing included in the Appendix.

LD
2668
R4
1970
H4488
C.2

TABLE OF CONTENTS

Page

LIST OF TABLES iv

LIST OF FIGURES v

Chapter

1. INTRODUCTION 1

2. DESCRIPTION OF GPM 4

THE STACK AND POINTERS 5

OPERATION OF GPM 10

3. IMPLEMENTATION 20

4. SUMMARY. 28

BIBLIOGRAPHY. 37

APPENDIXES. . . . . .

I Listing of the Program for GPM 38

II Input and Output of the Tests for GPM. 97

LIST OF TABLES

Table | Page

1. The Primitive (Control) Symbols and Their Functions 6

LIST OF FIGURES

Figure		Page
1.	Macro-call Examples.	12
2.	Macro-calls and Their Results.	15
3.	Macro-definition Calls	15
4.	Storage Representation in the Stack Cell . .	23
5.	Definition of the SUCCESSOR Macro.	33
6.	Definition of the FACT Macro	33

Chapter 1

INTRODUCTION

This report was concerned with the implementation of the General Purpose Macrogenerator (GPM) as described by C. Strachey.[1] Although originally designed as a macrogenerator to be used in conjunction with a symbolic assembler, GPM is actually a symbol string processor having the capabilities of function evaluation and function definition. Thus, one can consider that the collection of all the permitted input data forms the domain of GPM and the collection of all possible results forms the range of GPM.

The basic concept of GPM is that a program consists of strings containing sequences of functions which can be indefinitely nested. Evaluation of these strings proceeds from the innermost level to the outermost level, and from left to right within a level.

Function evaluation in GPM occurs as a result of a macro-call which contains the macro's name and possibly a set of actual parameters to be substituted for the formal parameters in the definition of the macro. One unusual feature of GPM is that macro or function definition is the result of a macro-call to a primitive

---

[1] "The General Purpose Macrogenerator", The Computer Journal, VIII, 3 (October 1965), 225-241.

or machine macro in which the macro-call includes the macro name and macro body as parameters. Both the data strings and the strings specifying function calls, the macro-calls, are intermingled in GPM.

Unlike most macrogenerators, GPM evaluates the parameters of a macro-call before substitution takes place, hence, allowing functions to be nested. Substitution involves physically substituting the actual parameters for the formal parameters in the definition according to the following rules as stated by Wegner.

1. The actual parameters are substituted for the formal parameters.

2. The resultant definition of the function is substituted "at the point of the macro-call".

3. With regards to the context of the macro-call, the macro definition is evaluated.[2]

GPM was written for the specific purpose of writing a compiler for CPL (Combined Programming Language) and differs from other macrogenerators principally in the primitives that are used. The term "general" is the result of (1) GPM being a separate program and not part of a symbolic assembler, and (2) GPM allowing the nesting of function calls.

The purpose of this report was to implement the General Purpose Macrogenerator, GPM, using an appropriate programming lan- and modifying the program to be operable on the IBM 360/50 at

---

[2]"Macro Generators and the Lambda Calculus", Programming Languages, Information Structures, and Machine Organization (New York: McGraw-Hill, Inc., 1968), p. 147.

Kansas State University. Implementation involved adjustments to the program due to the particular machine used, and additions to increase the efficiency and clarity of the program. Chapter 3 discusses the considerations involved in implementing GPM. The program was tested using examples given by Strachey and functions that were specifically designed to test various capabilities such as recursive functions and conditional expressions. The results of the various functions and symbol strings used to test GPM are included in Appendix II with a summary of these tests given in Chapter 4.

Before the implementation of GPM is discussed, a general description of the program is given in the second chapter. The structure, the specific control features and characters, and the various actions that GPM can perform while processing in one of its specific operation modes are covered. A listing of the program can be found in Appendix I for reference.

Chapter 2

DESCRIPTION OF GPM

GPM is a powerful symbol string processor that uses substitution techniques for processing. The input stream consists of arbitrary sequences of characters which are scanned left to right and directly copied to the output stream. The evaluation of a function (macro) call results in a macro being expanded, a macro being defined, or both.

GPM evaluates and defines macros using a single push-down stack, a few primitive symbols, which are the control characters, and a few primitive macros, which are the machine or system macros. Each action is determined by a primitive symbol encountered in the source stream while GPM is in one of its specific operation modes. Scanning a control character in any of the various source streams may change the operation mode as well as the source and/or target streams. A description of the primitive symbols used in this report and their effect is given in Table 1.

The input stream, which will sometimes be called the input string, forms the most important source stream. Other source streams are formed by scanning various sections of the stack with the processing results being placed either on the stack or in the output stream.

In order to describe GPM and its operation this chapter is divided into two sections. The first describes the stack, the pointers to the stack, and the uses of each. The second section proceeds to explain the operation of GPM in relation to the various operation modes that it may assume.

THE STACK AND POINTERS

The stack consists of the macro definitions, partially evaluated macro-calls, and strings of characters resulting from completed macro-calls. These items are kept organized with single-linked lists and are referred to as chains. Accessing the stack is by means of pointers. The bottom of the stack contains, as a chain, the primitive macro definitions. The purpose of the primitive macros is to perform the operations of macro definition, redefining macro definitions, control of the evaluation of the source stream, and simple numerical conversions. The rest of the stack contains the items formed as the program executes.

The stack is composed of sequentially arranged elements, or items, called cells. Each contains either a character, a pointer (which is the address of another cell on the stack), or a cell count (which is the length of a string). Since the basic objects handled by GPM are strings of characters with various lengths, several cells are required to

Table 1

The Primitive (Control) Symbols and Their Functions

Symbol	Purpose	Description of Results
$	The dollar sign initiates a macro-call.	When the dollar sign is encountered, the F-chain is pushed-down and GPM prepares to evaluate the parameters of a new macro-call by entering the Parameter Evaluation Mode.
,	The comma is used to separate the parameters of a macro-call.	When the comma is encountered the evaluation of a particular actual parameter is terminated in order that the length of the parameter may be placed on the stack.
;	The semi-colon terminates a macro-call.	The semi-colon signals that the evaluation of the actual parameters on top of the F-chain is finished and causes the top member of the F-chain to be pushed-down onto the top of the P-chain. GPM now enters the Macro-expansion mode.
¬	The not symbol indicates a formal parameter in a macro-definition.	The Parameter Substitution Mode is initiated in which an actual parameter is substituted from the P-chain for the formal parameter that is specified.

Table 1 (continued)

Symbol	Purpose	Description of Results
< >	The less than and greater than symbols form the quotes that are used as brackets	The "<" initiates the Quote Mode and increments a counter, Q, by one. The following string is copied to the target stream until the matching ">" is encountered and Q equals one.
?	The question mark terminates the macro definition in the table.	The top of the P-chain is popped up deleting any temporary macro-definitions. If the P-chain is not empty, GPM remains in the Macro-expansion Mode. If the F-chain is still not completed GPM enters the Parameter Evaluation Mode. If the P and F chains are empty GPM reinitiates the Copy Mode.
#	The 0-6-9 punch signals the "end of line".	When the 0-6-9 punch is encountered in the input stream the next data card is read. In the "echo check" a pound sign, #, is substituted since the 0-6-9 punch is a non-printable character.

hold the string. These cells must be preceeded by a cell that contains the length of the string including itself. For instance the string ABC is represented by the four cells 4 A B C.

The length of the incomplete string when forming the target stream at the top of the stack is indicated by the pointer variable H. The string being scanned as the source stream is sent character by character to the output stream as long as H equals zero. Otherwise, the string is added to the stack. The pointer variable, C, designates the string of characters forming the source stream. If C equals zero the source is the input stream; otherwise, C points to the stack cell where the next character is to be scanned. The pointer S is used to indicate the next available cell on the stack.

The macro-definition table is contained in the environment list or E-chain and is headed by the pointer E. The primitive macros are stacked on the bottom of the E-chain with the name-value pairs formed during the processing of the macro-definitions placed on top. Temporary macro definitions, to be used only during a macro-call, will be added to the macro definition table and deleted from it upon completion of the macro-call. Any permanent definitions

that might have been formed on top of the stack are copied back onto the E-chain. In other words, the difference between permanent and temporary macro definitions is whether or not the macro definition remains on the E-chain after completion of the macro-call.

Partially evaluated parameters of a macro-call comprise the F-chain. If a nested macro-call is encountered during evaluation, the F-chain is pushed down and the pointer to the chain is appropriately adjusted. The F-pointer contains three parts, unlike the pointers above. The first part contains the value of H when a macro-call occured. The second is used to point to the next member of the F-chain. The third is set to zero and exists only for compatibility with the pointer to the P-chain.

The fully evaluated parameters of a macro-call are stored on the P-chain. It is formed from the F-chain by popping up the top member of the F-chain and pushing the popped-up member down on the P-chain. The P-pointer requires three items as does the F-pointer. The first of which is the length of the evaluated parameter list including the terminating symbol, the marker. The second and third contain the values of P and C, respectively, to be used when the macro-call is completed.

OPERATION OF GPM

GPM begins operation in the Copy Mode which consists of scanning from left to right the sequences of symbols in the input stream. In the Copy Mode, GPM directly copies the symbols to the output stream in their literal form, unlike other transliterative processors that perform transliterative encoding. Upon encountering a primitive control character in the input stream, the scan is interrupted and control is switched to another mode of operation.

Encountering a dollar sign in the input stream signals a macro-call and GPM prepares for the evaluation of that macro. A macro-call is formed by a dollar sign followed by the macro name and the actual parameter list, if any. A comma is used as a separating symbol if the macro-call contains more than one parameter because in GPM the macro name itself can be considered to be a parameter. The terminating symbol of the macro-call is a semicolon which signals the end of the parameter list, and starts the procedure for entering the Macro-expansion Mode.

Nesting of macro-calls occurs when the actual parameter list contains other macro-calls. The only limitation

is that the domain and range of the functions do not overlap. Several examples of macro-calls are given in Figure 1.

When the dollar sign is scanned, the run-time stack is pushed down creating a new member of the F-chain and initiating the Parameter Evaluation Mode(PEM). The F-chain along with its corresponding pointer is used during the evaluation of the actual parameters in the macro-call. If one of the parameters contains or is a macro-call the F-chain on the stack is once more pushed down and PEM for that macro-call begins. Evaluation takes place from left to right with the results being sent to the target stream, the F-chain. For simple string values of the macro name and/or actual parameter strings the result is a string of characters. Enclosing the symbol strings in quotes will delay, if nested, or totally inhibit the evaluation of the string within. The evaluation of the actual parameters occurs before substitution for the formal parameters in the macro definition unlike most macrogenerators that evaluate afterwards.

Detecting a semicolon in the source stream signals the completion of the Parameter Evaluation Mode. The top member or element of the F-chain is popped up and then pushed down on the P-chain, such that the top of the F-chain becomes the top of the P-chain and the macro can now be expended, or if a primitive macro-call, executed.

`$MAC0;`	Macro name only
`$MAC1,PARM1;`	Macro Name with One Parameter
`$MAC2,PARM2,PARM3;`	Macro Name with Two Parameters
`$A,$A,B;;`	A Nested Macro-call

Figure 1

Macro-call Examples

The Macro-expansion Mode (MEM) is now entered. The source stream switches to the current list of macro definitions, the E-chain. The scan occurs in reverse chronological order (the newest entry to the oldest entry, the primitive macro DEF) for the macro name of the present macro-call. If the macro name is found the macro body in the E-chain forms a new source stream for the Macro-expansion Mode. If a negative integer is encountered during the scan of the E-chain a primitive macro is signaled. If the call is for a system macro it is then executed. If it is not a system macro or the macro is undefined and an error message is outputed.

During the scan of the macro body, detecting a "not" symbol, ¬ , initiates the Parameter Substitution Mode (PSM) where an actual parameter is substituted for the formal parameter in the macro body. The PSM substitutes the corresponding actual parameter from the P-chain into the target stream. Upon completion of PSM, the Macro-expansion Mode is restarted. If the actual parameter list is too short, an error message is printed. If the actual parameter list is too long (i.e., more parameters are supplied than are needed) the extra parameters are ignored. But, the latter allows for macro-definitions and other macro-calls to be part of another macro-call.

Processing of the macro body continues between these two modes until the marker, a question mark, is detected signaling the end of the macro-definition. The macro in its fully expanded form is sent to the target stream. If the Macro-call is nested, the target is the run-time stack. If it is the outer-

most macro-call or an unnested macro-call the results are sent to the output stream.

Completing the macro-call pops up the run time stack deleting the top member of the P-chain. If there are no more items on the P-chain the Copy Mode is re-entered. If a partially evaluated macro-call is on top of the P-chain the Parameter Evaluation Mode begins. If the top of the stack contains a completed parameter list the processor continues in the Macro-expansion Mode. Figure 2 contains several examples of macro-calls and their results.

One unique feature of GPM is that function definition is accomplished by means of a macro-call to the primitive macro DEF. The DEF macro-call requires two parameters, the macro name and the macro body of the macro being defined, and has the following form:

$DEF,NAME,⟨BODY⟩;

The string quotes enclosing the body of the macro prevent evaluation of this parameter. If no formal parameters are indicated or other macro-calls, then the quotes can be removed as evaluation would only result in the literal string itself.

In the macro body the formal parameters are preceeded by a "not", ¬ , symbol followed by an unsigned positive integer indicating the position of the actual parameter which will be substituted when the macro is called. The i-th actual parameter is thereby identified by ¬ i. The macro name is substituted when ¬ 0 appears as a formal parameter. Examples of some macro definitions appear in Figure 3.

Macro-call	Definition	Result
$B;	Z	Z
$D,C;	D¬ 1D	DCD
$G,X,Y;	¬ 1¬ 2¬ 2¬ 1	XYYX
$D,$D,B;;		DDBDD

Figure 2

Macro-calls and Their Results

```
$DEF,QB,B;
$DEF,ABA,⟨Y¬ 1¬ 2Z⟩;
$DEF,ZZ,⟨$DEF,AC,⟨¬ 1AC⟩;,FF⟩;
```

Figure 3

Macro-definition Calls

The name-value pair forming the definition is attached to the E-chain, the only target stream. The body of each macro definition on the chain is terminated by an internal marker, the question mark, whose purpose is to signal the end of the macro body.

Until completion of the Parameter Evaluation Mode, the processing of a macro-call to DEF operates the same as for any other macro-call. Upon entering MEM from PEM, DEF is recognized as one of the primitive macros, and the Macro-definition Mode is entered. Then processing switches the target stream to the macro definition table.

GPM is capable of forming temporary definitions by including the macro definition call in the parameter list of another macro-call. This feature, a result of evaluation of parameters before substitution, allows a macro to be defined within a call to itself. For instance, the macro call

$$\$A,X,U,\$DEF,A,\langle\neg\ 1\neg\ 2\neg\ 1\rangle;;$$

defines a macro, A, by interrupting the PEM and them immediately uses that definition. The definition is temporary since upon completion of a macro-call, definitions which are part of the parameter list are deleted from the E-chain.

In addition to the DEF macro, GPM has five other primitive macros each initiated by a macro-call. The first is UPDATE whose purpose is to save space on the stack when revising a macro. This primitive macro takes two arguments and forms a name-value pair as does DEF, except that the value of its second parameter becomes the value of the macro body associated with

the first argument of UPDATE. Since the result is placed in the E-chain in the same position as the original definition, its length can not be longer than the original macro body. The length of the string is checked before altering the body and another terminator is added in case the new body is shorter. The length established by the first definition is not changed.

VAL requires one argument, the name of the macro being referenced. The call to VAL produces the macro body without evaluating it. The same effect can be gained by using a double set of quotes around the second argument of DEF and then calling that macro. The string representing the macro body will be copied to the target stream until the first marker is detected as more than one may exist if the macro was previously updated.

Simple arithmetic operations can be accomplished through the use of the BIN, DEC, and BAR primitive macros. External to internal and vice-versa integer conversion is handled by the use of BIN and DEC. BIN converts a digit string with a possible preceding sign into a signed binary integer. DEC forms a decimal digit string with sign if minus from a signed binary integer. Both require one argument whose maximum length depends on the cell size of the stack. BAR requires three parameters, the first of which is the signal character +, -, *, /, or R. The following two parameters are the binary numbers which are to be added, subtracted, multiplied, divided, or used to form the remainder, R, from a division.

A macro-call is not the only reason for an interruption in a processing mode. Detection of the opening quote, the less than symbol, in the source stream switches GPM into the Quote Mode. No change in the source and target streams takes place, but all symbols following the "<" are copied to the target without being evaluated. Any primitive symbols that would change the mode except the matching closing quote, a greater than symbol, are ignored. Since the outer layer of quotes is removed by not being copied to the target stream, evaluation of the string may only be delayed depending on where the quotes are used.

A counter, Q, keeps track of the possible nested quotes. Starting with an initial value of one, it is incremented and decremented by one for each opening and closing quote that is scanned. When the scan of the enclosed string is finished, that is when Q is once more equal to one, GPM then reinstates the mode that was interrupted by the Quote Mode. Care should be taken, as an unmatched opening quote gives no error message and the entire source stream could possibly be copied. Protection for an unmatched closing quote does exist and is monitored by an error message.

As can be seen from the above discussion, programs written for GPM can consist of infinitely nested macro-calls each of which is bracketed by the primitive symbols "$" and ";". These macro-calls may refer to a programmer defined macro or a system macro. Macro-definition is the result of a macro-call

to the primitive macro, DEF, and may result in either a temporary or a permanent definition. However, the evaluation of any macro-call or change of the present operation mode due to a control character being scanned in the source stream may be delayed or inhibited by the use of another set of bracketing characters, the quotes.[1]

Thus, the true power of GPM is derived from the use of only a few primitives and a single push-down stack. Therefore, recursive functions and conditional expressions become relatively simple to evaluate no matter how complicated they may appear.

[1]Mention should be made that these two bracketing sets operate independently and do not nest.

Chapter 3

IMPLEMENTATION

The implementation of GPM was described using CPL (Combined Programming Language) as the vehicle of presentation, although the actual program was written in machine code for the Titan Computer. The present implementation is in assembly code for the IBM 360/50. Since an entirely different computer was used, several modifications and additions were required in the original program description. These changes, such as the form and means of addressing the stack and the Input/Output routines, are described in this chapter. Since many of the implementation decisions depended on the programming language in which GPM would be written, the reasons for choosing the Macro-Assembler over the possibilities of FORTRAN and PL/I are first discussed. Essentially, the major role in this decision was played by the peculiarities of machine organization. Problems involved with this particular choice of programming language are then elaborated.

At first, one of the high level languages, PL/I or FORTRAN, appeared to provide the easiest mode of attack. Since strings of characters form the input stream for GPM, PL/I appeared to be the best of the high level languages for implementing GPM, even though FORTRAN does have limited character manipulation capabilities. A significant drawback involved in the rejection

of PL/I was time, both for compilation and execution. The PL/I compiler has proven to be laboriously slow and the resulting program uses various and assundry subroutines to execute its "string" functions. FORTRAN was rejected due to the problems involved in implementing the GPM primitive macros, and in having to "trick" FORTRAN in order to do some of the character manipulating operations.

Many problems were circumvented by using the IBM 360 language. The program would run faster than one written in PL/I and implementing the primitive macros and manipulating the symbols would be no problem. The final deciding factor was the actual "moves" required by the GPM processor in order to keep the stack compressed. As with the Titan implementation of GPM, these seemed to dictate the use of an assembler language, hence, the OS/360 Macro-Assembler was selected for programming the GPM processor.

The next considerations were dependent on the programming language chosen. First, how should the push-down stack be described? What control characters should be used? How should the primitive macros be implemented? Second, of what should the input and output routines consist? What media should be used for the input and output streams? In answering these questions, other questions resulted with respect to protection against errors and the ability to acquire further information about the workings of GPM.

After deciding to program GPM with the OS Macro-Assembler, the final representation of the stack cell had to be determined.

The stack cell could be a pointer, a cell count, or could hold a character. As a pointer, the cell required a full word, four bytes, in order to contain an address. Though this was larger than needed to store a character all the cells were designated as full words in order to simplify accessing the stack. Cell counts and addresses were then right-justified as full words. The symbols were left-justified with blanks comprising the rest of the word. An example of each follows in Figure 4.

The pointers or link variables used to mark the ends of the F-chain and P-chain were permanently assigned to a specific general purpose register. The pointers H, P, F, C, and S were assigned to the general purpose registers four through seven, and ten, respectively. Registers eight and nine were assigned as working registers and contained the values of the working variables A and W, respectively. Generally, A contains the character from the source stream being processed. The pointer E was not assigned a register since it was presumed to be used relatively infrequently. E is needed only for adding a new macro to, or in searching the environment chain.

Since the IBM 360 is a byte oriented machine with a full word being equivalent to four bytes, cell counts and pointers had to be adjusted accordingly. For instance, incrementing the C-pointer to the next character in the source stream was programmed as C=C+4 (i.e. LA C,4(C)) instead of C=C+1. A cell count of four for the string ABC actually had to

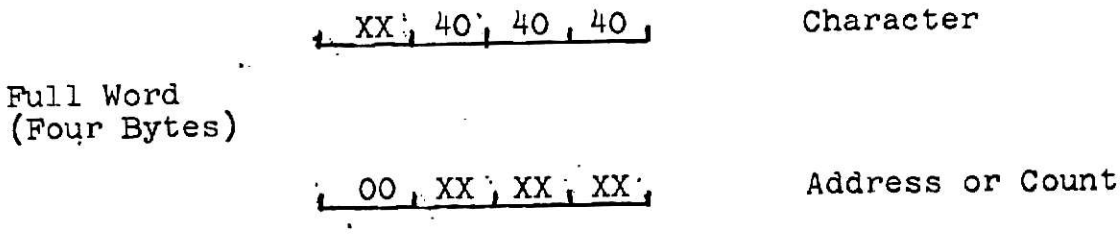
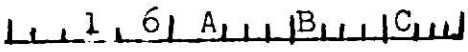


Figure 4

Storage Representation In
the Stack Cell

be programmed as sixteen since the string would be in the following form.[1]



This particular choice of cell representation formed one important limitation to programs written for this version of GPM. Since the internal representation of an external decimal digit is a zoned number, the integer indicating the position of the actual parameter to be substituted is stored as F0, F1,..., or F9 followed by three blanks. This limits the number of different parameters specified by a macro-definition to nine.

The choice of the primitive symbols used to implement GPM was dependent on the available character set of the computer. The particular control characters mentioned in Chapter 2 were chosen since they closely matched the descriptive ones in Strachey's paper and were present in the EBCDIC character set on the 360/50. The "not" symbol was substituted to indicate formal parameters. The string quotes did not exist so the inequality signs were used instead. The less than symbol formed the opening quote, and the greater than symbol formed the closing quote. A nonprintable internal marker was used to terminate macro-definitions on the E-chain. The present implementation substituted a question mark in order that the marker could be printed if necessary.

---

[1]The long marks are used to distinguish full word boundaries.

The question mark was also used at the end of the primitive macros. The object was to protect the primitive macros below the one that was being used during the "copy back" phase of operation. Another change involved in implementing the primitive macros was the use of a jump-table to find and execute the system macro. The primitives were marked with a negative integer that would be scanned when the E-chain was searched for a macro-definition. This negative number was then used to locate the address of the requested machine macro in the jump-table after the number had been changed to a comparable positive number.

One of the major considerations in implementing GPM was the design of the Input/Output routines. Card input was chosen to form the input stream and printer output was chosen for the output stream. The input procedure was handled by the RDSM routine, output by the PRNTMES and PRINTP routines. RDSM reads one input data card at a time and prints the data card as an "echo check". The data card is then scanned upon request from GPM and one character at a time is passed to be processed. Because many blanks at the end of a record are possible and are only copied to the out-put stream, a signal, the 0-6-9 punch, was added. This "end-of-line" punch causes a new input data card to be read when detected in the scan. The "end-of-line" character is not printable, therefore the pound sign, #, was substituted for the echo check.

The idea of continuous output is not destroyed by this addition. Basically, the 0-6-9 punch was an addition to

increase efficiency. However, the blanks should become part of the output stream to increase the readability of the output. To accomplish this another primitive macro, called EOB, was added. No parameters were required. The function of this primitive was simply to indicate that a new output record was to be started. Placing the macro-call to EOB before a symbol string resulted in the output string being placed at the beginning of a new output record.

The EOB macro-call sent the message to start a new output record to the PRNTMES routine. Since GPM sends only one character at a time to the output stream, PRNTMES waited until the record was full or an EOB macro-call occured before sending the output record to PRINTP. PRNTMES also set up any error messages that required printing and requested a memory dump for terminating errors.

The function of the PRINTP routine was to print the output record. At the same time the latest image was saved in case a terminating error occured. PRINTP ordered the output stream on an eighty column basis for direct comparison with the input stream.

Had one of the high level languages been chosen for writing GPM, the I/O routines would have been able to use the "built-in" I/O routines for the physical reading of an input data card and the actual printing of the output record. Since the Macro-Assembler Language was used these routines had to be written. They were designed as separate routines providing isolation

for the program, and were not included in Appendix I since they are strictly machine dependent.

Chapter 4

SUMMARY

The GPM processor implemented on the IBM 360/50 was debugged and tested by using some of the macro examples presented in Strachey's paper along with some additional macros so as to include recursive functions and conditional expressions. The results from these tests together with an "echo check" of the input stream can be found in Appendix II. The more important tests and subsequent results are discussed in this chapter and are used to outline the operating features of GPM. Program changes that were implemented due to testing other than the examples presented by Strachey are discussed along with the particular test. The last section of the chapter contains suggestions for additional machine macro's and program features that due to limited time were not added.

Testing began by checking the Copy Mode. A simple input string was given and then directly copied to the output stream to be printed. A simple macro definition, 5 BLANKS, was formed having no formal parameters and resulting in a simple output string. The definition is $DEF,5 BLANKS,ØØØØØ; where the "Ø" represents a blank. After defining 5 BLANKS the Copy Mode was reinstated and given a simple macro-call to 5 BLANKS which also involved two simple numeric strings. For instance, the test consisted of the input string,

12$5 BLANKS;34

and resulted in the output string 12bbbbb34. Succeeding tests were built in a similar manner. The rest of the chapter contains only a discussion of the tests.

One of the first tests was designed to check the isolation between permanent and temporary macro definitions. A permanent macro, A, was defined by $DEF,A,<A¬ 1A>; requiring one parameter. Later, a macro-call to A was given containing a temporary definition of A which required two parameters. The macro-call as used by Strachey[1] was

$A,X,U,$DEF,A,<¬ 1¬ 2¬ 1>;;

and executed giving the string XUX according to the later definition as expected.[2] To check that this second definition was removed from the E-chain a series of five macro-calls to A were formed;

1. $A,C;
2. $A,ACA;
3. $A,$A,C;;
4. $A,XDX;
5. $A,$A,D,X,$DEF,A,<¬ 2¬ 1¬ 2>;;;

---

[1]Strachey, p. 227

[2]The Macro-definition Mode has no protection against redefining the primitive macros. Any macro that had the same name as a primitive macro would be placed above the primitive on the E-chain, and hence be used as the macro-definition.

The results from these strings were:

1. ACA
2. AACAA
3. AACAA
4. AXDXA
5. AXDXA

obviously using the first definition of A and proving the second and temporary definition was temporary.

Mention should be made that the third macro-call in the series, $A,$A,C;;, involved a nested macro-call. The result of the inner macro-call is ACA which then forms the parameter for the second macro-call as in the second example in the series. In the fifth macro-call, $A,$A,D,X,$DEF,A,⟨¬ 2¬ 1¬ 2⟩;;;, the inner macro-call to A is a temporary definition and produces the result XDX, which in turn is the parameter in the outer macro-call to A where the permanent definition of A is used.

The idea of temporary definition was extended one step further by being included in a macro-definition. For this the successor of a digit, r, where r is a positive integer from zero to nine was used. The SUC macro as described by Strachey is defined as

$DEF,SUC,⟨$1,2,3,4,5,6,7,8,9,10,$DEF,1,⟨¬⟩,¬ 1;;⟩;

making use of the temporary definition of one. On the E-chain, SUC would have the following form.

$1,2,3,4,5,6,7,8,9,10,$DEF,1,⟨¬⟩¬ 1;;

The $SUC,2; macro-call would result in a temporary macro-call to "1" of the form

$DEF,1,¬ 2;

since the actual parameter "2" would be substituted for the formal parameter ¬ 1. The second parameter of the macro-call to "1" is "3"; therefore, "3" is the character placed in the out-put stream.

Since the E-chain is searched in reverse chronological order, the latter of two temporary definitions for the same macro is used. This forms the basis for GPM's ability to evaluate conditional expressions. A conditional expression consists of two temporary definitions nested within a macro-call which forms the false-true results of the expression. If the name of a given macro-call evaluates to the same macro-name as the second definition the later definition is used. This is equivalent to a true statement resulting. If the macro-names are not the same after evaluation, the case of a false statement, the former definition is used. For instance, given the macro-call

$X,$DEF,X,⟨G⟩;$DEF,Y,⟨H⟩;;

the result is G if X and Y are not equivalent, or H, if X and Y are equivalent where G, H, X, and Y are character strings.

To illustrate the conditional expression capabilities of GPM the SUCCESSOR macro (Figure 5) as described by Strachey was used. The SUCCESSOR macro finds the successor of a two digit positive integer. The successor of the second digit is found

unless that digit is nine. If the second digit is nine the successor of the first digit is found and added to the output stream followed by a zero.

The two digits form the first and second actual parameters in a macro-call to SUCCESSOR. The macro-call $SUCCESSOR,7,5; would result in the following macro-calls being formed.

$5,$DEF,5,7⟨$SUC,⟩5⟨;⟩⟩;$DEF,9, $SUC,⟩¬ 1⟨;,0⟩;;

In evaluating the macro-call to "5", a macro-call $SUC,5; would be evaluated returning the digit "6". The result of "7,5" would then be "7,6". If the digit string was "6,9" then the macro-call

$9,$DEF,9,6,⟨$SUC,⟩5⟨;⟩ ;$DEF,9,⟨$SUC,⟩¬ 1⟨;,0⟩;;

would result. Since the macro-name matches that of the second definition, which is on top of the E-chain, the second definition is used. The definition of "9" results in a macro-call to SUC whose parameter is "6". The digit returned would be "7" which is placed on the output string followed by ",0".

For testing the recursive aspect of GPM the factorial function was chosen. The design of the factorial macro, named FACT, involved the use of the primitive macros BIN, DEC and BAR and the programmer defined decimal arithmetic macros "*" and "-". The actual macro definition is given in Figure 6. FACT requires one parameter, N, which is an unsigned integer between zero and nine.

A macro-call of $FACT,N; results in a macro-call to "*" with parameters $FACT,(N-1); and N causing another macro-call

```
$¬ 2,$DEF,¬ 2,¬ 1,⟨$SUC,⟩¬ 2⟨;⟩;$DEF,9,⟨$SUC,⟩¬ 1⟨;,0⟩;;
```

Figure 5

Definition of the SUCCESSOR-Macro

```
$$DEC,$BIN,¬ 1;;$DEC,$BIN,¬ 1;;$DEC,$BIN,¬ 1;;,
⟨$¬ 1,¬ 1,$DEF,-1,⟨$*,$FACT,$-,¬1,1;;,¬ 1;⟩;
$DEF,1,⟨1⟩;; $DEF,0,⟨1⟩;;
```

Figure 6

Definition of the FACT Macro

to FACT. This process continues until one factorial or zero factorial, in the case that N equaled one originally, is reached. These return a value of one allowing the inner-most macro-call to "*" to be executed. For example, three factorial would generate the following:

$*,$FACT,2;3,;$*,$FACT,1;,2;

Since, $FACT,1; results in the value one, $*,1,2; can be executed giving the value "2". This then allows $*,2,3; to be executed giving "6" as the final result.

Tests were run with N varying from zero to nine. GPM performed the evaluation with ease once the macro definition was correctly determined. One flaw in GPM was discovered when an error in defining the factorial algorithm caused the stack to be filled and then proceeded to fill memory. To avoid this occuring again a "length-of-stack" check was added. Before the next stack cell was used, the S-pointer (which pointed to the next available cell on the stack) was compared with the specified length of the stack multiplied by four since each cell was four bytes in length. If the length was exceeded, the program was terminated.

Although GPM does contain a set of error diagnostics a more thorough set should be added to these. If GPM were to be used in a classroom situation or by fairly inexperienced programmers, those who have only worked with high level languages, such additions would be manditory. At this time such errors as an unquoted not symbol in the input stream or an unmatched

semicolon in the definition string are flagged with an error message and treated as though quoted. Processing then continues in order that as many errors as possible can be found at one time. Other errors cause termination of the processing and a memory dump to be given. Termination occurs for the errors of:

1. Impossible character (negative) as the argument of a number (nonexistant parameter).

2. Not enough arguments in the macro-call.

3. A question mark in a meaningless place.

4. Undefined macro-name.

5. Wrong exit taken (i.e. C≠H≠0), probably due to an unmatched ")".

6. The string for UPDATE to substitute being too long.

7. The argument for BIN containing a non-digit other than a sign.

Before termination a general monitor routine is used to print the macro-name and the actual parameters on the P and F chains.

Little information is available regarding which routines in GPM use most of the time. Two counters, KENDFN7 and KENDFN8, have been added to the "copy back" routine, ENDFN, at this time. These counters are used to count the number of times the permanent macro definitions on the E-chain have been recopied when temporary macro definitions are deleted. Other counters added to the other routines would give further information about the routines used by GPM. A system macro could then be used to send the appropriately labeled counts to the output stream upon

request by the programmer. At the time of this report the system macro is not available. KENDFN7 and KENDFN8 must be found in the memory dump in order to obtain the "copy back" count.

Another useful debugging aid would be to list all the macro-names on the E-chain. This macro should include an option to list the macro definitions along with the macro-names. This macro would be used as a result of a specific programmer request and not due to an error encountered by GPM. This feature would aid the programmer in finding an error possibly caused by re-definition of a macro-name or updating a macro with the UPDATE system macro. Furthermore, additional information as to the status of the environment chain could be obtained.

The above suggestions for additions to GPM were the consequence of the testing performed on GPM using both the macros presented by Strachey and additional programmer defined macros. No difficulty was found in using Strachey's macros, and they executed giving the predicted output strings. The programmer defined macros, once correctly defined, also were easily handled by GPM. It should be noted, though, that execution speed of GPM is slow due to the procedure used in processing the symbol stream. The successful implementation of these macros indicated that GPM does satisfy its specifications and has proven to be fully capable of evaluating functions no matter how deeply they are nested.

BIBLIOGRAPHY

Barron, D. W. and others. "The Main Features of CPL," The Computer Journal, VI 2 (July, 1963), 134-143

Strachey, C. "The General Purpose Macrogenerator," The Computer Journal, VIII 3 (October 1965), 225-241.

Wegner, Peter. Programming Languages, Information Structures, and Machine Organization. New York: McGraw-Hill, Inc., 1968

APPENDIX I

Listing of the Program for GPM

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    2          MACRO
                                    3 &NM      PRNTI &A
                                    4          CNOP  0,4
                                    5 &NM      BAL   1,*+12
                                    6          DC    A(&A)
                                    7          DC    V(PRNTBUFI)
                                    8          L     15,*-4
                                    9          BALR  14,15
                                   10          MEND
                                   11 *
                                   12          MACRO
                                   13 &NM      PRNTO &A
                                   14          CNOP  0,4
                                   15 &NM      BAL   1,*+12
                                   16          DC    A(&A)
                                   17          DC    V(PRNTBUFO)
                                   18          L     15,*-4
                                   19          BALR  14,15
                                   20          MEND
                                   21 *
                                   22          MACRO
                                   23 &NM      PRNT  &A
                                   24          CNOP  0,4
                                   25 &NM      STM   0,1,*+12
                                   26          BAL   1,*+20
                                   27          DC    A(&A)
                                   28          DC    A(0)
                                   29          DC    A(0)
                                   30          DC    A(PRNTMES)
                                   31          L     15,*-4
                                   32          BALR  14,15
                                   33          LM    0,1,*-18
                                   34          MEND
                                   35 *
                                   36          MACRO
                                   37 &NM      BALITEM
                                   38          CNOP  0,4
                                   39 &NM      ST    1,*+12
                                   40          BAL   1,*+12
                                   41          DC    A(*+4)
                                   42          DC    F'0'
                                   43          BAL   14,ITEM
                                   44          MEND
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

0000C0                            46 SYMBOL   CSECT
                                  47 *
                                  48 ****  EQUATE THE REGISTERS
                                  49 *
0000C4                            50 H        EQU   4
0000C5                            51 P        EQU   5
0000C6                            52 F        EQU   6
0000C7                            53 C        EQU   7
0000C8                            54 A        EQU   8
0000C9                            55 W        EQU   9
0000CA                            56 S        EQU   10
                                  57 *
000FA0                            58 LNTHST   EQU   4000          LNTHST -  LENGTH OF THE STACK
                                  59 *
                                  60          SAVE  (14,12),,*    SAVE REGISTERS
000000 47F0 F00C          0000C   61+         B     12(0,15) BRANCH AROUND ID
0000C4 06                         62+         DC    AL1(6)
0000C5 E2E8D4C2D6D3               63+         DC    CL6'SYMBOL' IDENTIFIER
0000CB 00
00000C 90EC D00C          0000C   64+         STM   14,12,12(13) SAVE REGISTERS
                                  65 *
000010 05C0                       66          BALR  12,0          ESTABLISH  REG 12  AS THE BASE REGISTER
000012                            67          USING *,12
                                  68 *
000012 41B0 C0AE          000C0   69          LA    11,SAVETH     GET ADDRESS OF SAVE AREA
000016 50B0 D008          00008   70          ST    11,8(,13)     STORE FORWARD POINTER
00001A 50D0 B004          00004   71          ST    13,4(,11)     STORE BACKWARD POINTER
00001E 18DB                       72          LR    13,11         SET UP REGISTER FOR SAVE AREA
                                  73 *
000020 58B0 C076          00088   74          L     11,=A(STACK)  REG 11  POINTS TO THE STACK -CSECT-
002488                            75          USING STACK,11
                                  76 *
0000C0                            77          USING SAVETH,13     -KLUDGE- FOR ADDRESSABILITY
                                  78 *
                                  79 ****SET THE -TIMER-
                                  80 *
                                  81          STIMER  TASK,TIMEOUT,BINTVL=TIME
000024 4110 C06E          00080   82+         LA    1,TIME LOAD PARAMETER REG 1
000028 4100 C08E          000A0   83+         LA    0,TIMEOUT LOAD PARAMETER REG 0
00002C 41E0 0010          00010   84+         LA    14,16(0,0) LOAD FLAG BYTE
000030 89E0 0018          00018   85+         SLL   14,24(0) SHIFT TO HI-ORDER BYTE
000034 160E                       86+         OR    0,14 AND PACK WITHEXIT ADDR
000036 0A2F                       87+         SVC   47 ISSUE STIMER SVC
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SCURCE STATEMENT
                                     89 ****  ZERO THE STACK ( -ST- )
                                     90 *
000038 986A C07A             0008C   91 SETSTO   LM    6,10,=A(0,0,4,LNTHST,LPLNST)
00003C 5067 BA60             02F18   92 LPLNST   ST    6,ST(7)
000040 1A78                          93          AR    7,8
000042 069A                          94          BCTR  9,10
                                     95 *
                                     96 ****  MOVE THE MACHINE MACRO'S TO THE STACK
                                     97 *
000044 D2EF BA60 B8E0 02F18 02D98    98          MVC   ST(60*4),STST                    00000860
                                     99 *
                                    100 ****  SET UP -S- AND -E-
                                    101 *
00004A 4130 BB30             02FE8  102          LA    3,XXMAC
00004E 5030 B068             02520  103          ST    3,EE
                                    104 *
000052 41A0 BB50             03008  105          LA    S,XXTOP
000056 50A0 B050             02508  106          ST    S,SS
                                    107 *
                                    108 ****  SET -H-, -P-, -F-, AND -C- TO ZERO
                                    109 *
00005A 1B44                         110          SR    H,H              H <--- 0
00005C 5040 B008             024C0  111          ST    H,HH
000060 1B55                         112          SR    P,P              P <--- 0
000062 5050 B014             024CC  113          ST    P,PP
000066 1B66                         114          SR    F,F              F <--- 0
000068 5060 B020             024D8  115          ST    F,FF
00006C 1B77                         116          SR    C,C              C <--- 0
00006E 5070 B02C             024E4  117          ST    C,CC
                                    118 *
                                    119 ****  SET -Q-  ( QQ )  TO  -1-
                                    120 *
000072 4120 0001             00001  121          LA    2,1              Q <--- 1
000076 5020 B05C             02514  122          ST    2,QQ
                                    123 *
00007A 47F0 D048             00108  124          B     START
                                    125 *
00007E 0000
000080 C0000BB8                     126 TIME     DC    F'3C00'          TIME = 30 SECS.
                                    127 *
000088                              128          LTORG
000088 C0002488                     129                =A(STACK)
00008C 0000000000000000             130                =A(0,0,4,LNTHST,LPLNST)
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0000A0                             132          USING *,15          SET -15- AS BASE REGISTER
                                   133 *
                                   134 TIMEOUT  SAVE  (14,12),,*    SAVE REGISTERS
0000A0 47F0 F00C           0000C   135+TIMEOUT  B     12(0,15) BRANCH AROUND ID
0000A4 07                          136+         DC    AL1(7)
0000A5 E3C9D4C5D6E4E3              137+         DC    CL7'TIMEOUT' IDENTIFIER
0000AC 90EC D00C           0000C   138+         STM   14,12,12(13) SAVE REGISTERS
                                   139 *
                                   140          ABEND 66,DUMP       -TIMER- ABEND
0000B0                             141+         CNOP  0,4
0000B0 47F0 F018           000B8   142+         B     *+8 BRANCH AROUND CONSTANT
0000B4 80                          143+         DC    AL1(128) DUMP/STEP CODE
0000B5 000042                      144+         DC    AL3(66) COMPLETION CODE
0000B8 5810 F014           000B4   145+         L     1,*-4 LOAD CODES INTO REG 1
0000BC 0A0D                        146+         SVC   13 LINK TO ABEND ROUTINE
                                   147 *
                                   148          DROP  15
                                   149 *
                                   150 *
                                   151 *
                                   152 ****    -SAVETH- IS PLACED HERE TO ALLOW -13- TO SPAN AS PART OF THE
                                   153 ****      ADDRESSABILITY -KLUDGE-.
                                   154 *
0000BE 0000
0000C0 C000000000000000            155 SAVETH   DC    18A(0)        SAVETH
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    157 ****    THE BASIC OR MAIN SCANNING CYCLE IS THE FOLLOWING.
                                    158 *
                                    159          PRINT NOGEN
                                    160 *
0001C8                              161          DS    0H
0001C8                              162 START    EQU   *                START
0001C8 900F B078            02530   163          STM   0,15,SVSTART
                                    164 *
0001CC FA30 B788 D10C 02C40 001CC   165          AP    KTSTART,=P'1'  KTSTART <--- KTSTART + 1                00001352
                                    166 *                                                                      00001354
000112 45E0 D18E            0024E   167          BAL   14,NEXTCH      GET NEXT CHARACTER
000116 5080 D0FC            001BC   168          ST    A,AA1
                                    169 *
00011A 954C B038      024F0         170          CLI   AA,C'<'
00011E 4770 D072            00132   171          BNE   START1
                                    172 *
000122 5820 B05C            02514   173          L     2,QQ           Q  <---  Q + 1
000126 4120 2001            00001   174          LA    2,1(,2)
00012A 5020 B05C            02514   175          ST    2,QQ
00012E 47F0 D0BA            0017A   176          B     Q2
                                    177 *
000132 9558 B038      024F0         178 START1   CLI   AA,C'$'          $
000136 4780 D2FA            003BA   179          BE    FN
                                    180 *
00013A 956B B038      024F0         181          CLI   AA,C','          ,
00013E 4780 D42A            004EA   182          BE    NEXTITEM
                                    183 *
000142 955E B038      024F0         184          CLI   AA,C';'          ;
000146 4780 D512            005D2   185          BE    APPLY
                                    186 *
00014A 955F B038      024F0         187          CLI   AA,C'¬'          ¬
00014E 4780 D47E            0053E   188          BE    LOADARG
                                    189 *
000152 956F B038      024F0         190          CLI   AA,C'?'          ?  -  INTERNAL MARKER
000156 4780 D1F2            002B2   191          BE    ENDFN
                                    192 *
00015A 956E B038      024F0         193          CLI   AA,C'>'          >
00015E 4780 D40E            004CE   194          BE    EXIT
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000162                              196 COPY     EQU   *              COPY
000162 900F B0B8             02570  197          STM   0,15,SAVECOPY
                                    198 *
000166 5080 D104             001C4  199          ST    A,AACOPY       OUTPUT CHARACTER
00016A 45E0 D10E             001CE  200          BAL   14,LOAD
                                    201 *
00016E 5820 B05C             02514  202 SCAN     L     2,QQ           DOES  Q = 1
000172 5920 D108             001C8  203          C     2,=F'1'
000176 4780 D048             00108  204          BE    START
                                    205 *
00017A 45E0 D18E             0024E  206 Q2       BAL   14,NEXTCH      GET NEXT CHARACTER
00017E 5080 D100             001C0  207          ST    A,AA2
                                    208 *
000182 954C B03B      024F0         209          CLI   AA,C'<'        <
000186 4770 D0DA             0019A  210          BNE   START2
                                    211 *
00018A 5820 B05C             02514  212          L     2,QQ           Q  <---  Q + 1
00018E 4120 2001             00001  213          LA    2,1(,2)
000192 5020 B05C             02514  214          ST    2,QQ
000196 47F0 D0A2             00162  215          B     COPY
                                    216 *
00019A 956E B03B      024F0         217 START2   CLI   AA,C'>'        >
00019E 4770 D0A2             00162  218          BNE   COPY
                                    219 *
0001A2 5820 B05C             02514  220          L     2,QQ           Q  <--- Q - 1
0001A6 5B20 D108             001C8  221          S     2,=F'1'
0001AA 5020 B05C             02514  222          ST    2,QQ
                                    223 *
0001AE 5920 D108             001C8  224          C     2,=F'1'        DOES  Q = 1
0001B2 4780 D048             00108  225          BE    START           ... YES
                                    226 *
0001B6 47F0 D0A2             00162  227          B     COPY            ... NO
                                    228 *
0001BA 0000
0001BC 00000000                     229 AA1      DC    F'0'           AA1
0001C0 0000C000                     230 AA2      DC    F'0'           AA2
0001C4 00000000                     231 AACOPY   DC    F'0'           AACOPY
0001C8                              232          LTORG
0001C8 00000001                     233                =F'1'
0001CC 1C                           234                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    236 ****    THE ROUTINE "LOAD" DISPOSES OF THE CHARACTER IN "A".
                                    237 *
0001CE                              238          DS    0H
0001CE                              239 LOAD     EQU   *               LOAD
0001CE 900F B188             02670  240          STM   0,15,SAVELOAD
                                    241 *
0001D2 FA30 B79C D18C 02C54 0024C   242          AP    KTLOAD,=P'1'    KTLOAD <--- KTLOAD + 1                00002072
                                    243 *                                                                      00002073
0001D8 1244                         244          LTR   H,H             DOES  H = 0
0001DA 4780 D142             00202  245          BZ    WRTSYM            .. YES
                                    246 *
0001DE 508A 0000             00000  247          ST    A,0(S)            .. NO,   ST(S)  <---  A
0001E2 41AA 0004             00004  248          LA    S,4(S)          S  <---  S + 1
0001E6 50A0 B050             02508  249          ST    S,SS
0001EA 59A0 D180             00240  250          C     S,=A(ST+4*LNTHST)                                      00002132
0001EE 4720 D134             001F4  251          BH    STOOBIG1                                               00002133
                                    252 *                                                                      00002134
0001F2 07FE                         253          BR    14              RETURN
                                    254 *                                                                      00002142
                                    255 STOOBIG1 ABEND 71,DUMP         S TOO BIG IN LOAD                     00002143
                                    262 *
0002C2 5080 D178             00238  263 WRTSYM   ST    A,AALOAD
0002C6 5010 D168             00228  264          ST    1,LOAD1SV        SAVE REG 1
0002CA 50F0 D16C             0022C  265          ST    14,LOAD14SV      SAVE REG 14
0002CE D200 D174 D178 00234 00238   266          MVC   WRSYM,AALOAD
000214 4110 D184             00244  267          LA    1,=A(WS)
000218 58F0 D188             00248  268          L     15,=A(PRNTMES)
00021C 05EF                         269          BALR  14,15
00021E 5810 D168             00228  270          L     1,LOAD1SV        RESTORE REG 1
000222 58E0 D16C             0022C  271          L     14,LOAD14SV      RESTORE REG 14
000226 07FE                         272          BR    14               RETURN
                                    273 *
000228                              274 LOAD1SV  DS    F                LOAD1SV
00022C                              275 LOAD14SV DS    F                LOAD14SV
000230 0001                         276 WS       DC    H'1'             WS
000232 0000                         277          DC    H'0'
000234 40                           278 WRSYM    DC    CL1' '           WRSYM
000235 000000
000238 FFFFFFFF                     279 AALOAD   DC    F'-1'            AALOAD
00023C 000CC000                     280 AALOADIN DC    F'0'             AALOADIN
000240                              281          LTORG
000240 00006D98                     282                =A(ST+4*LNTHST)
000244 00000230                     283                =A(WS)
000248 00001D90                     284                =A(PRNTMES)
00024C 1C                           285                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    287 ****     THE ROUTINE "NEXTCH" READS THE NEXT CHARACTER
                                    288 ****     FROM THE CURRENT INPUT STREAM INTO "A".
                                    289 *
00024E                              290          DS    0H
00024E                              291 NEXTCH   EQU   *                NEXTCH
00024E 900F B1F8              026B0 292          STM   0,15,SVNEXTCH
000252 FA30 B7B0 D1F0 02C68 002B0 293          AP    KTNEXTCH,=P'1' KTNEXTCH <--- KTNEXTCH + 1      00002402
                                    294 *                                                           00002403
                                    295 *
000258 1277                         296          LTR   C,C              DOES   C = 0
00025A 4780 D1B0              00270 297          BZ    RDSYMBL            .. YES
                                    298 *
00025E 5887 0000              00000 299          L     A,0(C)            .. NO,   A  <---  ST(C)
000262 5080 B038              024F0 300          ST    A,AA
000266 5080 D1E0              002A0 301          ST    A,AANXTCH
00026A 4177 0004              00004 302          LA    C,4(C)           C  <---  C + 1
00026E 07FE                         303          BR    14                RETURN
                                    304 *
000270 50E0 D1D4              00294 305 RDSYMBL  ST    14,SVNXCH14
000274 5010 D1D8              00298 306          ST    1,SVNXCH1
000278 4110 D1F8              002A8 307          LA    1,=A(AA)
00027C 58F0 D1EC              002AC 308          L     15,=A(RDSM)
000280 05EF                         309          BALR  14,15
                                    310 *
000282 58E0 D1D4              00294 311          L     14,SVNXCH14
000286 5810 D1D8              00298 312          L     1,SVNXCH1
00028A 5880 B038              024F0 313          L     A,AA
00028E 5080 D1DC              002A0 314          ST    A,SVANXTCH
000292 07FE                         315          BR    14                RETURN
                                    316 *
000294                              317 SVNXCH14 DS    F                 SVNXCH14
000298                              318 SVNXCH1  DS    F                 SVNXCH1
00029C 000CC000                     319 SVANXTCH DC    F'0'             SVANXTCH
0002A0 000CC000                     320 AANXTCH  DC    F'0'             AANXTCH
0002A8                              321          LTORG
0002A8 C00024F0                     322                =A(AA)
0002AC 00001B08                     323                =A(RDSM)
0002B0 1C                           324                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    27 MAY 70
                                    326 ****     THE ROUTINE "ENDFN" AND THE MARKER CHARACTER "?".
                                    327 *
                                    328 ****       THE CHARACTER,"?", IS AN INTERNAL TERMINATOR AND NEVER
                                    329 ****        OCCURS IN THE INPUT STREAM.  IT IS INSERTED AUTOMATICALLY
                                    330 ****         AT THE END OF ARGUMENT LISTS (BY THE PRIMITIVE
                                    331 ****          CHARACTER "?") AND AT THE END OF DEFINITION STRINGS (BY
                                    332 ****           THE PRIMITIVE MACRO "DEF").  IT SHOULD NEVER BE OUTPUT.
                                    333 *
0002B2                              334          DS    0H
0002B2                              335 ENDFN    EQU   *                ENDFN
0002B2 900F B0F8          025B0     336          STM   0,15,SVENDFN     ..... SAVE REGS .....
                                    337 *
0002B6 FA30 B7C4 D2F8 02C7C 003B8   338          AP    KTENDFN,=P'1'    KTENDFN <--- KTENDFN + 1                00002722
                                    339 *                                                                        00002723
0002BC 1965                         340          CR    F,P              F .GT. P
0002BE 4720 DBD4          00C94     341          BH    MON5               ... YES
                                    342 *
0002C2 5820 B068          02520     343          L     2,EE              ... NO,   ST(S) <---  E
0002C6 502A 0000          00000     344          ST    2,0(S)
                                    345 *
0002CA 188A                         346          LR    A,S              A <--- S
                                    347 *
                                    348 ****  REG 2  IS -ALPHA-
                                    349 ****  REG 3  IS -BETA-
                                    350 *
0002CC 1825                         351          LR    2,P              -2- <--- P-1
0002CE 5820 D2E8          003A8     352          S     2,=F'4'
0002D2 5832 0000          00000     353          L     3,0(2)           -3-  <---  ST(P-1)
0002D6 1802                         354          LR    0,2              -0- <--- P-1 + ST(P-1)
0002D8 1A03                         355          AR    0,3
                                    356 *
0002DA 5818 0000          00000     357 ENDFN1   L     1,0(A)           -1-  <---  ST(A)
0002DE 1910                         358          CR    1,0
0002E0 4740 D236          002F6     359          BL    ENDFN2
                                    360 *
0002E4 18E1                         361          LR    14,1             -14-  <---  ST(A)
0002E6 1B13                         362          SR    1,3              ST(A) <--- ST(A) - ST(P-1)
0002E8 5018 0000          00000     363          ST    1,0(A)
0002EC 188E                         364          LR    A,14             A  <---  -14-
0002EE 5080 B038          024F0     365          ST    A,AA
0002F2 47F0 D21A          002DA     366          B     ENDFN1
                                    367 *
0002F6 1891                         368 ENDFN2   LR    W,1              W  <---  ST(A)
0002F8 5090 B044          024FC     369          ST    W,WW
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

0002FC 1992                        371 ENDFN3   CR    W,2
0002FE 47D0 D24E             0030E 372          BNH   ENDFN4
                                   373 *
0003C2 5899 0000             00000 374          L     W,0(W)        W  <---  ST(W)
000306 5090 B044             024FC 375          ST    W,WW
0003CA 47F0 D23C             002FC 376          B     ENDFN3
                                   377 *
0003CE 5098 C000             00000 378 ENDFN4   ST    W,0(A)        ST(A) .<---  W
000312 581A 0000             00000 379          L     1,0(S)        E  <---  ST(S)
000316 5010 B068             02520 380          ST    1,EE
                                   381 *
00031A 1244                        382          LTR   H,H
00031C 4780 D27A             0033A 383          BZ    ENDFN6
                                   384 *
000320 1945                        385          CR    H,P
000322 4720 D274             00334 386          BH    ENDFN5
                                   387 *
000326 5814 0000             00000 388          L     1,0(H)        ST(H)  <---  ST(H) - ST(P-1)
00032A 1813                        389          SR    1,3
00032C 5014 0000             00000 390          ST    1,0(H)
000330 47F0 D27A             0033A 391          B     ENDFN6
                                   392 *
000334 1B43                        393 ENDFN5   SR    H,3           H <--- H - ST(P-1)
000336 5040 B008             024C0 394          ST    H,HH
                                   395 *
00033A 1815                        396 ENDFN6   LR    1,P           -1- <--- P
00033C 5855 0000             00000 397          L     P,0(P)        P  <---  ST(P)
000340 5050 B014             024CC 398          ST    P,PP
000344 5870 1004             00004 399          L     C,4(,1)       C  <---  ST(P+1)
000348 5070 B02C             024E4 400          ST    C,CC
00034C 1BA3                        401          SR    S,3           S <--- S - ST(P-1)
00034E 50A0 B050             02508 402          ST    S,SS
000352 1882                        403          LR    A,2           A <--- P-1
000354 5080 B038             024F0 404          ST    A,AA
000358 5080 D2E0             003A0 405          ST    A,AENDFN
                                   406 *
00035C 1892                        407          LR    W,2           W <--- P-1 + ST(P-1)
00035E 1A93                        408          AR    W,3
000360 5090 B044             024FC 409          ST    W,WW
000364 5090 D2E4             003A4 410          ST    W,WENDFN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000368 9813 D2EC             003AC   412          LM    1,3,=A(4,ENDFN7,ENDFN8)
00036C 198A                          413 ENDFN7   CR    A,S
00036E 0783                          414          BCR   8,3
000370 5809 0000             00000   415          L     0,0(W)          ST(A)  <---  ST(W)
000374 5008 0000             00000   416          ST    0,0(A)
000378 1A81                          417          AR    A,1             A <--- A + 1
00037A 1A91                          418          AR    W,1             W <--- W + 1
00037C FA30 D2D8 D2F8 00398 003B8    419          AP    KENDFN7,=P'1'
000382 07F2                          420          BR    2
                                     421 *
                                     422 *
000384 5080 B038             024F0   423 ENDFN8   ST    A,AA            SAVE -A- AND -W-
000388 5090 B044             024FC   424          ST    W,WW
                                     425 *
00038C FA30 D2DC D2F8 0039C 003B8    426          AP    KENDFN8,=P'1'
                                     427 *
000392 47F0 D048             00108   428          B     START
                                     429 *
000398                               430          DS    0F
000398 0000000C                      431 KENDFN7  DC    PL4'0'          KENDFN7
00039C C000000C                      432 KENDFN8  DC    PL4'0'          KENDFN8
                                     433 *
0003A0 0000C000                      434 AENDFN   DC    A(0)            AENDFN
0003A4 0000C000                      435 WENDFN   DC    A(0)            WENDFN
0003A8                               436          LTORG
0003A8 00000004                      437                =F'4'
0003AC C0000004000036C               438                =A(4,ENDFN7,ENDFN8)
0003B8 1C                            439                =P'1'
```

```
   LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        27 MAY 70

                                     441 ****     THE ROUTINE "FN" AND THE PRIMITIVE CHARACTER "$".
                                     442 *
0003BA                               443          DS    0H
0003BA                               444 FN       EQU   *              FN
0003BA 900F B138             025F0   445          STM   0,15,SAVEFN
                                     446 *
0003BE FA30 B7D8 D354 02C90  00414   447          AP    KTFN,=P'1'     KTFN <--- KTFN + 1                00003862
                                     448 *                                                            00003863
0003C4 504A 0000             00000   449          ST    H,0(S)         ST(S)  <---  H
0003C8 506A 0004             00004   450          ST    F,4(S)         ST(S+1)  <---  F
0003CC 1B22                          451          SR    2,2
0003CE 502A C008             00008   452          ST    2,8(S)         ST(S+2)  <---  0
0003D2 502A 000C             0000C   453          ST    2,12(S)        ST(S+3)  <---  0
                                     454 *
0003D6 414A 000C             0000C   455          LA    H,12(S)        H  <---  S+3
0003DA 5040 B008             024C0   456          ST    H,HH
0003DE 416A 0004             00004   457          LA    F,4(S)         F  <---  S+1
0003E2 5060 B020             024D8   458          ST    F,FF
0003E6 41AA 0010             00010   459          LA    S,16(S)        S  <---  S+4
0003EA 50A0 B050             02508   460          ST    S,SS
0003EE 59A0 D350             00410   461          C     S,=A(ST+4*LNTHST)                              00003882
0003F2 4720 D33C             003FC   462          BH    STOOBIG2                                       00003883
                                     463 *
0003F6 47F0 D048             00108   464          B     START
                                     465 *                                                            00003902
                                     466 STOOBIG2 ABEND 72,DUMP        S TOO BIG IN FN                00003903
                                     473 *
000410                               474          LTORG
000410 00006D98                      475                =A(ST+4*LNTHST)
000414 1C                            476                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000415 00
000416 5010 D374             00434   478 JPIFMAR  ST    1,XMPIFMAR     JUMP IF MARKED
                                     479 *
00041A FA30 B7EC D3A0 02CA4  00460   480           AP    KTMARK,=P'1'   KTMARK <--- KTMARK + 1          00003932
                                     481 *                                                              00003933
000420 1211                          482           LTR   1,1            IS -1- MINUS
000422 47D0 D368             00428   483           BNH   SKPJIM          ... YES
000426 07FE                          484           BR    14              ... NO,  RETURN
                                     485 *
000428 1011                          486 SKPJIM    LPR   1,1
00042A 8910 0002             00002   487           SLL   1,2
00042E 5821 D378             00438   488           L     2,JMPTBL(1)
000432 07F2                          489           BR    2
                                     490 *
000434 C00CC000                      491 XMPIFMAR  DC    F'0'
                                     492 *
000438 C000C000                      493 JMPTBL    DC    A(0)           JUMP TABLE FOR MACHINE MACRO'S
00043C C000069A                      494           DC    A(DEF)
000440 C00006E6                      495           DC    A(VAL)
000444 0000073A                      496           DC    A(UPDATE)
000448 0000079A                      497           DC    A(BIN)
00044C C000085E                      498           DC    A(DEC)
000450 00000906                      499           DC    A(BAR)
000454 C000C982                      500           DC    A(EOB)
000458 000009AA                      501           DC    A(SIGN)          00004122
                                     502 *
000460                               503           LTORG
000460 1C                            504                 =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  27 MAY 70
                                    506 ****     'FINDX' IS THE ROUTINE USED TO SEARCH THE ENVIRONMENT CHAIN,
                                    507 ****      THE "E-CHAIN", FOR THE NAME OF THE MACRO.
                                    508 *
000462                              509          DS    0H
000462                              510 FINDX    EQU   *              FINDX
000462 900F B178            02630   511          STM   0,15,SVFINDX
                                    512 *
000466 FA30 B800 D40C 02CB8 004CC   513          AP    KTFINDX,=P'1'  KTFINDX <--- KTFINDX + 1                  00004172
                                    514 *                                                                       00004173
00046C 5880 B068            02520   515          L     A,EE           A  <---  E
000470 5080 B038            024F0   516          ST    A,AA
000474 1891                         517          LR    W,1             W = X   X IS IN REG 1
000476 5090 B044            024FC   518          ST    W,WW
                                    519 *
00047A 1B33                         520 LPFNDX   SR    3,3            R = 0   ( -3- IS -R- )
00047C 5809 0000            00000   521          L     0,0(W)         -0-  <---  ST(W)-1
000480 5B00 D408            004C8   522          S     0,=F'4'
                                    523 *
000484 4129 3000            00000   524 LPR      LA    2,0(W,3)       -2-  <---  LOC( ST(W+R) )
000488 4118 3004            00004   525          LA    1,4(A,3)       -1-  <---  LOC( ST(A+R+1) )
                                    526 *
00048C D503 2000 1000 00000 00000   527          CLC   0(4,2),0(1)    ST(W+R) = ST(A+R+1)
000492 4770 D3F0            00480   528          BNE   NEXTFNDX         ... NO
                                    529 *
000496 4130 3004            00004   530          LA    3,4(,3)          ... YES,   R  <---  R + 1
00049A 1930                         531          CR    3,0
00049C 47D0 D3C4            00484   532          BNH   LPR
                                    533 *
0004A0 5899 0000            00000   534          L     W,0(W)         W  <---  A + 1 + ST(W)
0004A4 4199 0004            00004   535          LA    W,4(W)
0004A8 1A98                         536          AR    W,A
0004AA 5090 B044            024FC   537          ST    W,WW
0004AE 07FE                         538          BR    14             RETURN
                                    539 *
0004B0 5888 0000            00000   540 NEXTFNDX L     A,0(A)         A  <---  ST(A)
0004B4 5080 B038            024F0   541          ST    A,AA
                                    542 *
0004B8 1288                         543          LTR   A,A
0004BA 4740 DCCE            00D8E   544          BL    MON7
                                    545 *
0004BE 47F0 D3BA            0047A   546          B     LPFNDX
                                    547 *
0004C8                              548          LTORG
0004C8 0000C004                     549                =F'4'
0004CC 1C                           550                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

0004CD 00
0004CE 1277                          552 EXIT     LTR   C,C
0004D0 4770 DD10             00DD0   553          BNZ   MON8
                                     554 *
0004D4 1244                          555          LTR   H,H
0004D6 4770 DD10             00DD0   556          BNZ   MON8
                                     557 *
                                     558 *
                                     559 *
                                     560 *
                                     561 ****  -FINISH- CONDITION LEADS TO -ABEND 13-
                                     562 *
                                     563          ABEND 13,DUMP        FINISH
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  27 MAY 70
                                    571 ****    THE ROUTINE "NEXTITEM" AND THE PRIMITIVE CHARACTER ",".
                                    572 *
0004EA                              573          DS    0H
0004EA                              574 NEXTITEM EQU   *              NEXTITEM
0004EA 900F B238             026F0  575          STM   0,15,SVNXTITM
                                    576 *
0004EE FA30 8814 D47C 02CCC 0053C  577          AP    KTNXTITM,=P'1' KTNXTITM <--- KTNXTITM + 1          00004692
                                    578 *                                                                  00004693
0004F4 1244                         579          LTR   H,H            DOES  H = 0
0004F6 4780 D0A2             00162  580          BZ    COPY            ... YES,   GO TO COPY
                                    581 *
0004FA 182A                         582          LR    2,S            ST(H)  <---  S - H - ST(H)
0004FC 1B24                         583          SR    2,H
0004FE 5B24 0000             00000  584          S     2,0(H)
0005C2 5024 0000             00000  585          ST    2,0(H)
                                    586 *
0005C6 1B33                         587          SR    3,3            ST(S)  <---  0
0005C8 503A 0000             00000  588          ST    3,0(S)
                                    589 *
0005CC 184A                         590          LR    H,S            H  <---  S
0005CE 5040 B008             024C0  591          ST    H,HH
                                    592 *
000512 41AA 0004             00004  593          LA    S,4(S)         S  <---  S + 1
000516 50A0 B050             02508  594          ST    S,SS
00051A 59A0 D478             00538  595          C     S,=A(ST+4*LNTHST)                                  00004852
00051E 4720 D468             00528  596          BH    STOOBIG3                                           00004853
                                    597 *
000522 47F0 D048             00108  598          B     START
                                    599 *                                                                  00004872
                                    600 STOOBIG3 ABEND 73,DUMP        S TOO BIG IN NEXTITEM              00004873
                                    607 *
000538                              608          LTORG
000538 00006D98                     609                =A(ST+4*LNTHST)
00053C 1C                           610                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    612 ****    THE ROUTINE "LOADARG" AND THE PRIMITIVE CHARACTER "¬".
                                    613 *
00053E                              614          DS    0H
00053E                              615 LOADARG  EQU   *              LOADARG
00053E 900F B278             02730  616          STM   0,15,SVLDARG
                                    617 *
000542 FA30 B828 D510 02CE0 005D0   618          AP    KTLDARG,=P'1'  KTLDARG <--- KTLDARG + 1                00004922
                                    619 *                                                                      00004923
000548 1255                         620          LTR   P,P            TEST -P-
00054A 4770 D498             00558  621          BNZ   LNXTCH
                                    622 *
00054E 1244                         623          LTR   H,H            TEST -H-
000550 4780 D0A2             00162  624          BZ    COPY
                                    625 *
000554 47F0 DAAE             00B6E  626          B     MON2
                                    627 *
000558 45EC D18E             0024E  628 LNXTCH    BAL   14,NEXTCH
00055C 5080 B038             024F0  629          ST    A,AA
                                    630 *
000560 4195 0008             00008  631          LA    W,8(P)         W <--- P+2
000564 5090 B044             024FC  632          ST    W,WW
                                    633 *
000568 4330 B038             024F0  634          IC    3,AA           -3-  <---  NUMBER( A )
00056C 5430 D508             005C8  635          N     3,=F'15'
000570 1233                         636          LTR   3,3
000572 4740 DB16             00BD6  637          BL    MON3
                                    638 *
000576 1B11                         639          SR    1,1            -1-  <---0   (-R- IS -1-)
000578 0630                         640          BCTR  3,0            -3-  <--- NUMBER(A)-1
                                    641 *
00057A 1913                         642 LDARGO   CR    1,3
00057C 4720 D4DA             0059A  643          BH    LDARG1
                                    644 *
000580 5829 0000             00000  645          L     2,0(W)         W  <---  W + ST(W)
000584 1A92                         646          AR    W,2
000586 5090 B044             024FC  647          ST    W,WW
                                    648 *
00058A 5920 B074             0252C  649          C     2,MARKER       ST(W) = MARKER
00058E 4780 DB5C             00C1C  650          BE    MON4            ... YES
                                    651 *
000592 4110 1001             00001  652          LA    1,1(,1)
000596 47F0 D4BA             0057A  653          B     LDARGO
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
00059A 5829 0000              00000   655 LDARG1   L ~    2,0(W)          -2-  <---  ST(W) - 1
00059E 5B20 D50C              005CC   656          S      2,=F'4'
0005A2 4130 0004              00004   657          LA     3,4             -3- <--- 4     ( -3- IS -R- )
                                      658 *
0005A6 1932                           659 LDARG2   CR     3,2
0005A8 4720 D048              00108   660          BH     START
0005AC 1819                           661          LR     1,W             A <--- ST(W+R)
0005AE 1A13                           662          AR     1,3
0005B0 5881 0000              00000   663          L      A,0(1)
0005B4 5080 B038              024F0   664          ST     A,AA
0005B8 45E0 D10E              001CE   665          BAL    14,LOAD
0005BC 4130 3004              00004   666          LA     3,4(,3)         -3- <--- -3- + 4
0005C0 47F0 D4E6              005A6   667          B      LDARG2
                                      668 *
0005C8                                669          LTORG
0005C8 C000C00F                       670                 =F'15'
0005CC C000C004                       671                 =F'4'
0005D0 1C                             672                 =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                  27 MAY 70
                                    674 ****    THE ROUTINE "APPLY" AND THE PRIMITIVE CHARACTER ";".
                                    675 *
0005C2                              676          DS    0H
0005C2                              677 APPLY    EQU   *               APPLY
0005C2 900F B2B8             02770  678          STM   0,15,SVAPPLY
                                    679 *
0005C6 FA30 B83C D5D8 02CF4 00698   680          AP    KTAPPLY,=P'1'   KTAPPLY <--- KTAPPLY + 1          00005492
                                    681 *                                                                  00005493
0005CC 1956                         682          CR    P,F             IS  P .GT. F
0005CE 4720 DA48             00808  683          BH    MON1             ... YES
                                    684 *
0005E2 1244                         685          LTR   H,H
0005E4 4780 DQA2             00162  686          BZ    COPY
                                    687 *
0005E8 5826 0000             00000  688          L     2,0(F)          -2-  <---  ST(F)
0005EC 18E6                         689          LR    14,F            -14- <--- F-1
0005EE 5BE0 D5D0             00690  690          S     14,=F'4'
0005F2 5830 E000             00000  691          L     3,0(,14)        -3-  <---  ST(F-1)
0005F6 180A                         692          LR    0,S             -0- <--- S-H
0005F8 1804                         693          SR    0,H
0005FA 181A                         694          LR    1,S             -1- <--- S-F-2
0005FC 1B16                         695          SR    1,F
0005FE 4110 1008             00008  696          LA    1,8(,1)
                                    697 *
0006C2 5056 0000             00000  698          ST    P,0(F)          ST(F)  <---  P
0006C6 5076 0004             00004  699          ST    C,4(F)          ST(F+1)  <---  C
0006CA 5010 E000             00000  700          ST    1,0(,14)        ST(F-1)  <---  -1-
0006CE 5004 0000             00000  701          ST    0,0(H)          ST(H)  <---  -0-
                                    702 *
000612 1843                         703          LR    H,3             H <--- -3-
000614 5040 B008             024C0  704          ST    H,HH
000618 1856                         705          LR    P,F             P <--- F
00061A 5C50 B014             024CC  706          ST    P,PP
00061E 1862                         707          LR    F,2             F <--- -2-
000620 5060 B020             024D8  708          ST    F,FF
                                    709 *
000624 5820 B074             0252C  710          L     2,MARKER        ST(S) <--- MARKER
000628 502A 0000             00000  711          ST    2,0(S)
                                    712 *
00062C 41AA 0004             00004  713          LA    S,4(S)          S <--- S + 1
000630 50A0 B050             02508  714          ST    S,SS                                               00005822
000634 59A0 D5D4             00694  715          C     S,=A(ST+4*LNTHST)                                  00005823
000638 4720 D5C0             00680  716          BH    STOOBIG4                                           00005824
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
00063C 1244                        718          LTR   H,H            H = 0
00063E 4780 D598              00658 719          BZ    AP1             ... YES
                                    720 *
000642 18E5                        721          LR    14,P            ... NO,   ST(H)  <---  ST(H) + ST(P-1)
000644 5BE0 D5D0              00690 722          S     14,=F'4'
000648 50E0 D5B8              00678 723          ST    14,P1APPLY
00064C 582C E000              00000 724          L     2,0(,14)
000650 5A24 0000              C0000 725          A     2,0(H)
000654 5024 C000              00000 726          ST    2,0(H)
                                    727 *
000658 4115 0008              00008 728 AP1      LA    1,8(P)          -1- <--- P+2
00065C 45E0 D3A2              00462 729          BAL   14,FINDX
000660 5090 D5BC              0067C 730          ST    W,WAPPLY
                                    731 *
000664 5819 0000              00000 732          L     1,0(W)          -1-  <---  ST(W)
000668 45E0 D356              00416 733          BAL   14,JPIFMAR
                                    734 *
00066C 4179 0004              00004 735          LA    C,4(W)          C <--- W + 1
000670 5070 B02C              024F4 736          ST    C,CC
                                    737 *
000674 47FC D048              00108 738          B     START
                                    739 *
000678 00000000                     740 P1APPLY    DC    F'0'            P1APPLY
00067C C0000000                     741 WAPPLY     DC    F'0'            WAPPLY
                                    742 *
                                    743 STOOBIG4   ABEND 74,DUMP        'S TOO BIG IN APPLY                    00006082
                                    750 *                                                                      00006083
000690                              751          LTORG
000690 C0000004                     752                =F'4'
000694 00006098                     753                =A(ST+4*LNTHST)
000698 1C                           754                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    
                                    756 ****    THE PRIMITIVE MACRO "DEF" HAS AN ORDINARY RESULT WHICH IS THE
                                    757 ****     NAME-VALUE PAIR, AND A SIDE EFFECT OF ATTACHING THIS RESULT
                                    758 ****      TO THE "E-CHAIN".
                                    759 *
00069A                              760          DS    0H
00069A                              761 DEF      EQU   *              DEF
00069A 900F B2F8            027B0   762          STM   0,15,SAVEDEF
                                    763 *
00069E FA30 B850 D624 02D08 006E4   764          AP    KTDEF,=P'1'    KTDEF <--- KTDEF + 1                  00006142
                                    765 *                                                                     00006143
0006A4 1815                         766          LR    1,P            -1-  <---  P-1
0006A6 5B10 D620            006E0   767          S     1,=F'4'
                                    768 *
0006AA 1244                         769          LTR   H,H            H = 0
0006AC 4780 D600            006C0   770          BE    DEF1           ... YES
                                    771 *
0006B0 5824 0000            00000   772          L     2,0(H)         ST(H)  <---  ST(H) - ST(P-1) + 6
0006B4 5B20 1000            00000   773          S     2,0(,1)
0006B8 4120 2018            00018   774          LA    2,24(,2)
0006BC 5024 0000            00000   775          ST    2,0(H)
                                    776 *
0006C0 4120 0018            00018   777 DEF1     LA    2,24           ST(P-1) <--- 6
0006C4 5020 1000            00000   778          ST    2,0(,1)
0006C8 5830 B068            02520   779          L     3,EE           ST(P+5) <--- E
0006CC 5035 0014            00014   780          ST    3,20(P)
0006D0 4115 0014            00014   781          LA    1,20(P)        E <--- P+5
0006D4 5010 B068            02520   782          ST    1,EE
                                    783 *
0006D8 47F0 D1F2            002B2   784          B     ENDFN
                                    785 *
0006E0                              786          LTORG
0006E0 C00CC004                     787                =F'4'
0006E4 1C                           788                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0006E6                             790          DS    0H
0006E6                             791 VAL      EQU   *               VAL
0006E6 900F B338                   792          STM   0,15,SAVEVAL
                                   793 *
0006EA FA30 B864 D678 02D1C 00738  794          AP    KTVAL,=P'1'     KTVAL <--- KTVAL + 1                00005402
                                   795 *                                                                    00005403
0006F0 4115 0018             00018 796          LA    1,24(P)         -1-  <---  P+6
                                   797 *
0006F4 45E0 D3A2             00462 798          BAL   14,FINDX
0006F8 5090 D668             00728 799          ST    W,SVWVAL
                                   800 *
0006FC 5829 0004             00004 801 VALI     L     2,4(W)          IS  ST(W+1) = MARKER
000700 5920 B074             0252C 802          C     2,MARKER
000704 4780 D1F2             002B2 803          BE    ENDFN            ... YES
                                   804 *
000708 1882                        805          LR    A,2              ... NO,   A  <---  ST(W+1)
00070A 5080 B038             024F0 806          ST    A,AA
00070E 5080 D66C             0072C 807          ST    A,SVAVAL
                                   808 *
000712 4199 0004             00004 809          LA    W,4(W)          W <--- W + 1
000716 5090 B044             024FC 810          ST    W,WW
00071A 5090 D670             00730 811          ST    W,WWVAL
                                   812 *
00071E 45E0 D10E             001CE 813          BAL   14,LOAD
                                   814 *
000722 47F0 D63C             006FC 815          B     VALI
                                   816 *
000726 C000
000728 C00CC000                    817 SVWVAL   DC    F'0'            SVWVAL
00072C C000C000                    818 SVAVAL   DC    F'0'            SVAVAL
000730 C000C000                    819 WWVAL    DC    F'0'            WWVAL
                                   820 *
000738                             821          LTORG
000738 1C                          822                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
00073A                              824          DS    0H
00073A                              825 UPDATE   EQU   *               UPDATE
00073A 900F B378                    826          STM   0,15,SVUPDATE
                                    827 *
00073E FA30 B878 D6D8 02D30 00798   828          AP    KTUPDATE,=P'1' KTUPDATE <--- KTUPDATE + 1          00006692
                                    829 *                                                                  00006693
000744 4115 0024                    830          LA    1,36(P)          -1-  <---  P+9
                                    831 *
000748 45E0 D3A2                    832          BAL   14,FINDX
00074C 5090 D6D0                    833          ST    W,SVWUPD
                                    834 *
000750 1885                         835          LR    A,P               A = P + 9 + ST(P+9)
000752 4188 0024                    836          LA    A,36(A)
000756 5A85 0024                    837          A     A,36(P)
00075A 5080 B038                    838          ST    A,AA
                                    839 *
00075E 5828 C000                    840          L     2,0(A)           ST(A) .GT. ST(W)
000762 5929 0000                    841          C     2,0(W)
000766 4720 DD3A                    842          BH    MON9              ... YES
                                    843 *
00076A 4120 0004                    844          LA    2,4              -2- (R) <--- 1
00076E 1818                         845          LR    1,A              -1- <--- A
000770 1839                         846          LR    3,W              -3- <--- W
000772 4110 1004                    847 UPDATE1  LA    1,4(,1)
000776 4130 3004                    848          LA    3,4(,3)
00077A D2C3 3000 1000 00000         849          MVC   0(4,3),0(1)      ST(W+R) <--- ST(A+R)
000780 5928 C000                    850          C     2,0(A)           R .GT. ST(A)
000784 4780 D1F2                    851          BE    ENDFN             ... YES
                                    852 *
000788 4120 2004                    853          LA    2,4(,2)          R <--- R + 1
00078C 47F0 D6B2                    854          B     UPDATE1
                                    855 *
000790 C00CC000                     856 SVWUPD   DC    F'0'             SVWUPD
                                    857 *
000798                              858          LTORG
000798 1C                           859                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    
                                    861 ****    THE PRIMITIVE MACRO "BIN" CONVERTS A DIGIT STREAM, POSSIBLY
                                    862 ****     PROCEDED BY A SIGN, INTO A SIGNED BINARY INTEGER.
                                    863 *
00079A                              864          DS    0H
00079A                              865 BIN      EQU   *                 BIN
00079A 900F B3B8           02870    866          STM   0,15,SAVEBIN
                                    867 *
00079E FA30 B88C D79C 02D44 0085C   868          AP    KTBIN,=P'1'       KTBIN <--- KTBIN + 1                00007012
                                    869 *                                                                      00007013
0007A4 1899                         870          SR    W,W               W  <---  0
0007A6 5090 B044           024FC    871          ST    W,WW
0007AA 4125 001C           0001C    872          LA    2,28(P)           -2-  <---  LOC( ST(P+7) )
0007AE 954E 2000  00000             873          CLI   0(2),C'+'
0007B2 4780 D70A           007CA    874          BE    BIN1
0007B6 9560 2000  00000             875          CLI   0(2),C'-'
0007BA 4760 D70A           007CA    876          BE    BIN1
0007BE 4185 001C           0001C    877          LA    A,28(P)           A <--- P+7
0007C2 5080 B038           024F0    878          ST    A,AA
0007C6 47F0 D712           007D2    879          B     BIN2
0007CA 4185 0020           00020    880 BIN1     LA    A,32(P)           A <--- P+8
0007CE 5080 B038           024F0    881          ST    A,AA
0007D2 5818 C000           00000    882 BIN2     L     1,0(A)            -1- <--- X
0007D6 5910 B074           0252C    883          C     1,MARKER
0007DA 4780 D74E           0080E    884          BE    BIN3
                                    885 *
0007DE 8810 0018           00018    886          SRL   1,24              -1- <--- NUMBER( ST(A) )
0007E2 5410 D790           00850    887          N     1,=F'15'
                                    888 *
0007E6 1211                         889          LTR   1,1
0007E8 4740 DD80           00E40    890          BL    MON10
                                    891 *
0007EC 5910 D794           00854    892          C     1,=F'9'
0007F0 4720 DD80           00E40    893          BH    MON10                                                   00007252
                                    894 *
0007F4 4130 000A           0000A    895          LA    3,10              W <--- 10*W + X
0007F8 1C29                         896          MR    2,W
0007FA 1A31                         897          AR    3,1               -3- <--- 10*W + X
0007FC 1893                         898          LR    W,3                W = 10 * W + X
0007FE 5090 B044           024FC    899          ST    W,WW
000802 4188 0004           00004    900          LA    A,4(A)            A <--- A + 1
0008C6 5080 B038           024F0    901          ST    A,AA
0008CA 47F0 D712           007D2    902          B     BIN2
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0008CE 4125 001C            0001C   904 BIN3     LA~    2,28(P)         -2-  <---  LOC( ST(P+7) )
000812 9560 2000      00000         905          CLI    0(2),C'-'       MINUS SIGN
000816 4770 D75C            0081C   906          BNE    BIN4             ... NO
00081A 1199                         907          LNR    W,W              ... YES
00081C 509A 0000            00000   908 BIN4     ST     W,0(S)          ST(S)  <---  W
000820 5090 BC44            024FC   909          ST     W,WW
000824 5090 D78C            0084C   910          ST     W,SVWBIN
                                    911 *
000828 41AA 0004            00004   912          LA     S,4(S)          S  <---  S + 1
00082C 50A0 B050            02508   913          ST     S,SS
000830 59AC D798            00858   914          C      S,=A(ST+4*LNTHST)                              00007432
000834 4720 D77C            0083C   915          BH     STOOBIG5                                       00007433
                                    916 *
000838 47F0 D1F2            002B2   917          B      ENDFN
                                    918 *                                                             00007452
                                    919 STOOBIG5 ABEND  75,DUMP          S TOO BIG IN BIN          00007453
                                    926 *
00084A C00C
00084C 00000000                     927 SVWBIN   DC     F'0'            SVWBIN
                                    928 *
000850                              929          LTORG
000850 C000C00F                     930                 =F'15'
000854 00000009                     931                 =F'9'
000858 C0006D98                     932                 =A(ST+4*LNTHST)
00085C 1C                           933                 =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    935 ****    THE PRIMITIVE MACRO "DEC" CONVERTS A SIGNED BINARY INTEGER
                                    936 ****     INTO A DECIMAL DIGIT STRING OF CHARACTERS, PRECEDED BY A
                                    937 ****      MINUS IF NECESSARY.
                                    938 *
00085E                              939          DS    0H
00085E                              940 DEC      EQU   *               DEC
00085E 900F B3F8              02880 941          STM   0,15,SAVEDEC
                                    942 *
000862 FA30 B8A0 D844 02D58 00904   943          AP    KTDEC,=P'1'     KTDEC <--- KTDEC + 1                00007542
                                    944 *                                                                    00007543
000868 5895 001C              0001C 945          L     W,28(P)         W  <---  ST(P+7)
00086C 5090 B044              024FC 946          ST    W,WW
000870 5090 D830              008F0 947          ST    W,WWDEC
000874 1299                         948          LTR   W,W
000876 47BC D7D2              00892 949          BNL   DEC1
00087A 1099                         950          LPR   W,W             W <--- ABS ( W )
00087C 5090 B044              024FC 951          ST    W,WW
000880 D203 B038 D838 024F0 008F8   952          MVC   AA,=C'-    '    A <--- '-'
000886 5880 B038              024F0 953          L     A,AA
00088A 5080 D828              008E8 954          ST    A,SVADEC1
00088E 45E0 010E              001CE 955          BAL   14,LOAD
                                    956 *
000892 4110 0001              00001 957 DEC1     LA    1,1             Z <--- 1
000896 4100 000A              0000A 958          LA    0,10            -0- <--- 10
00089A 1831                         959 DEC2     LR    3,1             -3- <--- 10*Z
00089C 1C20                         960          MR    2,0
00089E 1939                         961          CR    3,W
0008A0 4720 D7EA              008AA 962          BH    DEC3
0008A4 1813                         963          LR    1,3             Z <--- 10*Z
0008A6 47FC D7DA              0089A 964          B     DEC2
                                    965 *
0008AA 1829                         966 DEC3     LR    2,W             -2- <--- W
0008AC 8E20 0020              00020 967          SRDA  2,32            EXTEND SIGN
0008B0 1D21                         968          DR    2,1
0008B2 1892                         969          LR    W,2             -W- <--- REM( W / Z )
0008B4 5090 B044              024FC 970          ST    W,WW
                                    971 *
0008B8 D203 D82C D83C 008EC 008FC   972          MVC   AADEC,=CL4'  '  -A- <--- CHAR( QUOT( W / Z ) )
0008BE 4230 D82C              008EC 973          STC   3,AADEC
0008C2 96F0 D82C        008EC       974          OI    AADEC,X'F0'
0008C6 5880 D82C              008EC 975          L     A,AADEC
0008CA 5080 B038              024F0 976          ST    A,AA
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

0008CE 1821                       978          LR    2,1          Z <--- Z/10
0008D0 8E20 0020            00020 979          SRDA  2,32         EXTEND SIGN
0008D4 1D20                       980          DR    2,0
0008D6 1813                       981          LR    1,3
0008D8 45E0 D10E            00ICE 982          BAL   14,LOAD
                                  983 *
0008DC 5910 D840            00900 984          C     1,=F'1'      Z .LT. 1
0008E0 4780 D7EA            008AA 985          BNL   DEC3          ... NO
                                  986 *
0008E4 47F0 D1F2            002B2 987          B     ENDFN          ... YES
                                  988 *
0008E8 C0000000                   989 SVADEC1  DC    F'0'         SVADEC1
0008EC C000C000                   990 AADEC    DC    F'0'         AADEC
0008F0 0000C000                   991 WWDEC    DC    F'0'         WWDEC
                                  992 *
0008F8                            993          LTORG
0008F8 60404040                   994                =C'-   '
0008FC 40404040                   995                =CL4' '
0009C0 0000C001                   996                =F'1'
0009C4 1C                         997                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0009C6                             999          DS    0H
0009C6                            1000 BAR      EQU   *               BAR
0009C6 900F B438                  028F0 1001    STM   0,15,SAVEBAR
                                        1002 *
00090A FA30 B8B4 D8C0 02D6C 00980 1003          AP    KTBAR,=P'1'     KTBAR <--- KTBAR + 1                00008082
                                        1004 *                                                              00008083
000910 5895 0024                  00024 1005    L     W,36(P)         W  <---  ST(P+9)
000914 5090 B044                  024FC 1C06    ST    W,WW
000918 5885 002C                  0002C 1007    L     A,44(P)         A  <---  ST(P+11)
00091C 5080 B038                  024F0 1008    ST    A,AA
000920 4115 001C                  0001C 1009    LA    1,28(P)         -1-  <---  LOC( ST(P+7) )
000924 954E 1000           00000        1010    CLI   0(1),C'+'
000928 4770 D872                  00932 1011    BNE   BAR1
00092C 1A89                             1012    AR    A,W             A  <---  W+A
00092E 47F0 D8AC                  0096C 1013    B     BAR5
000932 9560 1000           00000        1014 BAR1     CLI   0(1),C'-'
000936 4770 D882                  00942 1015    BNE   BAR2
00093A 1388                             1016    LCR   A,A             A  <---  W-A
00093C 1A89                             1017    AR    A,W
00093E 47F0 D8AC                  0096C 1018    B     BAR5
000942 955C 1000           00000        1019 BAR2     CLI   0(1),C'*'
000946 4770 D894                  00954 1020    BNE   BAR3
00094A 1839                             1021    LR    3,W             A  <---  W*A
00094C 1C28                             1022    MR    2,A
00094E 1883                             1023    LR    A,3
000950 47F0 D8AC                  0096C 1024    B     BAR5
000954 1829                             1025 BAR3     LR    2,W             -3-  <--- QUOT( W/A )
000956 8E20 0020                  00020 1026    SRDA  2,32            -2-  <---  REM( W/A )
00095A 1D28                             1027    DR    2,A
00095C 9561 1000           00000        1028    CLI   0(1),C'/'        ST(P+7) = '/'
000960 4770 D8AA                  0096A 1029    BNE   BAR4
000964 1883                             1030    LR    A,3               A = QUOTIENT OF W/A
000966 47F0 D8AC                  0096C 1031    B     BAR5
00096A 1882                             1032 BAR4     LR    A,2               A = REMAINDER OF W/A
00096C 5080 B038                  024F0 1033 BAR5     ST    A,AA
000970 5080 D8BC                  0097C 1034    ST    A,AABAR
000974 45E0 D10E                  001CE 1035    BAL   14,LOAD
000978 47F0 D1F2                  002B2 1036    B     ENDFN
                                        1037 *
00097C 0000C000                         1038 AABAR    DC    F'0'            AABAR
                                        1039 *
000980                                  1040    LTORG
000980 1C                               1041          =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                    00008492
000982                             1043          DS    0H                                                     00008493
000982                             1044 EOB      EQU   *               EOB
000982 900F B478             02930 1045          STM   0,15,SAVEEOB
                                   1046 *
000986 FA30 B8C8 D8E8 02D80 009A8  1047          AP    KTEOB,=P'1'     KTEOB <--- KTEOB + 1
                                   1048 *
00098C 4110 D8E0             009A0 1049          LA    1,=A(WREOB)
000990 58F0 D8E4             009A4 1050          L     15,=A(PRNTMES)
000994 05EF                        1051          BALR  14,15
                                   1052 *
000996 47F0 D1F2             002B2 1053          B     ENDFN
                                   1054 *
00099C                             1055          DS    0F                                                     00008552
00099C 0000                        1056 WRECB    DC    H'0'            WREOB
00099E 5CD5                        1057          DC    CL2'*N'
                                   1058 *
0009A0                             1059          LTORG
0009A0 0000099C                    1060                =A(WREOB)
0009A4 00001D90                    1061                =A(PRNTMES)
0009A8 1C                          1062                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0009AA                             1064          DS   0H
0009AA                             1065 SIGN     EQU  *                 SIGN
0009AA 900F B4B8             02970 1066          STM  0,15,SAVESIGN
                                   1067 *
0009AE FA30 B8DC D958 02D94 00A18  1068          AP   KTSIGN,=P'1'      KTSIGN <--- KTSIGN + 1
                                   1069 *
0009B4 5895 001C             0001C 1070          L    W,28(P)           W <--- ST(P+7)
0009B8 5090 B044             024FC 1071          ST   W,WW
0009BC 5090 D944             00A04 1072          ST   W,WWSIGN
                                   1073 *
0009C0 1299                        1074          LTR  W,W               TEST -W-
0009C2 4770 D90E             009CE 1075          BNZ  WNOTO
0009C6 5890 D948             00A08 1076          L    W,=F'0'
0009CA 47F0 D91E             009DE 1077          B    STOREW
0009CE 472C D91A             009DA 1078 WNOTO    BP   WPLUS
0009D2 5890 D94C             00A0C 1079          L    W,=F'-1'
0009D6 47F0 D91E             009DE 1080          B    STOREW
0009DA 5890 D950             00A10 1081 WPLUS    L    W,=F'1'
0009DE 509A 0000             00000 1082 STOREW   ST   W,0(S)            ST(S) <--- W
0009E2 41AA 0004             00004 1083          LA   S,4(S)            S <--- S + 1
0009E6 50A0 B050             02508 1084          ST   S,SS
0009EA 59A0 D954             00A14 1085          C    S,=A(ST+LNTHST*4)
0009EE 47D0 D1F2             002B2 1086          BNH  ENDFN
                                   1087 *
                                   1088 STOCBIG6 ABEND 76,DUMP          S TOO BIG IN SIGN
                                   1095 *
000AC2 0000
000AC4 00000000                    1096 WWSIGN   DC   F'0'               WWSIGN
                                   1097 *
000AC8                             1098          LTORG
000AC8 000CC000                    1099                =F'0'
000ACC FFFFFFFF                    1100                =F'-1'
000A10 0000C001                    1101                =F'1'
000A14 00006D98                    1102                =A(ST+LNTHST*4)
000A18 1C                          1103                =P'1'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1105 ****     ITEM   THIS ROUTINE OUTPUTS THE ITEM ON THE STACK STARTING AT
                                    1106 ****              ST(X).  IF THE ITEM IS NOT COMPLETE, PRINTING STOPS
                                    1107 ****                AT  ST(S-1)  AND IS FOLLOWED BY "...  (INCOMPLETE)".
                                    1108 *
000A1A                              1109           DS     0H
000A1A                              1110 ITEM      EQU    *                 ITEM
000A1A 900F B4F8            029B0   1111           STM    0,15,SVITEM       SAVE REGISTERS
                                    1112 *
000A1E 5810 1000            00000   1113           L      1,0(,1)           LOAD ADDRESS OF -X- INTO -1-
000A22 5010 DA14            00AD4   1114           ST     1,SV1A
                                    1115 *
000A26 5810 1000            00000   1116           L      1,0(,1)            LOAD -X- INTO -1-
000A2A 5010 DA18            00AD8   1117           ST     1,SV1B
                                    1118 *
000A2E 5080 DA20            00AE0   1119           ST     A,AAA             SAVE -A- AND -H-
000A32 5040 DA24            00AE4   1120           ST     H,HHH
                                    1121 *
000A36 4140 0000            00000   1122           LA     H,0               SET -H- TO ZERO
000A3A 5040 B008            024C0   1123           ST     H,HH
                                    1124 *
000A3E 4130 0001            00001   1125           LA     3,1               SET -K- TO -1-    ( -3- IS -K- )
000A42 1823                         1126 LPITM     LR     2,3               SET -2- TO 4*K
000A44 8920 0002            00002   1127           SLL    2,2
000A48 1A21                         1128           AR     2,1               SET -2- TO  X + K
000A4A 5020 DA28            00AE8   1129           ST     2,XKSTAR
000A4E 5882 0000            00000   1130           L      A,0(2)            A <--- ST(X+K)
000A52 5080 DA1C            00ADC   1131           ST     A,AAITEM
000A56 5080 B038            024F0   1132           ST     A,AA
000A5A 45EC D10E            C01CE   1133           BAL    14,LOAD           LOAD
                                    1134 *
000A5E 5821 0000            00000   1135           L      2,0(1)            -2- <--- ST(X)
000A62 1222                         1136           LTR    2,2               DOES  ST(X) = 0
000A64 4780 D9BC            00A7C   1137           BZ     TESTK              .. YES
000A68 8820 0002            00002   1138           SRL    2,2                .. NO,  DOES  K = ST(X)-1
000A6C 0620                         1139           BCTR   2,0
                                    1140 *
000A6E 1932                         1141           CR     3,2
000A70 47B0 D9D0            00A90   1142           BNL    EXITM              ..... YES
                                    1143 *
000A74 4130 3001            00001   1144           LA     3,1(,3)            ..... NO,  K <--- K + 1
000A78 47F0 D982            00A42   1145           B      LPITM
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

000A7C 182A                       1147 TESTK    LR    2,S          -2- <--- S-X-1
000A7E 1821                       1148          SR    2,1
000A80 0620                       1149          BCTR  2,0
                                  1150 *
000A82 1932                       1151          CR    3,2          DOES  K = S-X-1
000A84 47B0 D9D0          00A90   1152          BNL   EXITM         .. YES
                                  1153 *
000A88 4130 3001          00001   1154          LA    3,1(,3)      .. NO, K <--- K + 1
000A8C 47F0 D982          00A42   1155          B     LPITM
                                  1156 *
000A90 5821 0000          00000   1157 EXITM    L     2,0(1)       -2- <--- ST(X)
000A94 1222                       1158          LTR   2,2          DOES  ST(X) = 0
000A96 4770 D9FE          00A8E   1159          BNZ   SKIPITEM      ... NO
                                  1160 *
                                  1161          PRNT  EMITM         .. YES
                                  1172 *
000A8E 5880 DA20          00AE0   1173 SKIPITEM L     A,AAA        RESTORE -A- AND -H-
000AC2 5080 B038          024F0   1174          ST    A,AA
000AC6 5840 DA24          00AE4   1175          L     H,HHH
000ACA 5040 B008          024C0   1176          ST    H,HH
                                  1177 *
000ACE 980F B4F8          02980   1178          LM    0,15,SVITEM  RESTORE REGISTERS
                                  1179 *
000AD2 07FE                       1180          BR    14           RETURN
                                  1181 *
000AD4 C00CC000                   1182 SV1A     DC    F'0'         SV1A
000AD8 C000C000                   1183 SV1B     DC    F'0'         SV1B
000ADC 000CC000                   1184 AAITEM   DC    F'0'         AAITEM
000AE0 C00CC000                   1185 AAA      DC    F'0'         AAA
000AE4 C000C000                   1186 HHH      DC    F'0'         HHH
000AE8 000CC000                   1187 XKSTAR   DC    F'0'         XKSTAR
                                  1188 *
000AEC 0014                       1189 EMITM    DC    H'20'
000AEE C000                       1190          DC    H'0'
000AF0 4B4B4B4040404040          1191          DC    C'...     (INCOMPLETE)'
                                  1192 *
000BC8                            1193          LTORG
                                  1194 *
                                  1195          PRINT GEN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    1197 ****    MONITOR  1       UNMATCHED ";" IN DEFINITION STRING.  TREATED
                                    1198 ****                     AS <;>.
                                    1199 *
000BC8                              1200         DS    0H
000BC8                              1201 MON1    EQU   *               MON1
000BC8 900F B538            029F0   1202         STM   0,15,SAVEMON1
                                    1203 *
                                    1204         PRNT  EM1A
000BCC                              1205+        CNOP  0,4
000BCC 9001 DA58            00B18   1206+        STM   0,1,*+12
000BD0 4510 DA64            00B24   1207+        BAL   1,*+20
000BD4 00001040                     1208+        DC    A(EM1A)
000BD8 00000000                     1209+        DC    A(0)
000BDC 000CC000                     1210+        DC    A(0)
000BE0 00001D90                     1211+        DC    A(PRNTMES)
000BE4 58F0 DA60            00B20   1212+        L     15,*-4
000BE8 05EF                         1213+        BALR  14,15
000BEA 9801 DA58            00B18   1214+        LM    0,1,*-18
                                    1215 *
000BEE 4115 0008            00008   1216         LA    1,8(P)          -1-  <---  P+2
                                    1217 *
                                    1218         BALITEM
000BF2 0700                         1219+        CNOP  0,4
000BF4 5010 DA80            00B40   1220+        ST    1,*+12
000BF8 4510 DA84            00B44   1221+        BAL   1,*+12
000BFC 00000B40                     1222+        DC    A(*+4)
000C00 0000C000                     1223+        DC    F'0'
000C04 45E0 D95A            00A1A   1224+        BAL   14,ITEM
                                    1225 *
                                    1226         PRINT NOGEN                                          00009490
                                    1227 *
                                    1228         PRNT  EM1B
                                    1239 *
000B6A 47F0 D0A2            00162   1240         B     COPY
                                    1241 *
                                    1242         PRINT NOGEN                                          00009492
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    1244 ****     MONITOR  2        UNQUOTED  ¬  IN ARGUMENT LIST IN INPUT
                                    1245 ****                       STREAM.  TREATED AS  <¬>.
                                    1246 *
000B6E                              1247          DS     0H
000B6E                              1248 MON2     EQU    *                MON2
000B6E 900F B578            02A30   1249          STM    0,15,SAVEMON2
                                    1250 *
                                    1251          PRNT   EM2A
000B96 4116 0008            00008   1262          LA     1,8(F)           -1-  <---  F+2
                                    1263          BALITEM
                                    1270          PRNT   EM2B
000BC2 47F0 D0A2            00162   1281          B      COPY                                          00009620
                                    1282 *
                                    1283 *
                                    1284 *
                                    1285 ****     MONITOR  3        IMPOSSIBLE CHARACTER (NEGATIVE) AS ARGUMENT
                                    1286 ****                       NUMBER.  TERMINATE.
                                    1287 *
000BC6                              1288          DS     0H
000BC6                              1289 MON3     EQU    *                MON3
000BC6 900F B5B8            02A70   1290          STM    0,15,SAVEMON3
                                    1291 *
                                    1292          PRNT   EM3A
000BFE 4115 0008            00008   1303          LA     1,8(P)           -1-  <---  P+2
                                    1304          BALITEM
000C18 47F0 DDAC            00E6C   1311          B      MON11
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1313 ****     MONITOR  4        NOT ENOUGH ARGUMENTS SUPPLIED IN CALL.
                                    1314 ****                       TERMINATE.
                                    1315 *
000C1C                              1316          DS     0H
000C1C                              1317 MON4     EQU    *                MON4
000C1C 900F B5F8            02AB0   1318          STM    0,15,SAVEMON4
                                    1319 *
                                    1320          PRNT   EM4A
000C42 1B44                         1331          SR     H,H              H  <---  0
000C44 5040 B008            024C0   1332          ST     H,HH
000C48 5080 0BD0            00C90   1333          ST     A,AALOADX
000C4C 45E0 D10E            001CE   1334          BAL    14,LOAD
                                    1335          PRNT   EM4B
000C72 4115 0008            00008   1346          LA     1,8(P)           -1-  <---  P+2
                                    1347          BALITEM
000C8C 47F0 DDAC            00E6C   1354          B      MON11
                                    1355 *
000C90 C0000000                     1356 AALOADX  DC     F'0'             AALOADX
                                    1357 *
                                    1358 *
                                    1359 *
                                    1360 ****     MONITOR  5        TERMINATOR ?  IN IMPOSSIBLE PLACE; IF  C = 0,
                                    1361 ****                       THIS IS THE INPUT STREAM.  PROBABLY MACHINE
                                    1362 ****                       ERROR: TERMINATE.  IF  C ¬= 0,  THIS IS AN
                                    1363 ****                       ARGUMENT LIST.  PROBABLY DUE TO A MISSING
                                    1364 ****                       SEMICOLON: FINAL SEMICOLON INSERTED.
                                    1365 *
000C94                              1366          DS     0H
000C94                              1367 MON5     EQU    *                MON5
000C94 900F B638            02AF0   1368          STM    0,15,SAVEMON5
                                    1369 *
                                    1370          PRNT   EM5A
000CBA 1277                         1381          LTR    C,C              C = 0
000CBC 4770 DC28            00CE8   1382          BNE    M51               ... NO
                                    1383          PRNT   EM5B
000CE2 47F0 DDAC            00E6C   1394          B      MON11
                                    1395 M51      PRNT   EM5C
000DCA 4116 0008            00008   1406          LA     1,8(F)           -1-  <---  F+2
                                    1407          BALITEM
                                    1414          PRNT   EM5D
000D46 4115 0008            00008   1425          LA     1,8(P)           -1-  <---  P+2
                                    1426          BALITEM
                                    1433          PRNT   EM5E
000D82 5B70 DDA8            00E68   1444          S      C,=F'4'          C <--- C - 1
000D86 5070 B02C            024E4   1445          ST     C,CC
000D8A 47F0 D512            005D2   1446          B      APPLY
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1448 ****     MONITOR  7       UNDEFINED MACRO NAME: TERMINATE.
                                    1449 *
000D8E                              1450           DS     0H
000D8E                              1451 MON7      EQU    *                MON7
000D8E 900F B678              02B30 1452           STM    0,15,SAVEMON7
                                    1453 *
                                    1454           PRNT   EM7A
000DB6 1819                         1465           LR     1,W
                                    1466           BALITEM
000DCC 47F0 DDAC              00E6C 1473           B      MON11
                                    1474 *
                                    1475 *
                                    1476 *
                                    1477 ****     MONITOR  8       WRONG EXIT  ( NOT C = H = 0 ).  MACHINE
                                    1478 ****                       ERROR: TERMINATE.
                                    1479 *
000DC0 900F B6B8              02B70 1480 MON8      STM    0,15,SVMON8
                                    1481 *
                                    1482           PRNT   EM8A             MON8
000DF6 47F0 DDAC              00E6C 1493           B      MON11
                                    1494 *
                                    1495 *
                                    1496 *
                                    1497 ****     MONITOR  9       UPDATE STRING TOO LONG: TERMINATE.
                                    1498 *
000DFA                              1499           DS     0H
000DFA                              1500 MON9      EQU    *                MON9
000DFA 900F B6F8              02B80 1501           STM    0,15,SAVEMON9
                                    1502 *
                                    1503           PRNT   EM9A
000E22 4115 0024              00024 1514           LA     1,36(P)          -1-  <---  P+9
                                    1515           BALITEM
000E3C 47F0 DDAC              00E6C 1522           B      MON11
                                    1523 *
                                    1524 *
                                    1525 *
                                    1526 ****     MONITOR 10       NON-DIGIT IN ARGUMENT FOR "BIN".  TERMINATE.
                                    1527 *
                                    1528 ****     MONITOR 11       GENERAL MONITOR AFTER IRREMEDIABLE ERRORS.
                                    1529 *
                                    1530 MON10     PRNT   EM10A            MON10
000E62 47F0 DDAC              00E6C 1541           B      MON11
                                    1542 *
000E68                              1543           LTORG
000E68 C00CC004                     1544                  =F'4'
```

```
  LOC   OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000E6C 90CF B738              02BF0  1546 MON11    STM   0,15,SVMON11   MON11
                              1547 *
000E70 4190 0014              00014  1548          LA    W,20           W  <---  20
000E74 5090 B044              024FC  1549          ST    W,WW
                              1550 *
                              1551          PRNT  EM11A
                              1562 *
000E9A 1255                   1563 TESTPF   LTR   P,P            TEST -P- AND -F-
000E9C 4770 0DE6              00EA6  1564          BNE   PTOF
000EA0 1266                   1565          LTR   F,F
000EA2 4780 DF2C              00FEC  1566          BE    ENDMON11
                              1567 *
000EA6 1956                   1568 PTOF     CR    P,F
000EA8 4720 DE22              00EE2  1569          BH    ZP2
                              1570 *
000EAC 4136 C008              00008  1571          LA    3,8(F)         Z  <---  F+2   ( -3- IS -Z- )
000EB0 5030 DF68              01028  1572          ST    3,ZM11
000EB4 5866 C000              00000  1573          L     F,0(F)         F  <---  ST(F)
000EB8 5060 B020              024D8  1574          ST    F,FF
                              1575 *
                              1576          PRNT  EM11C
                              1587 *
000ECE 47F0 DE56              00F16  1588          B     SETR1
                              1589 *
000EE2 4135 0008              00008  1590 ZP2      LA    3,8(P)         Z  <---  P+2   ( -3- IS -Z- )
000EE6 5030 DF68              01028  1591          ST    3,ZM11
000EEA 5855 0000              00000  1592          L     P,0(P)         P  <---  ST(P)
000EEE 5050 B014              024CC  1593          ST    P,PP
                              1594 *
                              1595          PRNT  EM11B
                              1606 *
000F16 4120 0001              00001  1607 SETR1    LA    2,1            R <--- 1   (-2- IS -R-)
000F1A 4110 DF70              01030  1608 LOOPR    LA    1,=A(ZM11)
000F1E 45E0 D95A              00A1A  1609          BAL   14,ITEM
                              1610 *
000F22 5813 0000              00000  1611          L     1,0(3)         -1- <--- ST(Z)
000F26 1211                   1612          LTR   1,1            DOES  ST(Z) = 0
000F28 4780 DF20              00FE0  1613          BE    SETW1           ... YES
                              1614 *
000F2C 1A31                   1615          AR    3,1               ... NO,   Z  <---  Z + ST(Z)
000F2E 5030 DF68              01028  1616          ST    3,ZM11
                              1617 *
000F32 5800 3000              00000  1618          L     0,0(,3)        DOES   ST(Z) = MARKER
000F36 5900 B074              0252C  1619          C     0,MARKER
000F3A 4780 DF20              00FE0  1620          BE    SETW1           ... YES
                              1621 *
000F3E 5900 DF74              01034  1622          C     W,=F'1'         ... NO,   DOES  W = 1
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
000F42 4780 DF02             00FC2  1623          BE   SKIPM11D          ..... YES
                                    1624 *
                                    1625          PRNT EM11D             ..... NO
                                    1636 *
000F6A 4E20 DF18             00FD8  1637          CVD  2,REM11
000F6E D203 DF14 DF78 00FD4 01038  1638          MVC  EM11G+4(4),=X'40202120'
000F74 DE03 DF14 DF1E 00FD4 00FDE  1639          ED   EM11G+4(4),REM11+6
                                    1640 *
                                    1641          PRNT EM11G
                                    1652 *
                                    1653          PRNT EM11E
                                    1664 *
000FC2 1929                         1665 SKIPM11D CR   2,W               IS  R .LT. W
000FC4 4780 DF20             00FE0  1666          BNL  SETW1             ... NO
000FC8 4120 2001             00001  1667          LA   2,1(,2)           ... YES,    R <--- R + 1
000FCC 47F0 DE5A             00F1A  1668          B    LOOPR
                                    1669 *
000FD0 0004                         1670 EM11G    DC   H'4'              EM11G
000FD2 0000                         1671          DC   H'0'
000FD4 40404040                     1672          DC   CL4' '
000FD8                              1673 REM11    DS   D                 REM11
                                    1674 *
                                    1675 *
000FE0 4190 0001             00001  1676 SETW1    LA   W,1               W <--- 1
000FE4 5090 B044             024FC  1677          ST   W,WW
000FE8 47F0 DDDA             00E9A  1678          B    TESTPF
                                    1679 *
                                    1680 ENDMON11 PRNT EM11F
                                    1691 *
00100E D203 B038 DF7C 024F0 0103C  1692          MVC  AA,=CL4'Q'        SET -A- TO 'Q'
001014 5880 B038             024F0  1693          L    A,AA
001018 45E0 D10E             001CE  1694          BAL  14,LOAD
                                    1695 *
00101C 1956                         1696          CR   P,F               P .GT. F
00101E 4720 D1F2             002B2  1697          BH   ENDFN              .. YES
001022 47F0 D048             00108  1698          B    START              .. NO
                                    1699 *
001026 0000
001028 00CC0000                     1700 ZM11     DC   F'0'              ZM11
                                    1701 *
001030                              1702          LTORG
001030 00001028                     1703                =A(ZM11)
001034 00CC0001                     1704                =F'1'
001038 40202120                     1705                =X'40202120'
00103C D8404040                     1706                =CL4'Q'
                                    1707 *
                                    1708          PRINT GEN
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

001040                           1710 MESSAGES CSECT
                                 1711 *
                                 1712 ****  -ERROR- MESSAGES
                                 1713 *
001040 002E                      1714 EM1A     DC    H'46'
001042 5CD5                      1715          DC    CL2'*N'
001044 D4D6D5C9E3D6D97A          1716          DC    C'MONITOR: UNMATCHED SEMICOLON IN DEFINITION OF '
001072 002C                      1717 EM1B     DC    H'44'
001074 5CD5                      1718          DC    CL2'*N'
001076 C9C640E3C8C9E240          1719          DC    C'IF THIS HAD BEEN QUOTED THE RESULT WOULD BE '
0010A2 0000                      1720 EM1C     DC    H'0'
0010A4 5CD5                      1721          DC    CL2'*N'
0010A6 002C                      1722 EM2A     DC    H'44'
0010A8 5CD5                      1723          DC    CL2'*N'
0010AA D4D6D5C9E3D6D97A          1724          DC    C'MONITOR: UNQUOTED TILDE IN ARGUMENT LIST OF '
0010D6 002C                      1725 EM2B     DC    H'44'
0010D8 5CD5                      1726          DC    CL2'*N'
0010DA C9C640E3C8C9E240          1727          DC    C'IF THIS HAD BEEN QUOTED THE RESULT WOULD BE '
0011C6 0000                      1728 EM2C     DC    H'0'
0011C8 5CD5                      1729          DC    CL2'*N'
0011CA 0035                      1730 EM3A     DC    H'53'
0011CC 5CD5                      1731          DC    CL2'*N'
0011CE D4D6D5C9E3D6D97A          1732          DC    C'MONITOR: IMPOSSIBLE ARGUMENT NUMBER IN DEFINITION OF '
001143 00
001144 0015                      1733 EM4A     DC    H'21'
001146 5CD5                      1734          DC    CL2'*N'
001148 D4D6D5C9E3D6D97A          1735          DC    C'MONITOR: NO ARGUMENT '
00115D 00
00115E 000C                      1736 EM4B     DC    H'12'
001160 0000                      1737          DC    H'0'
001162 C9D540C3C1D3D340          1738          DC    C'IN CALL FOR '
00116E 0017                      1739 EM5A     DC    H'23'
001170 5CD5                      1740          DC    CL2'*N'
001172 D4D6D5C9E3D6D97A          1741          DC    C'MONITOR: TERMINATOR IN '
001189 00
00118A 0026                      1742 EM5B     DC    H'38'
00118C 0000                      1743          DC    H'0'
00118E C9D5D7E4E340E2E3          1744          DC    C'INPUT STREAM.  PROBABLY MACHINE ERROR.'
0011B4 0012                      1745 EM5C     DC    H'18'
0011B6 000C                      1746          DC    H'0'
0011B8 C1D9C7E4D4C5D5E3          1747          DC    C'ARGUMENT LIST FOR '
0011CA 003B                      1748 EM5D     DC    H'59'
0011CC 5CD5                      1749          DC    CL2'*N'
0011CE D7D9D6C2C1C2D3E8          1750          DC    C'PROBABLY DUE TO A SEMICOLON MISSING FROM '
0011F7 E3C8C540C4C5C6C9          1751          DC    C'THE DEFINITION OF '
0012C9 00
0012CA 002C                      1752 EM5E     DC    H'44'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
00120C 5CD5                       1753          DC    CL2'*N'
00120E C9C640C140C6C9D5           1754          DC    C'IF A FINAL SEMICOLON IS ADDED THE RESULT IS '
00123A C000                       1755 EM5F     DC    H'0'
00123C 5CD5                       1756          DC    CL2'*N'
00123E 0018                       1757 EM7A     DC    H'24'
001240 5CD5                       1758          DC    CL2'*N'
001242 D4D6D5C9E3D6D97A           1759          DC    C'MONITOR: UNDEFINED NAME '
00125A 002D                       1760 EM8A     DC    H'45'
00125C 5CD5                       1761          DC    CL2'*N'
00125E D4D6D5C9E3D6D97A           1762          DC    C'MONITOR: UNMATCHED >.  PROBABLY MACHINE ERROR.'
00128C 0026                       1763 EM9A     DC    H'38'
00128E 5CD5                       1764          DC    CL2'*N'
001290 D4D6D5C9E3D6D97A           1765          DC    C'MONITOR: UPDATE ARGUMENT TOO LONG FOR '
0012B6 001D                       1766 EM10A    DC    H'29'
0012B8 5CD5                       1767          DC    CL2'*N'
0012BA D4D6D5C9E3D6D97A           1768          DC    C'MONITOR: NON-DIGIT IN NUMBER '
0012D7 00
0012D8 0013                       1769 EM11A    DC    H'19'
0012DA 5CD5                       1770          DC    CL2'*N'
0012DC C3E4D9D9C5D5E340           1771          DC    C'CURRENT MACROS ARE '
0012EF 00
0012F0 0010                       1772 EM11B    DC    H'16'
0012F2 5CD5                       1773          DC    CL2'*N'
0012F4 C1D3D9C5C1C4E840           1774          DC    C'ALREADY ENTERED '
0013C4 0010                       1775 EM11C    DC    H'16'
0013C6 5CD5                       1776          DC    CL2'*N'
0C13C8 D5D6E340E8C5E340           1777          DC    C'NOT YET ENTERED '
001318 0004                       1778 EM11D    DC    H'4'
00131A 5CD5                       1779          DC    CL2'*N'
00131C C1D9C740                   1780          DC    C'ARG '
001320 0005                       1781 EM11E    DC    H'5'
001322 000C                       1782          DC    H'0'
0C1324 4040404040                 1783          DC    CL5' '
001329 00
00132A 0018                       1784 EM11F    DC    H'24'
00132C 5CD5                       1785          DC    CL2'*N'
00132E C5D5C440D6C640D4           1786          DC    C'END OF MONITOR PRINTING.'
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

001348                           1788 ABENDS   CSECT
                                 1789 *
001348 4040404040404040          1790          DC    CL64' '
001388 C1C2C5D5C44040F0          1791          DC    C'ABEND  013  IS THE  FINISH  ABEND'
0013A9 4040404040404040          1792          DC    CL64' '
0013E9 C1C2C5D5C44040F0          1793          DC    C'ABEND  014  IS THE  EOF  ABEND'
001407 4040404040404040          1794          DC    CL64' '
001447 C1C2C5D5C44040F0          1795          DC    C'ABEND  031  IS THE  PRDCB TOO MANY LINES  ABEND'
001476 4040404040404040          1796          DC    CL64' '
0014B6 C1C2C5D5C44040F0          1797          DC    C'ABEND  032  IS THE  PRDCB DID NOT OPEN  ABEND'
0014E3 4040404040404040          1798          DC    CL64' '
001523 C1C2C5D5C44040F0          1799          DC    C'ABEND  040  IS THE  EOF  ABEND'
001541 4040404040404040          1800          DC    CL64' '
001581 C1C2C5D5C44040F0          1801          DC    C'ABEND  041  IS THE  CRDCB DID NOT OPEN  ABEND'
0015AE 4040404040404040          1802          DC    CL64' '
0015EE C1C2C5D5C44040F0          1803          DC    C'ABEND  066  IS THE TIME LIMIT  ABEND'
001612 4040404040404040          1804          DC    CL64' '
001652 C1C2C5D5C44040F0          1805          DC    C'ABEND  071  IS THE  S TOO BIG IS LOAD  ABEND'
00167E 4040404040404040          1806          DC    CL64' '
0016BE C1C2C5D5C44040F0          1807          DC    C'ABEND  072  IS THE  S TOO BIG IN FN  ABEND'
0016E8 4040404040404040          1808          DC    CL64' '
001728 C1C2C5D5C44040F0          1809          DC    C'ABEND  073  IS THE  S TOO BIG IN NEXTITEM  ABEND'
001758 4040404040404040          1810          DC    CL64' '
001798 C1C2C5D5C44040F0          1811          DC    C'ABEND  074  IS THE  S TOO BIG IN APPLY  ABEND'
0017C5 4040404040404040          1812          DC    CL64' '
001805 C1C2C5D5C44040F0          1813          DC    C'ABEND  075  IS THE  S TOO BIG IN BIN  ABEND'
001830 4040404040404040          1814          DC    CL64' '
001870 C1C2C5D5C44040F0          1815          DC    C'ABEND  076  IS THE  S TOO BIG IN SIGN  ABEND'
00189C 4040404040404040          1816          DC    CL64' '
0018DC C1C2C5D5C44040F0          1817          DC    C'ABEND  091  IS THE  KT LESS THAN ZERO  ABEND'
0019C8 4040404040404040          1818          DC    CL64' '
001948 C1C2C5D5C44040F0          1819          DC    C'ABEND  092  IS THE  KT GREATER THAN 80  ABEND'
001975 4040404040404040          1820          DC    CL64' '
0019B5 C1C2C5D5C44040F0          1821          DC    C'ABEND  094  IS THE  PC NOT ZERO OR ASTERISK N  ABEND'
0019E9 4040404040404040          1822          DC    CL64' '
001A29 C1C2C5D5C44040F1          1823          DC    C'ABEND  131  IS THE  ECDCB TOO MANY LINES  ABEND'
001A58 4040404040404040          1824          DC    CL64' '
001A98 C1C2C5D5C44040F1          1825          DC    C'ABEND  132  IS THE  ECDCB DID NOT OPEN  ABEND'
001AC5 4040404040404040          1826          DC    CL64' '
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

001BC8                            1828 INOUT    CSECT
                                  1829 *
                                  1830 *
                                  1831 ****  RDSM - READ SYMBOL SUBROUTINE
                                  1832 *
                                  1833 *
                                  1834 RDSM     SAVE  (14,12),,*     SAVE REGISTERS
001B08 47F0 F00A            0000A 1835+RDSM     B     10(0,15) BRANCH AROUND ID
001BCC 04                         1836+         DC    AL1(4)
001BCD D9C4E2D4                   1837+         DC    CL4'RDSM' IDENTIFIER
001B11 00
001B12 90EC D00C            0000C 1838+         STM   14,12,12(13) SAVE REGISTERS
                                  1839 *
001B16 05C0                       1840          BALR  12,0           ESTABLISH  REG 12  AS THE BASE REGISTER
001B18                            1841          USING *,12
                                  1842 *
001B18 4180 C16C            01C84 1843          LA    11,SVRDSM      GET ADDRESS OF THE SAVE AREA
001B1C 50BC D008            00008 1844          ST    11,8(,13)      STORE FORWARD POINTER
001B20 50DC B004            00004 1845          ST    13,4(,11)      STORE BACKWARD POINTER
001B24 18DB                       1846          LR    13,11          SET UP  REG 13  FOR SAVE AREA
                                  1847 *
001B26 5820 1000            00000 1848          L     2,0(,1)         -2- POINTS AT THE -WORD- IN THE -STACK-
001B2A 9012 C380            01E98 1849          STM   1,2,SV12RDSM   SAVE -1- AND -2-
                                  1850 *
001B2E 5830 C5C4            020DC 1851 ADDPTIN  L     3,PTIN          LOAD -PTIN-, THE -IN- POINTER, INTO -3-
001B32 4130 3001            00001 1852          LA    3,1(,3)        PTIN  <---  PTIN + 1
001B36 5030 C5C4            020DC 1853          ST    3,PTIN
                                  1854 *
001B3A 5930 C1C8            01CE0 1855          C     3,=A(IN+80)     END OF CARD
001B3E 4720 C050            01B68 1856          BH    RDCD            ..YES,   GO READ A CARD
                                  1857 *
001B42 9526 3000      00000       1858          CLI   0(3),X'26'      EOB  ( 0-9-6  CARD PUNCH )
001B46 4780 C050            01B68 1859          BE    RDCD            ... YES,  GO READ A CARD
                                  1860 *
001B4A D203 2000 C1CC 00000 01CE4 1861          MVC   0(4,2),=CL4' '   ..NO, MOVE IN THE CHARACTER
001B50 D200 2000 30C0 00000 00000 1862          MVC   0(1,2),0(3)
                                  1863 *
001B56 5800 D004            00004 1864          L     13,4(,13)       RESTORE REG -13-
                                  1865 *
                                  1866          RETURN (14,12),T,RC=0   RESTORE REGISTERS AND RETURN
001B5A 98EC D00C            0000C 1867+         LM    14,12,12(13) RESTORE THE REGISTERS
001B5E 92FF D00C      0000C       1868+         MVI   12(13),X'FF' SET RETURN INDICATION
001B62 41F0 0000            00000 1869+         LA    15,0(0,0) LOAD RETURN CODE
001B66 07FE                       1870+         BR    14 RETURN
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   1872 ****  RDCD - READ CARD SUBROUTINE
                                   1873 *
                                   1874 *
001B68 4110 C1D0           01CE8   1875 RDCD     LA     1,=A(IN1)         READ CARD
001B6C 58F0 C1D4           01CEC   1876          L      15,=V(READCARD)
001B70 05EF                        1877          BALR   14,15
001B72 50F0 C1B4           01CCC   1878          ST     15,RCRD           SAVE RETURN CODE
                                   1879 *
001B76 12FF                        1880          LTR    15,15             END OF DATA SET
001B78 4770 C1B8           01CD0   1881          BNZ EOF                   .. YES, GO TO -EOF-
                                   1882 *
001B7C FA20 C58D C1F0 020D5 01D08  1883          AP     KREAD,=P'1'       .. NO, KREAD <---KREAD +1
                                   1884 *
001B82 4700 C076           01B8E   1885 NOPRDSM  NOP    LNRD1             FIRST R
                                   1886 *
001B86 96F0 C06B    01B83          1887          OI     NOPRDSM+1,X'F0'   RESET FIRST -R-
001B8A 47F0 C0B6           01BCE   1888          B      PRHDR
                                   1889 *
001B8E FA10 C5B7 C1F0 020CF 01D08  1890 LNRC1    AP     LNRD,=P'1'        LNRD <--- LNRD + 1
                                   1891 *
001B94 F911 C5B7 C1E8 020CF 01D00  1892          CP     LNRD,=P'40'       END OF PAGE
001B9A 47D0 C114           01C2C   1893          BNH    PRNTLNRD          ... NO
                                   1894 *
                                   1895 ****  PRINT PAGE FOOTING
                                   1896 *
001B9E D284 C61D C894 02135 023AC  1897          MVC    LINOUT,IDOTMARK   .. YES
001BA4 92F0 C61D    02135          1898          MVI    LINOUT,C'0'
                                   1899 *
                                   1900          PRNTI LINOUT
001BA8                             1901+         CNOP   0,4
001BA8 4510 C09C           01BB4   1902+         BAL    1,*+12
001BAC 00002135                    1903+         DC     A(LINOUT)
001BB0 000CC000                    1904+         DC     V(PRNTBUFI)
001BB4 58F0 C098           01BB0   1905+         L      15,*-4
001BB8 05EF                        1906+         BALR   14,15
                                   1907 *
                                   1908          PRINT NOGEN                                    00012951
                                   1909 *
                                   1910          PRNTI TTLELN
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    1918 ****  PRINT PAGE HEADING
                                    1919 *
001BCE D284 C61D C78B 02135 022A3   1920 PRHDR     MVC   LINOUT,HEADIN
001BD4 FA10 C5B9 C1F0 020D1 01D08   1921           AP    PAGEIN,=P'1'
001BDA DE03 C68F C5B9 021A7 020D1   1922           ED    LINOUT+114(4),PAGEIN
                                    1923 *
                                    1924           PRNTI LINOUT
                                    1931 *
001BF2 D284 C61D C919 02135 02431   1932           MVC   LINOUT,TTLELN
001BF8 92F0 C61D      02135         1933           MVI   LINOUT,C'0'
                                    1934 *
                                    1935           PRNTI LINOUT
                                    1942 *
                                    1943           PRNTI IDOTMARK
                                    1950 *
001C22 F810 C5B7 C1F1 020CF 01D09   1951           ZAP   LNRD,=P'0'       LNRD <--- 0
001C28 47F0 C076              01B8E 1952           B     LNRD1
                                    1953 *
                                    1954 *
                                    1955 ****  PRINT LINE IMAGE
                                    1956 *
001C2C D284 C61D C1F2 02135 01D0A   1957 PRNTLNRD MVC   LINOUT,=CL133' '
001C32 D205 C627 C1EA 0213F 01D02   1958           MVC   LINOUT+10(6),=X'402020202120'
001C38 DE05 C627 C5BD 0213F 020D5   1959           ED    LINOUT+10(6),KREAD
001C3E D24F C632 C5CD 0214A 020E5   1960           MVC   LINOUT+21(80),IN1
001C44 D205 C686 C1EA 0219E 01D02   1961           MVC   LINOUT+105(6),=X'402020202120'
001C4A DE05 C686 C5C0 0219E 020D8   1962           ED    LINOUT+105(6),KPRINT
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    1964 ****  REPLACE -EOB-  ( X'26' )  WITH  C'#'
                                    1965 *
001C50 9898 C1D8             01CF0  1966          LM    9,11,=A(LINOUT+21,1,LINOUT+100)
001C54 9526 9000       00000       1967 LPECB     CLI   0(9),X'26'      EOB
001C58 4770 C148             01C60  1968          BNE   *+8
001C5C 9278 9000       00000       1969          MVI   0(9),C'#'
001C60 879A C13C             01C54  1970          BXLE  9,10,LPEOB
                                    1971 *
                                    1972          PRNTI LINOUT
                                    1979 *
                                    1980 *
001C76 5830 C1E4             01CFC  1981          L     3,=A(IN)        PTIN  <---  LOC( IN )
001C7A 5030 C5C4             020DC  1982          ST    3,PTIN
                                    1983 *
001C7E 47F0 C016             01B2E  1984          B     ADDPTIN
                                    1985 *
001C82 C000
001C84 C000000000000000             1986 SVRCSM    DC    18A(0)           SVRDSM
                                    1987 *
001CCC FFFFFFFF                     1988 RCRD      DC    F'-1'            RCRD
                                    1989 *
                                    1990 ****   -EOF-  LEADS TO  ABEND 14
                                    1991 *
                                    1992 EOF       ABEND 14,DUMP         -EOF-
                                    1999 *
001CE0                              2000          LTORG
001CE0 00002134                     2001                =A(IN+80)
001CE4 40404040                     2002                =CL4' '
001CE8 C00020E5                     2003                =A(IN1)
001CEC 000CC000                     2004                =V(READCARD)
001CF0 0000214A00000001             2005                =A(LINOUT+21,1,LINOUT+100)
001CFC C00020E4                     2006                =A(IN)
001DC0 040C                         2007                =P'40'
001DC2 402020202120                 2008                =X'402020202120'
001DC8 1C                           2009                =P'1'
001DC9 0C                           2010                =P'0'
001DCA 4040404040404040             2011                =CL133' '
                                    2012 *
                                    2013          PRINT GEN                                00013361
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   2015 ****  PRNTMES  -  PRINT MESSAGE SUBROUTINE
                                   2016 *
                                   2017 *
                                   2018 PRNTMES  SAVE   (14,12),,*      SAVE REGISTERS
001D8F 00
001D90 47F0 F00C             0000C 2019+PRNTMES  B      12(0,15) BRANCH AROUND ID
001D94 07                          2020+          DC     AL1(7)
001D95 D7D9D5E3D4C5E2              2021+          DC     CL7'PRNTMES' IDENTIFIER
001D9C 90EC D00C             0000C 2022+          STM    14,12,12(13) SAVE REGISTERS
                                   2023 *
001DA0 05C0                        2024           BALR   12,0           ESTABLISH  REG 12  AS THE BASE REGISTER
001DA2                             2025           USING  *,12
                                   2026 *
001DA2 4180 C0AE             01E50 2027           LA     11,SVPRMES     GET THE ADDRESS OF THE SAVE AREA
001DA6 5080 D008             00008 2028           ST     11,8(,13)      STORE FORWARD POINTER
001DAA 50D0 B004             00004 2029           ST     13,4(,11)      STORE BACKWARD POINTER
001DAE 18DB                        2030           LR     13,11          SET UP  REG 13  FOR SAVE AREA
                                   2031 *
001DB0 5820 1000             00000 2032           L      2,0(,1)         -2- POINTS AT -PRNTMES- CALLING SEQUENCE
001DB4 9012 C0FE             01EA0 2033           STM    1,2,SV12PRMS   SAVE -1- AND -2-
                                   2034 *
001DB8 4830 2000             00000 2035           LH     3,0(,2)         LOAD -3- WITH -KT-,THE LENGTH OF MESSAGE
001DBC 5030 C2CE             02070 2036           ST     3,KT           SAVE -KT-  ( THE LENGTH OF THE MESSAGE )
                                   2037 *
001DC0 1233                        2038           LTR    3,3             TEST -KT-
001DC2 4740 C116             01EB8 2039           BM     ERRORKTL         ... KT < 0, GO TO -ERRORKTL-
                                   2040 *
001DC6 4780 C08A             01E2C 2041           BZ     KTZERO           ... KT = 0, GO TO -KTZERO-
                                   2042 *
001DCA 5930 C106             01EA8 2043           C      3,=F'80'       IS -KT- GREATER THAN -80-
001DCE 4720 C126             01EC8 2044           BH     ERRORKTH         ... YES,  GO TO -ERRORKTH-
                                   2045 *
001DD2 0501 2002 C112 00002  01EB4 2046           CLC    2(2,2),=C'*N'   IS  PC = C'*N'
001DD8 4770 C080             01E22 2047           BNE    PCZERO           ... NO,  GO TO -PCZERO-
                                   2048 *
001DDC 58F0 C10A             01EAC 2049 PP1       L      15,=A(PRINTP)    ... YES,  -PRINTP-
001DE0 05EF                        2050           BALR   14,15
                                   2051 *
001DE2 5840 C33E             020E0 2052 TEST1     L      4,POUT          LOAD -4- WITH -TEST-
001DE6 1A43                        2053           AR     4,3               WHERE
001DE8 5040 C2D6             02078 2054           ST     4,TEST              TEST <--- POUT + KT
                                   2055 *
001DEC 5940 C10E             01EB0 2056           C      4,=A(OUT+80)    IS -TEST- GREATER THAN -80-
001DF0 4720 C03A             01DDC 2057           BH     PP1              ... NO,  GO TO -PP1-  (PRINTP)
                                   2058 *
001DF4 5850 C33E             020E0 2059           L      5,POUT          LOAD -5- WITH -MOVE-
001DF8 4150 5001             00001 2060           LA     5,1(,5)           WHERE
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
001DFC 5050 C2D2             02074  2061           ST    5,MOVE              MOVE <--- POUT + 1
001E00 5040 C33E             020E0  2062           ST    4,POUT            POUT <--- TEST
001E04 0630                         2063           BCTR  3,0                ADJUST -3- (KT) FOR -EX-
001E06 4430 C07A             01E1C  2064           EX    3,MOVE1
                                    2065 *
001E0A 58D0 D004             00004  2066           L     13,4(,13)         RESTORE REG -13-
                                    2067 *
                                    2068           RETURN (14,12),T,RC=0   RESTORE REGISTERS AND RETURN
001E0E 98EC D00C             0000C  2069+          LM    14,12,12(13) RESTORE THE REGISTERS
001E12 92FF D00C       Q000C        2070+          MVI   12(13),X'FF' SET RETURN INDICATION
001E16 41F0 0000             00000  2071+          LA    15,0(0,0) LOAD RETURN CODE
001E1A 07FE                         2072+          BR    14 RETURN
                                    2073 *
001E1C D200 5000 2004 00000 00004  2074 MOVE1     MVC   0(1,5),4(2)       MOVE1
                                    2075 *
001E22 D501 2002 C114 00002 01EB6  2076 PCZERO    CLC   2(2,2),=X'0000'   IS  PC = X'0000'
001E28 4780 C040             01DE2  2077           BE    TEST1              ... YES,  GO TO -TEST1-
                                    2078 *
001E2C D501 2002 C112 00002 01EB4  2079 KTZERO    CLC   2(2,2),=C'*N'     IS  PC = C'*N'
001E32 4770 C146             01EE8  2080           BNE   ERRPCKTO           ... NO,  GO TO -ERRPCKTO-
                                    2081 *
001E36 58F0 C10A             01EAC  2082           L     15,=A(PRINTP)      ... YES,  -PRINTP-
001E3A 05EF                         2083           BALR  14,15
                                    2084 *
001E3C 58D0 D004             00004  2085           L     13,4(,13)         RESTORE REG -13-
                                    2086 *
                                    2087           RETURN (14,12),T,RC=0   RESTORE REGISTERS AND RETURN
001E40 98EC D00C             0000C  2088+          LM    14,12,12(13) RESTORE THE REGISTERS
001E44 92FF D00C       0000C        2089+          MVI   12(13),X'FF' SET RETURN INDICATION
001E48 41F0 0000             00000  2090+          LA    15,0(0,0) LOAD RETURN CODE
001E4C 07FE                         2091+          BR    14 RETURN
                                    2092 *
001E4E C000
001E50 0000000000000000             2093 SVPRMES  DC    18A(0)             SVPRMES
                                    2094 *
001E58 000C000000000000             2095 SV12RDSM DC    2A(0)             SV12RDSM
001EA0 000C000000000000             2096 SV12PRMS DC    2A(0)             SV12PRMS
                                    2097 *
001EA8                              2098           LTORG
001EA8 00000050                     2099                 =F'80'
001EAC 0000EFB                      2100                 =A(PRINTP)
001EB0 C000220A                     2101                 =A(OUT+80)
001EB4 5C05                         2102                 =C'*N'
001EB6 0000                         2103                 =X'0C00'
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                    2105 ERRORKTL ABEND 91,DUMP          -ERRORKTL-
001EB8                              2106+          CNOP  0,4
001EB8 47F0 C11E            01ECO   2107+ERRORKTL B     *+8 BRANCH AROUND CONSTANT
001EBC 80                           2108+          DC    AL1(128) DUMP/STEP CODE
001EBD 00005B                       2109+          DC    AL3(91) COMPLETION CODE
001EC0 5810 C11A            01EBC   2110+          L     1,*-4 LOAD CODES INTO REG 1
001EC4 0A0D                         2111+          SVC   13 LINK TO ABEND ROUTINE
                                    2112 *
                                    2113 ERRORKTH ABEND 92,DUMP          -ERRORKTH-
001EC6 0700                         2114+          CNOP  0,4
001EC8 47F0 C12E            01ED0   2115+ERRORKTH B     *+8 BRANCH AROUND CONSTANT
001ECC 80                           2116+          DC    AL1(128) DUMP/STEP CODE
001ECD 00005C                       2117+          DC    AL3(92) COMPLETION CODE
001ED0 5810 C12A            01ECC   2118+          L     1,*-4 LOAD CODES INTO REG 1
001ED4 0A0D                         2119+          SVC   13 LINK TO ABEND ROUTINE
                                    2120 *
                                    2121 PCERROR  ABEND 93,DUMP          -PCERROR-
001ED6 0700                         2122+          CNOP  0,4
001ED8 47F0 C13E            01EE0   2123+PCERROR  B     *+8 BRANCH AROUND CONSTANT
001EDC 80                           2124+          DC    AL1(128) DUMP/STEP CODE
001EDD 00005D                       2125+          DC    AL3(93) COMPLETION CODE
001EE0 5810 C13A            01EDC   2126+          L     1,*-4 LOAD CODES INTO REG 1
001EE4 0A0D                         2127+          SVC   13 LINK TO ABEND ROUTINE
                                    2128 *
                                    2129 ERRPCKTO ABEND 94,DUMP          -ERRPCKTO-
001EE6 0700                         2130+          CNOP  0,4
001EE8 47F0 C14E            01EF0   2131+ERRPCKTO B     *+8 BRANCH AROUND CONSTANT
001EEC 80                           2132+          DC    AL1(128) DUMP/STEP CODE
001EED 00005E                       2133+          DC    AL3(94) COMPLETION CODE
001EF0 5810 C14A            01ECC   2134+          L     1,*-4 LOAD CODES INTO REG 1
001EF4 0A0D                         2135+          SVC   13 LINK TO ABEND ROUTINE
                                    2136 *
001EF8                              2137           LTORG
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    2139 ****     PRINTP  -  PRINT LINE SUBROUTINE
                                    2140 *
                                    2141 *
                                    2142 PRINTP   SAVE   (14,12),,*      SAVE REGISTERS
001EF8 47F0 F00C             0000C  2143+PRINTP   B      12(0,15) BRANCH AROUND ID
001EFC 06                           2144+          DC     AL1(6)
001EFD D7D9C9D5E3D7                 2145+          DC     CL6'PRINTP' IDENTIFIER
001F03 00
001F04 90EC D00C             0000C  2146+          STM    14,12,12(13) SAVE REGISTERS
                                    2147 *
001F08 05C0                         2148           BALR   12,0            ESTABLISH  REG 12  AS THE BASE REGISTER
001F0A                              2149           USING *,12
                                    2150 *
001F0A 41B0 C022             01F2C  2151           LA     11,SVPRINTP     GET THE ADDRESS OF THE SAVE AREA
001F0E 50B0 D008             00008  2152           ST     11,8(,13)       STORE FORWARD POINTER
001F12 50D0 B004             00004  2153           ST     13,4(,11)       STORE BACKWARD POINTER
001F16 18DB                         2154           LR     13,11           SET UP  REG 13  FOR SAVE AREA
                                    2155 *
001F18 FA20 C1CE C312 020D8  0221C  2156           AP     KPRINT,=P'1'     KPRINT <--- KPRINT + 1
                                    2157 *
001F1E 4700 C06A             01F74  2158 SWTCHP    NOP    LNPR1            FIRST P
                                    2159 *
001F22 96F0 C015       01F1F        2160           OI     SWTCHP+1,X'F0'   RESET FIRST P
001F26 47F0 C0AC             01FB6  2161           B      PRHDP
                                    2162 *
001F2A 0000
001F2C 0000C00000000000             2163 SVPRINTP DC     18A(0)          SVPRINTP
                                    2164 *
001F74 FA10 C1C3 C312 020CD  0221C  2165 LNPR1     AP     LNPR,=P'1'       LNPR <--- LNPR + 1
                                    2166 *
001F7A F911 C1C3 C30A 020CD  02214  2167           CP     LNPR,=P'40'      END OF PAGE
001F80 47D0 C1C6             02010  2168           BNH    PRNTLNPR          .. NO
                                    2169 *
                                    2170 ****  PRINT PAGE FOOTING
                                    2171 *
001F84 D284 C22B C4A2 02135  023AC  2172           MVC    LINOUT,IDOTMARK .. YES
001F8A 92F0 C22B       02135        2173           MVI    LINOUT,C'0'
                                    2174 *
                                    2175           PRNTO LINOUT
001F8E 0700                         2176+          CNOP   0,4
001F90 4510 C092             01F9C  2177+          BAL    1,*+12
001F94 00002135                     2178+          DC     A(LINOUT)
001F98 C00CC000                     2179+          DC     V(PRNTBUFO)
001F9C 58F0 C08E             01F98  2180+          L      15,*-4
001FA0 05EF                         2181+          BALR   14,15
                                    2182 *
                                    2183           PRINT NOGEN                                          00014651
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
                                    2184 *
                                    2185          PRNTO TTLELN
                                    2192 *
                                    2193 *
                                    2194 ****  PRINT PAGE HEADING
                                    2195 *
001FB6 D284 C22B C41E 02135 02328   2196 PRHDP     MVC   LINOUT,HEADOUT
001FBC FA10 C1C9 C312 020D3 0221C   2197           AP    PAGEOUT,=P'1'
001FC2 DE03 C29D C1C9 021A7 020D3   2198           ED    LINOUT+114(4),PAGEOUT
                                    2199 *
                                    2200          PRNTO LINOUT
                                    2207 *
001FCA D284 C22B C527 02135 02431   2208           MVC   LINOUT,TTLELN
001FE0 92F0 C22B      02135         2209           MVI   LINOUT,C'0'
                                    2210 *
                                    2211          PRNTO LINOUT
                                    2218 *
                                    2219          PRNTO IDOTMARK
                                    2226 *
0020CA F81C C1C3 C313 020CD 0221D   2227           ZAP   LNPR,=P'0'      LNPR <--- 0
                                    2228 *
                                    2229 *
                                    2230 ****  PRINT LINE IMAGE
                                    2231 *
002010 D284 C22B C314 02135 0221E   2232 PRNTLNPR MVC   LINOUT,=CL133' '
002016 D205 C235 C30C 0213F 02216   2233           MVC   LINOUT+10(6),=X'402020202120'
00201C DE05 C235 C1CB 0213F 020D5   2234           ED    LINOUT+10(6),KREAD
002022 D24F C240 C2B1 0214A 0216B   2235           MVC   LINOUT+21(80),OUT+1
002028 D205 C294 C30C 0219E 02216   2236           MVC   LINOUT+105(6),=X'402020202120'
00202E DE05 C294 C1CE 0219F 020D8   2237           ED    LINOUT+105(6),KPRINT
                                    2238 *
                                    2239          PRNTO LINOUT
                                    2246 *
                                    2247          PRINT GEN                                          00014971
                                    2248 *
002046 D250 C172 C2B0 0207C 021BA   2249           MVC   LASTOUT,OUT     SAVE LATEST IMAGE IN -LASTOUT-
                                    2250 *
00204C 9240 C2B0      021BA         2251           MVI   OUT,C' '        BLANK -OUT-
002050 D24F C2B1 C2B0 021BB 021BA   2252           MVC   OUT+1(80),OUT
                                    2253 *
002056 5840 C306            02210   2254           L     4,=A(OUT)       POUT <--- LOC(OUT)
00205A 5040 C1D6            020E0   2255           ST    4,POUT
                                    2256 *
00205E 58D0 D004            00004   2257           L     13,4(,13)       RESTORE REG -13-
                                    2258 *
                                    2259          RETURN (14,12),T,RC=0   RESTORE REGISTERS AND RETURN
002062 98EC D00C            0000C   2260+          LM    14,12,12(13) RESTORE THE REGISTERS
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
002066 92FF D00C       0000C        2261+          MVI   12(13),X'FF' SET RETURN INDICATION
002C6A 41F0 0000             00000  2262+          LA    15,0(0,0) LOAD RETURN CODE
00206E 07FE                         2263+          BR    14 RETURN
                                    2264 *
002C70 FFFFFFFF                     2265 KT        DC    F'-1'           KT
                                    2266 *
002074                              2267 MOVE      DS    F               MOVE
002C78                              2268 TEST      DS    F               TEST
                                    2269 *
00207C 4040404040404040             2270 LASTOUT   DC    CL81' '          LASTOUT
                                    2271 *
0020CD C00C                         2272 LNPR      DC    PL2'0'
0020CF 000C                         2273 LNRD      DC    PL2'0'          LNRD
                                    2274 *
0020D1 000C                         2275 PAGEIN    DC    PL2'0'           PAGEIN
0020D3 C00C                         2276 PAGEOUT   DC    PL2'0'          PAGEOUT
                                    2277 *
0020D5 00000C                       2278 KREAD     DC    PL3'0'           KREAD  XX  XX  XS
0020D8 C0000C                       2279 KPRINT    DC    PL3'0'           KPRINT
                                    2280 *
0020DB 00
0020DC C0002135                     2281 PTIN      DC    A(IN+81)        PTIN
0020E0 0000220B                     2282 POUT      DC    A(OUT+81)       POUT
                                    2283 *
0020E4                              2284 IN        DS    C               IN
0020E5                              2285 IN1       DS    80C             IN1
                                    2286 *
002135 4040404040404040             2287 LINOUT    DC    CL133' '        LINOUT                    00015330
00218A 4040404040404040             2288 OUT       DC    CL81' '         OUT                       00015340
                                    2289 *
002210                              2290           LTORG
002210 C000218A                     2291                 =A(OUT)
002214 040C                         2292                 =P'40'
002216 402020202120                 2293                 =X'402020202120'
00221C 1C                           2294                 =P'1'
00221D 0C                           2295                 =P'0'
00221E 4040404040404040             2296                 =CL133' '
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
0022A3 F140404040404040            2298 HEADIN   DC    CL31'1'
0022C2 60C9D5D7E4E36040            2299          DC    CL12'-INPUT-'
0022CE E2C9D5C7D3C560E2            2300          DC    CL41'SINGLE-STACK SYMBOL STRING PROCESSOR'
0022F7 60C9D5D7E4E36040            2301          DC    CL26'-INPUT-'
002311 D7C1C7C5                    2302          DC    CL4'PAGE'
002315 40202120                    2303 PGIN     DC    X'40202120'     OUT+114
002319 4040404C40404040            2304          DC    CL15' '
                                   2305 *
002328 F140404040404040            2306 HEADOUT  DC    CL30'1'
002346 60D6E4E3D7E4E360            2307          DC    CL12'-OUTPUT-'
002352 E2C9D5C7D3C560E2            2308          DC    CL41'SINGLE-STACK SYMBOL STRING PROCESSOR'
00237B 60D6E4E3D7E4E360            2309          DC    CL26'-OUTPUT-'
002395 D7C1C7C5                    2310          DC    CL4'PAGE'
002399 40202120                    2311 PGOUT    DC    X'40202120'
00239D 4040404040404040            2312          DC    CL15' '
                                   2313 *
0023AC                             2314 IDOTMARK DS    0CL133
0023AC 4040404040404C40            2315          DC    CL20' '
0023C0 4E                          2316          DC    C'+'
0023C1 4B4B4B4B4E4B4B4B            2317          DC    16CL5'....+'
002411 404C404040404040            2318          DC    CL32' '
                                   2319 *
002431                             2320 TTLELN   DS    0CL133
002431 4040404040404040            2321          DC    CL13' '
00243E C9D5404040404040            2322          DC    CL8'IN'
002446 F1404040F5404040            2323          DC    CL40'1   5   10   15   20   25   30   35   40'
00246E 404C40F4F5404040            2324          DC    CL40'   45   50   55   60   65   70   75   80'
002496 40404040404040              2325          DC    CL6' '
00249C D6E4E34040404040            2326          DC    CL26'OUT'
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
002488                           2328 STACK    CSECT
002488 404040C8C8404040          2329          DC     CL8'   HH   '
0024C0 000CC000                  2330 HH       DC     F'0'          HH
0024C4 404040D7D7404040          2331          DC     CL8'   PP   '
0024CC 000CCC00                  2332 PP       DC     F'0'          PP
0024C0 404C40C6C6404040          2333          DC     CL8'   FF   '
0024C8 0000C000                  2334 FF       DC     F'0'          FF
0024DC 404040C3C3404040          2335          DC     CL8'   CC   '
0024E4 000CC000                  2336 CC       DC     F'0'          CC
0024E8 404C40C1C1404040          2337          DC     CL8'   AA   '
0024F0 C00CCC00                  2338 AA       DC     F'0'          AA
0024F4 404C40E6E6404040          2339          DC     CL8'   WW   '
0024FC 0000C000                  2340 WW       DC     F'0'          WW
0025C0 404040E2E2404040          2341          DC     CL8'   SS   '
0025C8 000CC000                  2342 SS       DC     F'0'          SS
                                 2343 *
0025CC 404040D8D8404040          2344          DC     CL8'   QQ   '
002514 C00CC000                  2345 QQ       DC     F'0'          QQ
002518 404040C5C5404040          2346          DC     CL8'   EE   '
002520 000CC000                  2347 EE       DC     F'0'          EE
                                 2348 *
002524 40D4C1D9D2C5D940          2349          DC     CL8' MARKER '
00252C 6F404040                  2350 MARKER   DC     C'?   '
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT
002530 000CC00000000000            2352 SVSTART  DC     16A(0)          SVSTART
002570 C000000000000000            2353 SAVECOPY DC     16A(0)          SAVECOPY
0025B0 000CC00000000000            2354 SVENDFN  DC     16A(0)          SVENDFN
0025F0 000CC00000000000            2355 SAVEFN   DC     16A(0)          SAVEFN
002630 C000000000000000            2356 SVFINDX  DC     16A(0)          SVFINDX
002670 C00C000000000000            2357 SAVELOAD DC     16A(0)          SAVELOAD
0026B0 C000C00000000000            2358 SVNEXTCH DC     16A(0)          SVNEXTCH
0026F0 C00CC00C00000000            2359 SVNXTITM DC     16A(0)          SVNXTITM
002730 CCC000C000000000            2360 SVLOARG  DC     16A(0)          SVLOARG
002770 C00CC000C0000000            2361 SVAPPLY  DC     16A(0)          SVAPPLY
0027B0 C000C00000000000            2362 SAVEDEF  DC     16A(0)          SAVEDEF
0027F0 C000C00000000000            2363 SAVEVAL  DC     16A(0)          SAVEVAL
002830 C00CC00000000000            2364 SVUPDATE DC     16A(0)          SVUPDATE
002870 C000000000000000            2365 SAVEBIN  DC     16A(0)          SAVEBIN
0028B0 0000C00000000000            2366 SAVEDEC  DC     16A(0)          SAVEDEC
0028F0 C00CC00000000000            2367 SAVEBAR  DC     16A(0)          SAVEBAR
002930 000C000000000000            2368 SAVEEOB  DC     16A(0)          SAVEEOB
002970 000CC00000000000            2369 SAVESIGN DC     16A(0)          SAVESIGN
0029B0 000CC00000000000            2370 SVITEM   DC     16A(0)          SVITEM
0029F0 C00CC00000000000            2371 SAVEMON1 DC     16A(0)          SAVEMON1
002A30 C000C00000000000            2372 SAVEMON2 DC     16A(0)          SAVEMON2
002A70 C00CC00000000000            2373 SAVEMON3 DC     16A(0)          SAVEMON3
002AB0 C000C00000000000            2374 SAVEMON4 DC     16A(0)          SAVEMON4
002AF0 0000C00000000000            2375 SAVEMON5 DC     16A(0)          SAVEMON5
002B30 C000C00000000000            2376 SAVEMON7 DC     16A(0)          SAVEMON7
002B70 0000C00000000000            2377 SVMON8   DC     16A(0)          SVMON8
002BB0 C000C00000000000            2378 SAVEMON9 DC     16A(0)          SAVEMON9
002BF0 0000C00000000000            2379 SVMON11  DC     16A(0)          SVMON11
```

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT

                                   2381 ****     COUNTERS
                                   2382 *
                                   2383 *
002C30                             2384          DS    0D
002C30 40404040D2E3E2E3            2385          DC    CL16'    KTSTART     '
002C40 0000C00C                    2386 KTSTART  DC    PL4'0'         KTSTART      00002002
002C44 40404040D2E3D3D6            2387          DC    CL16'    KTLOAD      '
002C54 0000C00C                    2388 KTLOAD   DC    PL4'0'         KTLOAD       00002332
002C58 40404040D2E3D5C5            2389          DC    CL16'    KTNEXTCH    '
002C68 000CC00C                    2390 KTNEXTCH DC    PL4'0'         KTNEXTCH     00002652
002C6C 40404040D2E3C5D5            2391          DC    CL16'    KTENDFN     '
002C7C 0000C00C                    2392 KTENDFN  DC    PL4'0'         KTENDFN      00003692
002C80 40404040D2E3C6D5            2393          DC    CL16'    KTFN        '
002C90 0000C00C                    2394 KTFN     DC    PL4'0'         KTFN
002C94 40404040D2E3D4C1            2395          DC    CL16'    KTMARK      '
002CA4 0000C00C                    2396 KTMARK   DC    PL4'0'         KTMARK       00004032
002CA8 40404040D2E3C6C9            2397          DC    CL16'    KTFINDX     '
002CB8 C000C00C                    2398 KTFINDX  DC    PL4'0'         KTFINDX      00004502
002CBC 40404040D2E3D5E7            2399          DC    CL16'    KTNXTITM    '
002CCC 0000C00C                    2400 KTNXTITM DC    PL4'0'         KTNXTITM     00004877
002CD0 40404040D2E3D3C4            2401          DC    CL16'    KTLDARG     '
002CE0 C00CC00C                    2402 KTLDARG  DC    PL4'0'         KTLDARG      00005432
002CE4 40404040D2E3C1D7            2403          DC    CL16'    KTAPPLY     '
002CF4 0000C00C                    2404 KTAPPLY  DC    PL4'0'         KTAPPLY      00005072
002CF8 404C4040D2E3C4C5            2405          DC    CL16'    KTDEF       '
002D08 0000C00C                    2406 KTDEF    DC    PL4'0'         KTDEF        00006342
002D0C 40404040D2E3E5C1            2407          DC    CL16'    KTVAL       '
002D1C 0000C00C                    2408 KTVAL    DC    PL4'0'         KTVAL        00005642
002D20 40404040D2E3E4D7            2409          DC    CL16'    KTUPDATE    '
002D30 0000C00C                    2410 KTUPDATE DC    PL4'0'         KTUPDATE     00006962
002D34 40404040D2E3C2C9            2411          DC    CL16'    KTBIN       '
002D44 000CC00C                    2412 KTBIN    DC    PL4'0'         KTBIN        00007472
002D48 40404040D2E3C4C5            2413          DC    CL16'    KTDEC       '
002D58 000CC00C                    2414 KTDEC    DC    PL4'0'         KTDEC        00008012
002D5C 40404040D2E3C2C1            2415          DC    CL16'    KTBAR       '
002D6C 0000C00C                    2416 KTBAR    DC    PL4'0'         KTBAR        00008422
002D70 40404040D2E3C5D6            2417          DC    CL16'    KTEOB       '
002D80 0000C00C                    2418 KTEOB    DC    PL4'0'         KTEOB        00008582
002D84 40404040D2E3E2C9            2419          DC    CL16'    KTSIGN      '
002D94 0000000C                    2420 KTSIGN   DC    PL4'0'         KTSIGN
```

```
LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

2D98                              2422 STST     DS    0D              STST
                                  2423 *
2D98 FFFFFFFF                     2424 XDEF     DC    F'-1'           XDEF
2D9C 00000010                     2425          DC    F'16'
2DA0 C4404040C5404040             2426          DC    CL4'D',CL4'E',CL4'F'
2DA8 FFFFFFFF                     2427          DC    F'-1'
2DAC 6F404040                     2428          DC    CL4'?'
                                  2429 *
2DB4 00002F18                     2430 XVAL     DC    A(XXDEF)        XVAL
2DB8 00000010                     2431          DC    F'16'
2DBC E5404040C1404040             2432          DC    CL4'V',CL4'A',CL4'L'
2DC8 FFFFFFFE                     2433          DC    F'-2'
2DCC 6F404040                     2434          DC    CL4'?'
                                  2435 *
2DD0 00002F34                     2436 XUPDATE  DC    A(XXVAL)        XUPDATE
2DD4 00000010                     2437          DC    F'28'
2DD8 E4404040D7404040             2438          DC    CL4'U',CL4'P',CL4'D',CL4'A',CL4'T',CL4'E'
2DE0 FFFFFFFD                     2439          DC    F'-3'
2DE4 6F404040                     2440          DC    CL4'?'
                                  2441 *
2DE8 00002F50                     2442 XBIN     DC    A(XXUPDATE)     XBIN
2DEC 00000010                     2443          DC    F'16'
2DF0 C2404040C9404040             2444          DC    CL4'B',CL4'I',CL4'N'
2DF8 FFFFFFFC                     2445          DC    F'-4'
2DFC 6F404040                     2446          DC    CL4'?'
                                  2447 *
2E00 00002F78                     2448 XDEC     DC    A(XXBIN)        XDEC
2E04 00000010                     2449          DC    F'16'
2E08 C4404040C5404040             2450          DC    CL4'D',CL4'E',CL4'C'
2E10 FFFFFFFB                     2451          DC    F'-5'
2E14 6F404040                     2452          DC    CL4'?'
                                  2453 *
2E18 00002F94                     2454 XBAR     DC    A(XXDEC)        XBAR
2E1C 00000010                     2455          DC    F'16'
2E20 C2404040C1404040             2456          DC    CL4'B',CL4'A',CL4'R'
2E28 FFFFFFFA                     2457          DC    F'-6'
2E2C 6F4C4040                     2458          DC    CL4'?'
                                  2459 *
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

002E4C 00002FB0                   2461 XEOB     DC    A(XXBAR)        XEOB
002E50 C000C010                   2462          DC    F'16'
002E54 C5404040D6404040           2463          DC    CL4'E',CL4'O',CL4'B'
002E60 FFFFFFF9                   2464          DC    F'-7'
002E64 6F404040                   2465          DC    CL4'?'
                                  2466 *
002E68 00002FCC                   2467 XSIGN    DC    A(XXEOB)        XSIGN
002E6C 000CC014                   2468          DC    F'20'
002E70 E2404040C9404040           2469          DC    CL4'S',CL4'I',CL4'G',CL4'N'
002E80 FFFFFFF8                   2470          DC    F'-8'
002E84 6F404040                   2471          DC    CL4'?'
                                  2472 *
002E88 C000CC00                   2473 XTOP     DC    F'0'            XTOP
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

002E90                            2475           DS    0D
002E90 4040404040404040           2476           DC    CL128' '
002F10 4040E2E3C1C3D240           2477           DC    CL8'  STACK '
                                  2478 *
002F16                            2479 ST        DS    0F              ST
                                  2480 *
002F18                            2481 XXDEF     DS    7F              XXDEF
002F34                            2482 XXVAL     DS    7F              XXVAL
002F50                            2483 XXUPDATE  DS    10F             XXUPDATE
002F78                            2484 XXBIN     DS    7F              XXBIN
002F94                            2485 XXDEC     DS    7F              XXDEC
002FB0                            2486 XXBAR     DS    7F              XXBAR
002FCC                            2487 XXEOB     DS    7F              XXEOB
002FE8                            2488 XXMAC     DS    0F              XXMAC
002FE8                            2489 XXSIGN    DS    8F              XXSIGN
003008                            2490 XXTOP     DS    0F              XXTOP
003CC8                            2491           DS    3940F
006D70                            2492 ENDST     EQU   *-40            ENDST
                                  2493 *
000CC0                            2494           END   SYMBOL
```

APPENDIX II

Input and Output of the Tests for GPM

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
  1    THE GENERAL PURPOSE MACRO GENERATOR, "GPM", STARTS IN THE "COPY" MODE.                    0
  2                                                                                             1
  3                                                                                             2
  4    THIS MEANS THE "INPUT" STREAM IS COPIED DIRECTLY TO THE "OUTPUT" STREAM.                 3
  5                                                                                             4
  6                                                                                             5
  7    THE PRIMITIVE MACRO 'EOB' IS USED TO FORCE THE "GPM" PROCESSOR TO START A NEW            6
  8     80 COLUMN CARD IMAGE IN THE "OUTPUT" STREAM.  THIS ACTION IS SOMETIMES CALLED          7
  9      LINE FEED OR LINE ADVANCE.                                                            8
 10    #                                                                                       9
 11    $EOB;$EOB;#                                                                             9
 12    THE USE OF THE 'EOB' MACRO AND THE "END-OF-LINE" CHARACTER ARE DEMONSTRATED BY         11
 13     THE PREVIOUS TWO INPUT CARDS.                                                         11
 14                                                                                           12
 15    THE "END-OF-LINE" CHARACTER, AN 0-6-9 PUNCH IN THE INPUT CARD, IS USED TO              13
 16     TERMINATE THE SCANNING PROCESS FOR THAT DATA CARD.  THE SYMBOL #, THE POUND           14
 17      SIGN, IS USED IN THE LISTING OF THE "INPUT" STREAM TO INDICATE THE LOCATION          15
 18       OF THE "END-OF-LINE" CHARACTERS.  NOTICE THAT THE "END-OF-LINE" CHARACTER           16
 19        HAS NO EFFECT ON THE LISTING OF THE "OUTPUT" STREAM, WHERE THE 'EOB'               17
 20         PRIMITIVE MACRO MAKES ITS PRESENCE KNOWN.  THE USE OF THE "END-OF-LINE"          18
 21          CHARACTER, THE 0-6-9 PUNCH, ALLOWS THE USER OF "GPM" TO ELIMINATE HOLDING       19
 22           UNUSED BLANKS ON THE "STACK".  THE USE OF THE 'EOB' MACRO IN CONTROLLING       20
 23            THE APPEARANCE OF THE LISTING OF THE "OUTPUT" STREAM IS SELF-EVIDENT.         21
 24    #                                                                                      22
 25    #  THE USE OF THE END-OF-LINE CHARACTER, THE 0-6-9 PUNCH, ALLOWS COMMENTS TO BE        22
 26    #   INSERTED IN THE INPUT STREAM (THE INPUT STRING).  THESE COMMENTS WILL NOT         22
 27    #    APPEAR IN THE LISTING OF THE OUTPUT STREAM.                                      22
 28    #                                                                                     22
 29    $DEF,5 BLANKS,     ;#                                                                 22
 30    $EOB;$EOB;INSERT BLANKS - 12$5 BLANKS;34#                                            24
 31    #                                                                                     24
 32    $EOB;$EOB;THE MACRO BODY OF  5 BLANKS  CAN BE EXHIBITED USING THE PRIMITIVE #        26
 33    MACRO "VAL".#                                                                        26
 34    #                                                                                    26
 35    $EOB;$EOB;..$VAL,5 BLANKS;.. THE "5 BLANKS" APPEAR BETWEEN THE SETS OF TWO DOTS.     26
 36    #                                                                                    28
 37    #  USE OF THE "QUOTES"                                                               28
 38    #                                                                                    28
 39    $DEF,1 BLANK, ;#              THE SINGLE BLANK IS NOT QUOTED.                        28
 40    #
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
 41  $DEF,2EOB,<$EOB;$EOB;>;#       THE MACRO BODY IS QUOTED.                                28
 42  #                                                                                       28
 43  #  THE "VLMC" MACRO LISTS A PROGRAMMER MACRO DEFINITION.                                28
 44  #                                                                                       28
 45  #     "VLMC" REPRESENTS  'VALUE OF THE MACRO'.                                          28
 46  #                                                                                       28
 47  $DEF,VLMC,<$2EOB;MACRO DEFINITION#                                                      28
 48  $EOB;  MACRO NAME - ¬1#                                                                 28
 49  $EOB;  MACRO BODY - $VAL,¬1;>;#                                                         28
 50  #                                                                                       28
 51  $2EOB;"VLMC"  -  REPRESENTS THE "VALUE OF THE MACRO".#                                  28
 52  $VLMC,VLMC;#                                                                            30
 53  #                                                                                       35
 54  #   ADDITIONAL USES OF THE "QUOTES"                                                     35
 55  #                                                                                       35
 56  $DEF,TH1,THERESA;#                                                                      35
 57  $DEF,TH2,<HELLNER>;#                        THIS NEED NOT BE "QUOTED".                  35
 58  $DEF,TH3,<$TH1;$1 BLANK;$TH2;>;#            "QUOTE"                                     35
 59  $DEF,TH4,<$TH1; $TH2;>;#                    "QUOTE"                                     35
 60  $DEF,TH5,<$2EOB;$TH3;>;#                    "QUOTE"                                     35
 61  $DEF,TH6,<$2EOB;$TH4;>;#                    "QUOTE"                                     35
 62  $DEF,TH7,$TH1;$1 BLANK;$TH2;;#              "NO QUOTE"                                  35
 63  $DEF,TH8,$TH1; $TH2;;#                      "NO QUOTE"                                  35
 64  #                                                                                       35
 65  $VLMC,TH1;$VLMC,TH2;$VLMC,TH3;$VLMC,TH4;#                                               35
 66  $VLMC,TH5;$VLMC,TH6;$VLMC,TH7;$VLMC,TH8;#                                               51
 67  #                                                                                       67
 68  $2EOB;MACRO-CALLS TO "TH1","1 BLANK", AND "TH2" GIVE $TH1;$1 BLANK;$TH2;#               67
 69  #                                                                                       69
 70  $DEF,SEETH,<$EOB;A CALL TO  ¬1  GIVES ..$VAL,¬1;..>;#                                   69
 71  #                                                                                       69
 72  $VLMC,SEETH;#                                                                           69
 73  #                                                                                       73
 74  $2EOB;$SEETH,SEETH;#                                                                    73
 75  #                                                                                       76
 76  $EOB;#                                                                                  76
 77  $SEETH,TH1;$SEETH,TH2;$SEETH,TH3;$SEETH,TH4;#                                           77
 78  $SEETH,TH5;$SEETH,TH6;$SEETH,TH7;$SEETH,TH8;#                                           81
 79  #                                                                                       85
 80  $EOB;$EOB;1ST TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                      85
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
```

```
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
 81   #                                                                                      87
 82   $DEF,ABC,<AB¬1C¬2AB>;#                                                                 87
 83   $DEF,A,<A¬1A>;#                                                                        87
 84   $DEF,B,<B$A,X¬1X;B>;#                                                                  87
 85   $DEF,APA,<P¬1¬1P>;#                                                                    87
 86   #                                                                                      87
 87   $EOB;$EOB;2ND TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                     89
 88   #                                                                                      89
 89   $2EOB;-ABC- MACRO#                                                                     91
 90   #                                                                                      91
 91   $EOB;$ABC,XY,PQ;$EOB;  $ABC,99,88;$EOB;     $ABC,SS,TT;#                              94
 92   #                                                                                      94
 93   $2EOB;-A- MACRO$EOB;$A,C;$EOB;  $A,ACA;$EOB;     $A,$A,C;;$EOB;       $A,XDX;#       100
 94   #                                                                                     100
 95   $2EOB;-B- MACRO$EOB;$B,D;$EOB;  $B,GGG;$EOB;     $B,$A,H;;#                          105
 96   #                                                                                     105
 97   $2EOB;-APA- MACRO#                                                                    107
 98   #                                                                                     107
 99   $EOB;$APA,Y;$EOB;  $$A,P;,Y;$EOB;     $$A,P;,$A,Z;;#                                 110
100   #                                                                                     110
101   $EOB;$EOB;3RD TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                    112
102   #                                                                                     112
103   $EOB;Q<$A,C;>R#                                                                       113
104   $EOB;   $A,<$A,X;>;#                                                                  114
105   $EOB;     $B,<$A,X;>;#                                                                115
106   $EOB;       Q<$>R<;>#                                                                 116
107   $EOB;         Q<<$A,C;>>R#                                                            117
108   #                                                                                     117
109   $EOB;$EOB;4TH TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                    119
110   #                                                                                     119
111   $2EOB;-SUC- MACRO#                                                                    121
112   #                                                                                     121
113   $DEF,SUC,<$1,2,3,4,5,6,7,8,9,10,$DEF,1,<¬>¬1;;>;#                                     121
114   #                                                                                     121
115   $VLMC,SUC;#                                                                           125
116   #                                                                                     125
117   $2EOB;USE -SUC- MACRO#                                                                127
118   $EOB;#                                                                                128
119   $SUC,0; $SUC,1; $SUC,2; $SUC,3; $SUC,4; $SUC,5; $SUC,6; $SUC,7; $SUC,8; $SUC,9;       128
120   $EOB;#

     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
```

```
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
121  $SUC,0;$EOB;$SUC,6;$EOB;$SUC,9;$EOB;#                                                  129
122  $EOB;#                                                                                 132
123  $SUC,0; $SUC,1; $SUC,2; $SUC,3; $SUC,4; $SUC,5; $SUC,6; $SUC,7; $SUC,8; $SUC,9;       133
124  #                                                                                      133
125  $2EOB;-SUCCESSOR- MACRO#                                                               133
126  #                                                                                      135
127  $DEF,SUCCESSOR,<$¬2,$DEF,¬2,¬1<,$SUC,>¬2<;>;$DEF,9,<$SUC,>¬1<;,0>;;>;#                135
128  #                                                                                      135
129  $VLMC,SUCCESSOR;#                                                                      135
130  #                                                                                      139
131  $EOB;#                                                                                 139
132  $EOB;CALLS TO -SUCCESSOR- MACRO$EOB;#                                                  140
133  $EOB;2,2  =  $SUCCESSOR,2,2;     6,9  =  $SUCCESSOR,6,9;#                              142
134  #                                                                                      143
135  $EOB;$EOB;5TH TEMPORARY "A"    ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                      143
136  #                                                                                      145
137  $EOB;2,2  =  $SUCCESSOR,2,2;     6,9  =  $SUCCESSOR,6,9;#                              145
138  $EOB;1,9  =  $SUCCESSOR,1,9;     5,0  =  $SUCCESSOR,5,0;#                              146
139  $EOB;0,0  =  $SUCCESSOR,0,0;     8,8  =  $SUCCESSOR,8,8;#                              147
140  #                                                                                      148
141  $2EOB;DEFINE  +  -  *  QUOT  REM  MACROS#                                              148
142  #                                                                                      150
143  $DEF,+,<$DEC,$BAR,+,$BIN,¬1;,$BIN,¬2;;;>;#                                             150
144  $DEF,-,<$DEC,$BAR,-,$BIN,¬1;,$BIN,¬2;;;>;#                                             150
145  $DEF,*,<$DEC,$BAR,*,$BIN,¬1;,$BIN,¬2;;;>;#                                             150
146  $DEF,QUOT,<$DEC,$BAR,/,$BIN,¬1;,$BIN,¬2;;;>;#                                          150
147  $DEF,REM,<$DEC,$BAR,R,$BIN,¬1;,$BIN,¬2;;;>;#                                           150
148  #                                                                                      150
149  $VLMC,+;$VLMC,-;$VLMC,*;$VLMC,QUOT;$VLMC,REM;#                                         150
150  #                                                                                      170
151  $2EOB;USE  +  -  *  QUOT  REM  MACROS#                                                 170
152  #                                                                                      172
153  $EOB;#                                                                                 172
154  $EOB;2 + 3 = $+,2,3;#                                                                  173
155  $EOB;17 - 19 = $-,17,19;#                                                              174
156  $EOB;(-13) * (-13) = $*,-13,-13;#                                                      175
157  $EOB;(-29)/6 = $QUOT,-29,6;#                                                           176
158  $EOB;REM( (-29)/6 ) = $REM,-29,6;#                                                     177
159  #                                                                                      178
160  $2EOB;-1234 + -191 = $+,-1234,-191;#                                                   178
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
```

```
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
161   $2EOB;-1234 - -191 = $-,-1234,-191;#                                                  180
162   $2EOB;-1234 * -191 = $*,-1234,-191;#                                                  182
163   $EOB;REM( (-9876)/-17 ) = $REM,-9876,-17;#                                            184
164   #                                                                                     185
165   $DEF,SEECALC,<$EOB;¬2 ¬1 ¬3 = $¬1,¬2,¬3;>;$VLMC,SEECALC;#                             185
166   #                                                                                     189
167   $2EOB;#                                                                               189
168   $SEECALC,+,1,2;$SEECALC,*,-7,-19;#                                                    191
169   $SEECALC,*,9999,-9999;$SEECALC,-,-191,-1234;#                                         193
170   #                                                                                     195
171   $EOB;$EOB;6TH TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                    195
172   #                                                                                     197
173   $2EOB;FACTORIAL DEFINITION#                                                           197
174   #                                                                                     199
175   $DEF,FACT,<#                                                                          199
176   $$DEC,$BIN,¬1;;,$DEC,$BIN,¬1;;,#                                                      199
177   $DEF,$DEC,$BIN,¬1;;,<$¬1,¬1,$DEF,¬1,#                                                 199
178   <$*,$FACT,$-,¬1,1;;,¬1;>;$DEF,1,<1>;;#                                                199
179   >;$DEF,0,<1>;;#                                                                       199
180   >;#                                                                                   199
181   #                                                                                     199
182   $VLMC,FACT;#                                                                          199
183   #                                                                                     204
184   $2EOB;#                                                                               204
185   $2EOB;FACTORIALS#                                                                     206
186   $EOB;0 - $FACT,0;#                                                                    208
187   $EOB;1 - $FACT,1;#                                                                    209
188   $EOB;2 - $FACT,2;#                                                                    210
189   $EOB;3 - $FACT,3;#                                                                    211
190   $EOB;4 - $FACT,4;#                                                                    212
191   $EOB;5 - $FACT,5;#                                                                    213
192   $EOB;6 - $FACT,6;#                                                                    214
193   $EOB;7 - $FACT,7;#                                                                    215
194   $EOB;8 - $FACT,8;#                                                                    216
195   $EOB;9 - $FACT,9;#                                                                    217
196   #                                                                                     218
197   $EOB;$EOB;7TH TEMPORARY "A"     ...XUX - $A,X,U,$DEF,A,<¬1¬2¬1>;;#                    218
198   #                                                                                     220
199   $DEF,CNVB1TOD2,<$UPDATE,¬2,$DEC,$VAL,¬1;;;>;#                                         220
200   $DEF,CNVD1TOB2,<$UPDATE,¬2,$BIN,$VAL,¬1;;;>;#                                         220
     +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN    1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80    OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
201    #                                                                                          220
202    $VLMC,CNVD1TOB2;$VLMC,CNVB1TOD2;#                                                          220
203    #                                                                                          228
204    $DEF,0D,+0000000000;$DEF,1D,+0000000001;$DEF,-1D,-0000000001;#                             228
205    #                                                                                          228
206    $DEF,AAA,-37;$DEF,AAB,<-39>;$DEF,AAC,+17;$DEF,AAD,<+19>;#                                  228
207    #                                                                                          228
208    $DEF,SEEDEC,<¬1 = $VAL,¬1;>;$VLMC,SEEDEC;#                                                 228
209    #                                                                                          232
210    $2EOB;$SEEDEC,0D;      $SEEDEC,1D;      $SEEDEC,-1D;#                                      232
211    #                                                                                          234
212    #                                                                                          234
213    $DEF,DEFBIN,<$DEF,¬1,$BIN,¬2;;>;$VLMC,DEFBIN;#                                             234
214    #                                                                                          238
215    $DEFBIN,0B,$VAL,0D;;$DEFBIN,1B,$VAL,1D;;$DEFBIN,-1B,$VAL,-1D;;#                            238
216    #                                                                                          238
217    $DEF,SEEBINVAL,<$DEC,$VAL,¬1;;>;$VLMC,SEEBINVAL;#                                          238
218    #                                                                                          242
219    $2EOB;0B = $SEEBINVAL,0B;      1B = $SEEBINVAL,1B;      -1B = $SEEBINVAL,-1B;#             242
220    #                                                                                          244
221    $DEF,TESTFACT,<$EOB;   ¬1  FACTORIAL IS  $FACT,¬1;>;$VLMC,TESTFACT;#                       244
222    #                                                                                          248
223    $2EOB;FACTORIALS#                                                                          248
224    #                                                                                          250
225    $TESTFACT,0;$TESTFACT,1;$TESTFACT,9;#                                                       250
226    #                                                                                          253
227    $DEF,DEFDEC0,<$DEF,¬1,+0000000000;>;$VLMC,DEFDEC0;#                                        253
228    #                                                                                          257
229    $DEFDEC0,YYA;$DEFDEC0,YYB;$DEFDEC0,YYC;$DEFDEC0,YYD;#                                      257
230    #                                                                                          257
231    $DEFBIN,ZZA,0;$DEFBIN,ZZB,0;$DEFBIN,ZZC,0;$DEFBIN,ZZD,0;#                                  257
232    #                                                                                          257
233    $CNVD1TOB2,AAA,ZZA;#                                                                       257
234    $CNVD1TOB2,AAB,ZZB;#                                                                       257
235    $CNVD1TOB2,AAC,ZZC;#                                                                       257
236    $CNVD1TOB2,AAD,ZZD;#                                                                       257
237    #                                                                                          257
238    $2EOB;#                                                                                    257
239          ZZA = $SEEBINVAL,ZZA;#                                                               259
240          ZZB = $SEEBINVAL,ZZB;#                                                               259
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
241          ZZC = $SEEBINVAL,ZZC;#                                                          259
242          ZZD = $SEEBINVAL,ZZD;#                                                          259
243     #                                                                                    259
244     $CNVB1TOD2,ZZA,YYA;#                                                                 259
245     $CNVB1TOD2,ZZB,YYB;#                                                                 259
246     $CNVB1TOD2,ZZC,YYC;#                                                                 259
247     $CNVB1TOD2,ZZD,YYD;#                                                                 259
248     #                                                                                    259
249     $2EOB;$SEEDEC,YYA;      $SEEDEC,YYB;      $SEEDEC,YYC;      $SEEDEC,YYD;#            259
250     #                                                                                    261
251     $2EOB;#                                                                              261
252     .....  THE END  .....                                                                263
253     $2EOB;#                                                                              263
254     #                                                                                    265
255     >                                                                                    265
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
  1                                                                                            1
  2    THE GENERAL PURPOSE MACRO GENERATOR, "GPM", STARTS IN THE "COPY" MODE.                  2
  3                                                                                            3
  4                                                                                            4
  5    THIS MEANS THE "INPUT" STREAM IS COPIED DIRECTLY TO THE "OUTPUT" STREAM.                5
  6                                                                                            6
  7                                                                                            7
  8    THE PRIMITIVE MACRO 'EOB' IS USED TO FORCE THE "GPM" PROCESSOR TO START A NEW           8
  9     80 COLUMN CARD IMAGE IN THE "OUTPUT" STREAM.  THIS ACTION IS SOMETIMES CALLED          9
 11      LINE FEED OR LINE ADVANCE.                                                          10
 11                                                                                          11
 13    THE USE OF THE 'EOB' MACRO AND THE "END-OF-LINE" CHARACTER ARE DEMONSTRATED BY        12
 14     THE PREVIOUS TWO INPUT CARDS.                                                        13
 15                                                                                          14
 16    THE "END-OF-LINE" CHARACTER, AN 0-6-9 PUNCH IN THE INPUT CARD, IS USED TO            15
 17     TERMINATE THE SCANNING PROCESS FOR THAT DATA CARD.  THE SYMBOL #, THE POUND          16
 18      SIGN, IS USED IN THE LISTING OF THE "INPUT" STREAM TO INDICATE THE LOCATION        17
 19       OF THE "END-OF-LINE" CHARACTERS.  NOTICE THAT THE "END-OF-LINE" CHARACTER         18
 20        HAS NO EFFECT ON THE LISTING OF THE "OUTPUT" STREAM, WHERE THE 'EOB'             19
 21         PRIMITIVE MACRO MAKES ITS PRESENCE KNOWN.  THE USE OF THE "END-OF-LINE"        20
 22          CHARACTER, THE 0-6-9 PUNCH, ALLOWS THE USER OF "GPM" TO ELIMINATE HOLDING     21
 23           UNUSED BLANKS ON THE "STACK".  THE USE OF THE 'EOB' MACRO IN CONTROLLING     22
 30            THE APPEARANCE OF THE LISTING OF THE "OUTPUT" STREAM IS SELF-EVIDENT.       23
 30                                                                                         24
 32    INSERT BLANKS - 12     34                                                            25
 32                                                                                         26
 35    THE MACRO BODY OF  5 BLANKS  CAN BE EXHIBITED USING THE PRIMITIVE MACRO "VAL".       27
 35                                                                                         28
 51    ..     .. THE "5 BLANKS" APPEAR BETWEEN THE SETS OF TWO DOTS.                       29
 51                                                                                         30
 52    "VLMC"  -  REPRESENTS THE "VALUE OF THE MACRO".                                     31
 52                                                                                         32
 52    MACRO DEFINITION                                                                    33
 52      MACRO NAME - VLMC                                                                 34
 52      MACRO BODY - $2EOB;MACRO DEFINITION$EOB;  MACRO NAME - ¬1$EOB;  MACRO BODY - $    35
 65    VAL,¬1;                                                                             36
 65                                                                                        37
 65    MACRO DEFINITION                                                                    38
 65      MACRO NAME - TH1                                                                  39
 65      MACRO BODY - THERESA                                                              40
 65                                                                                        41
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
 65     MACRO DEFINITION                                                                        42
 65       MACRO NAME - TH2                                                                      43
 65       MACRO BODY - HELLNER                                                                  44
 65                                                                                             45
 65     MACRO DEFINITION                                                                        46
 65       MACRO NAME - TH3                                                                      47
 65       MACRO BODY - $TH1;$1 BLANK;$TH2;                                                      48
 65                                                                                             49
 65     MACRO DEFINITION                                                                        50
 65       MACRO NAME - TH4                                                                      51
 66       MACRO BODY - $TH1; $TH2;                                                              52
 66                                                                                             53
 66     MACRO DEFINITION                                                                        54
 66       MACRO NAME - TH5                                                                      55
 66       MACRO BODY - $2EOB;$TH3;                                                              56
 66                                                                                             57
 66     MACRO DEFINITION                                                                        58
 66       MACRO NAME - TH6                                                                      59
 66       MACRO BODY - $2EOB;$TH4;                                                              60
 66                                                                                             61
 66     MACRO DEFINITION                                                                        62
 66       MACRO NAME - TH7                                                                      63
 66       MACRO BODY - THERESA HELLNER                                                          64
 66                                                                                             65
 66     MACRO DEFINITION                                                                        66
 66       MACRO NAME - TH8                                                                      67
 68       MACRO BODY - THERESA HELLNER                                                          68
 68                                                                                             69
 72     MACRO-CALLS TO "TH1","1 BLANK", AND "TH2" GIVE THERESA HELLNER                          70
 72                                                                                             71
 72     MACRO DEFINITION                                                                        72
 72       MACRO NAME - SEETH                                                                    73
 74       MACRO BODY - $EOB;A CALL TO  ¬1  GIVES ..$VAL,¬1;..                                   74
 74                                                                                             75
 74                                                                                             76
 76     A CALL TO  SEETH  GIVES ..$EOB;A CALL TO  ¬1  GIVES ..$VAL,¬1;....                      77
 77                                                                                             78
 77     A CALL TO  TH1  GIVES ..THERESA..                                                       79
 77     A CALL TO  TH2  GIVES ..HELLNER..                                                       80
 77     A CALL TO  TH3  GIVES ..$TH1;$1 BLANK;$TH2;..                                           81
 78     A CALL TO  TH4  GIVES ..$TH1; $TH2;..                                                   82

       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
 78    A CALL TO  TH5  GIVES ..$2EOB;$TH3;..                                                   83
 78    A CALL TO  TH6  GIVES ..$2EOB;$TH4;..                                                   84
 78    A CALL TO  TH7  GIVES ..THERESA HELLNER..                                               85
 80    A CALL TO  TH8  GIVES ..THERESA HELLNER..                                               86
 80                                                                                            87
 87    1ST TEMPORARY "A"     ...XUX - XUX                                                      88
 87                                                                                            89
 89    2ND TEMPORARY "A"     ...XUX - XUX                                                      90
 89                                                                                            91
 91    -ABC- MACRO                                                                             92
 91    ABXYCPQAB                                                                               93
 91      AB99C88AB                                                                             94
 93        ABSSCTTAB                                                                         95
 93                                                                                          96
 93    -A- MACRO                                                                             97
 93    ACA                                                                                   98
 93      AACAA                                                                               99
 93        AACAA                                                                            100
 95          AXDXA                                                                          101
 95                                                                                         102
 95    -B- MACRO                                                                            103
 95    BAXDXAB                                                                              104
 95      BAXGGGXAB                                                                          105
 97        BAXAHAXAB                                                                        106
 97                                                                                         107
 99    -APA- MACRO                                                                          108
 99    PYYP                                                                                 109
 99      PYYP                                                                               110
101        PAZAAZAP                                                                        111
101                                                                                        112
103    3RD TEMPORARY "A"     ...XUX - XUX                                                  113
104    Q$A,C;R                                                                            114
105      A$A,X;A                                                                          115
106        BAX$A,X;XAB                                                                    116
107          Q$R;                                                                         117
109            Q<$A,C;>R                                                                  118
109                                                                                       119
111    4TH TEMPORARY "A"     ...XUX - XUX                                                 120
111                                                                                       121
115    -SUC- MACRO                                                                        122
115                                                                                       123
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
115    MACRO DEFINITION                                                                        124
115      MACRO NAME - SUC                                                                      125
117      MACRO BODY - $1,2,3,4,5,6,7,8,9,10,$DEF,1,<¬>¬1;;                                     126
117                                                                                            127
118    USE -SUC- MACRO                                                                         128
120    1 2 3 4 5 6 7 8 9 10                                                                    129
121    1                                                                                       130
121    7                                                                                       131
121    10                                                                                      132
122                                                                                            133
125    1 2 3 4 5 6 7 8 9 10                                                                    134
125                                                                                            135
129    -SUCCESSOR- MACRO                                                                       136
129                                                                                            137
129    MACRO DEFINITION                                                                        138
129      MACRO NAME - SUCCESSOR                                                                139
131      MACRO BODY - $¬2,$DEF,¬2,¬1<,$SUC,>¬2<;>;$DEF,9,<$SUC,>¬1<;,0>;;                      140
132                                                                                            141
132    CALLS TO -SUCCESSOR- MACRO                                                              142
133                                                                                            143
135    2,2  =  2,3     6,9  =  7,0                                                            144
135                                                                                            145
137    5TH TEMPORARY "A"     ...XUX - XUX                                                     146
138    2,2  =  2,3     6,9  =  7,0                                                            147
139    1,9  =  2,0     5,0  =  5,1                                                            148
141    0,0  =  0,1     8,8  =  8,9                                                            149
141                                                                                            150
149    DEFINE  +  -  *  QUOT  REM  MACROS                                                     151
149                                                                                            152
149    MACRO DEFINITION                                                                        153
149      MACRO NAME - +                                                                        154
149      MACRO BODY - $DEC,$BAR,+,$BIN,¬1;,$BIN,¬2;;;                                          155
149                                                                                            156
149    MACRO DEFINITION                                                                        157
149      MACRO NAME - -                                                                        158
149      MACRO BODY - $DEC,$BAR,-,$BIN,¬1;,$BIN,¬2;;;                                          159
149                                                                                            160
149    MACRO DEFINITION                                                                        161
149      MACRO NAME - *                                                                        162
149      MACRO BODY - $DEC,$BAR,*,$BIN,¬1;,$BIN,¬2;;;                                          163
149                                                                                            164

       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
149    MACRO DEFINITION                                                                        165
149      MACRO NAME - QUOT                                                                     166
149      MACRO BODY - $DEC,$BAR,/,$BIN,¬1;,$BIN,¬2;;;                                          167
149                                                                                            168
149    MACRO DEFINITION                                                                        169
149      MACRO NAME - REM                                                                      170
151      MACRO BODY - $DEC,$BAR,R,$BIN,¬1;,$BIN,¬2;;;                                          171
151                                                                                            172
153    USE  +  -  *  QUOT  REM  MACROS                                                         173
154                                                                                            174
155    2 + 3 = 5                                                                               175
156    17 - 19 = -2                                                                            176
157    (-13) * (-13) = 169                                                                     177
158    (-29)/6 = -4                                                                            178
160    REM( (-29)/6 ) = -5                                                                     179
160                                                                                            180
161    -1234 + -191 = -1425                                                                    181
161                                                                                            182
162    -1234 - -191 = -1043                                                                    183
162                                                                                            184
163    -1234 * -191 = 235694                                                                   185
165    REM( (-9876)/-17 ) = -16                                                                186
165                                                                                            187
165    MACRO DEFINITION                                                                        188
165      MACRO NAME - SEECALC                                                                  189
167      MACRO BODY - $EOB;¬2 ¬1 ¬3 = $¬1,¬2,¬3;                                               190
167                                                                                            191
168                                                                                            192
168    1 + 2 = 3                                                                               193
169    -7 * -19 = 133                                                                          194
169    $$99 * -9999 = -99980001                                                                195
171    -191 - -1234 = 1043                                                                     196
171                                                                                            197
173    6TH TEMPORARY "A"     ...XUX - XUX                                                      198
173                                                                                            199
182    FACTORIAL DEFINITION                                                                    200
182                                                                                            201
182    MACRO DEFINITION                                                                        202
182      MACRO NAME - FACT                                                                     203
182      MACRO BODY - $$DEC,$BIN,¬1;;,$DEC,$BIN,¬1;;,$DEF,$DEC,$BIN,¬1;;,<$¬1,¬1,$DEF,¬       204
184    1,<$*,$FACT,$-,¬1,1;;,¬1;>;$DEF,1,<1>;;>;$DEF,0,<1>;;                                   205
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
                                                                                              206
184                                                                                           207
185                                    -                                                      208
185                                                                                           209
186    FACTORIALS                                                                             210
187    0 - 1                                                                                  211
188    1 - 1                                                                                  212
189    2 - 2                                                                                  213
190    3 - 6                                                                                  214
191    4 - 24                                                                                 215
192    5 - 120                                                                                216
193    6 - 720                                                                                217
194    7 - 5040                                                                               218
195    8 - 40320                                                                              219
197    9 - 362880                                                                             220
197                                                                                           221
202    7TH TEMPORARY "A"     ...XUX - XUX                                                     222
202                                                                                           223
202    MACRO DEFINITION                                                                       224
202      MACRO NAME - CNVD1TOB2                                                               225
202      MACRO BODY - $UPDATE,¬2,$BIN,$VAL,¬1;;;                                              226
202                                                                                           227
202    MACRO DEFINITION                                                                       228
202      MACRO NAME - CNVB1TOD2                                                               229
208      MACRO BODY - $UPDATE,¬2,$DEC,$VAL,¬1;;;                                              230
208                                                                                           231
208    MACRO DEFINITION                                                                       232
208      MACRO NAME - SEEDEC                                                                  233
210      MACRO BODY - ¬1 = $VAL,¬1;                                                           234
210                                                                                           235
213    0D = +0000000000      1D = +0000000001      -1D = -0000000001                          236
213                                                                                           237
213    MACRO DEFINITION                                                                       238
213      MACRO NAME - DEFBIN                                                                  239
217      MACRO BODY - $DEF,¬1,$BIN,¬2;;                                                       240
217                                                                                           241
217    MACRO DEFINITION                                                                       242
217      MACRO NAME - SEEBINVAL                                                               243
219      MACRO BODY - $DEC,$VAL,¬1;;                                                          244
219                                                                                           245
221    0B = 0      1B = 1      -1B = -1                                                       246
221
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
```

```
IN      1   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80      OUT
       +....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
                                                                                              247
221    MACRO DEFINITION                                                                       248
221      MACRO NAME - TESTFACT                                                                249
223      MACRO BODY - $EOB;   ¬1  FACTORIAL IS  $FACT,¬1;                                     250
223                                                                                           251
225    FACTORIALS                                                                             252
225       0  FACTORIAL IS  1                                                                  253
225       1  FACTORIAL IS  1                                                                  254
227       9  FACTORIAL IS  362880                                                             255
227                                                                                           256
227    MACRO DEFINITION                                                                       257
227      MACRO NAME - DEFDECO                                                                 258
238      MACRO BODY - $DEF,¬1,+0000000000;                                                    259
238                                                                                           260
249          ZZA = -37      ZZB = -39      ZZC = 17      ZZD = 19                           261
249                                                                                           262
251    YYA = -37      YYB = -39      YYC = 17      YYD = 19                                   263
251                                                                                           264
253    .....  THE END  .....                                                                 265
253
```

IMPLEMENTATION OF A SINGLE STACK SYMBOL STREAM PROCESSOR

by

THERESA ADEL HELLNER

B.A. in Ed., Arizona State University, 1968

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Statistics and Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1970

The General Purpose Macrogenerator (GPM) was implemented on the IBM 360/50 at Kansas State University. Although designed as a macrogenerator, GPM is actually a symbol stream processor capable of function evaluation and function definition. Function evaluation is the result of a macro-call which may include parameters. Unlike most macrogenerators, GPM evaluates the parameters of a macro-call before they are substituted for the formal parameters in the macro definition allowing for the nesting of macro-calls. One unusual feature of GPM is that function or macro definition is the result of a macro-call to a primitive or machine macro.

GPM evaluates and defines macros using a single pushdown stack, a few primitive (control) characters, and a few primitive (machine) macros. Basically, processing consists of copying the symbols in the input stream directly to the output stream performing special action when a macro-call is encountered during the scan.

The present implementation used the OS/360 Macro-Assembler Language and required several modifications in the program. The results of testing confirmed that this version of GPM does satisfy its specifications, though execution speed is slow. Testing also indicated that several features should be added to the program if GPM were to be more widely used.