25
2-1-63

The RAND Corporation
1700 MAIN ST. · SANTA MONICA · CALIFORNIA

29 January 1963 L-2175

Mr. C. L. Baker
The RAND Corporation
4921 Auburn Avenue
Bethesda, Maryland

Dear Chuck:

Paul asked me to send you the enclosed examples. The copy of the letter to Bert will provide context. Would appreciate your comments, and the materials you used on your high school class.

Sincerely,

Hugh

Hugh S. Kelly
Computer Sciences Department

HSK/mac

Enclosures: (1) IPL Examples and Problems
(1) L-1829

24 January 1963 L-1829

Dr. Bert Green
Department of Psychology
Carnegie Institute of Technology
Pittsburgh 13, Pennsylvania

Dear Bert:

I am enclosing a rough draft of examples and some problems based on them, mainly to get your reaction to the proposed layout. At Paul's suggestion, Steff and I plan to combine the exercises and the logic theorist in one MM...Part I for the exercised, Part II for LT. There has been some indication that Prentice-Hall might be willing to publish it.

I'm not completely sold on the enclosed layout for Part I, but let me argue in favor of it first.

1. Putting the examples all in Section 1 with a brief comment for each makes Section 1 independent of the problems. One can plow through this whole section rather quickly and get a good picture of IPL.

2. The examples can be arranged to correspond closely to the introduction of concepts in Part I of the manual, so the instructor can assign reading from the manual corresponding examples.

3. Problem statements are cleaner, shorter and more precise, since they need only mention the name of the structures to be manipulated. Several problems can be based on the same structure or one problem can mention several structures without repeating the explanation and presentation of the data.

4. A problem statement can point to an example in Section 1 as a hint or model to guide the student.

The big disadvantage of this scheme is that in working a problem, the student has to refer back to the examples to get the data for the problem. I find this very annoying and a bit confusing, so say the word and I'll combine Section 1 and 2.

Here are some other questions to consider:

1. How many problems and examples do we want? fifty of each? one hundred of each?

2. Should each example refer to the appropriate pages in Part I of the manual? problems too?

3. Should we include the brief definition of the J's in this RM? If so, should they refer to the page in the manual where the full definition is given, or should we collect and include the full definitions?

4. Where would you like to see printline-readline, save for restart, restart, snapshot debugging introduced? They aren't covered in Part I of the manual.

5. I think the enclosed problems 5 through 12 are already pretty tough. How about some very simple examples and problems, not necessarily related to LT? I have many of these, courtesy of Ed Feigenbaum and Bob Hsu. Drills on the use of Q and P codes for example.

6. I object to providing drivers for the student's solutions. They can and should learn to make up their own decks early in the game. (See example 3 and 4.) It would be nice to provide the instructor with a routine that would accept complete decks from the student and check the solution by running them, but I don't think there is enough money in the pot to do this. Therefore, I propose that we include a couple of drivers as examples and then let the instructors decide whether he will run the student's codes or they should code drivers and run their own.

7. I think the examples should show alternative IPL representations of external objects, that problems should be stated in terms of the external objects rather than in terms of lists and symbols, and a given problem should have to be solved for a couple of different representations. If you agree, do you prefer separate problem statements (e.g., problem 10 and 11) or just one, as in problem 12?

8. I have access to GPS, Baseball, Map, LT, HC, etc., and plan to use data structures from these where possible. LT doesn't have enough variety by itself. O.K.?

9. Is the commentary with the enclosed examples appropriate, too long, too short?

Guess that's about it. Will await your reactions and suggestions.

Sincerely,

Hugh S. Kelly
Computer Sciences Department

HSK/mac

Enclosures: (1) IPL Examples and Problems.

cc: P. Armer
A. Newell
H. Simon
F. Tonge
K. Uncapher

SECTION I EXAMPLES

This section contains data structures and routines drawn from heuristic programs coded in IPL-V, as well as some simple tutorial examples. The examples are accompanied by commentary which explains the format, the context, the interpretation, or interesting IPL techniques. The arrangement of the examples roughly parallels the introduction of important concepts in Part I of the IPL-V Programming Manual, with the intent to demonstrate the concept with further examples in a real context. The student should find this section a useful supplement while studying Part I of the Manual.

Section I. EXAMPLE ROUTINES AND DATA STRUCTURES

1. Logic expressions are input to the Logic Theorist in a convenient external form and then are converted to an internal form for convenient manipulation. The external form is a string of characters. Examples 1A and 1B are strings that represent logic expressions.

1A. (AI(A ∨ B)) Read "A implies (A or B)".

1B. ((-P ∨ (Q ∨ R))IS)

2. L1 is a list which represents string 1A by using regional symbols to stand for the characters of the string. Inserting, deleting, or rearranging characters of the string would now correspond to manipulating symbols on list L1.

COMMENTS	TYPE	NAME	PQ	SYMB	LINK
Data Header	5		1		
(AI(A ∨ B))		L1		0	
				(0	
				A0	
				I0	
				(0	
				A0	
				∨0	
				B0	
				)0	
				)0	0

The data header is not part of the list L1 but serves to indicate to the IPL loader that data will follow, rather than routines. Note that the first character of the string should be written on the line below the line which names the list. The zero in the LINK field terminates the list.

3. R1 is a routine which will print L1 as a list.

COMMENTS	TYPE	NAME	PQ	SYMB	LINK
Routine Header	5				
Input the symbol "L1" to H0.		R1	10	L1	
Execute Process J151.			00	J151	0

Again note that the first card, with TYPE = 5, Q = blank, is not part of R1 but indicates to the IPL loader that R1 is a routine rather than data.

4. To actually print the list L1, we would need to input the following deck of cards to an IPL-V computer.

COMMENTS	TYPE	NAME	PQ	SYMB	LINK
Type-9 card	9				
	1				
Type-2 cards define regions.	2	A			1
The B-Region consists of	2	B			1
one cell, named "B0" or just "B".	2	I			1
	2	L			2
The R-region has 2 cells,	2	R			2
named "R0" and "R1".	2	∨			1
	2	(			1
	2	)			1
Type-1 cards are for comments.	1				
	1				
Type-5, Q=1 says "DATA FOLLOWS."	5		1		
L1---(AI(A ∨ B))		L1		0	
				(	
				A	
				I	
				(	
				A	
				∨	
				B	
				)	
				)	0
Type-5, Q=blank says "ROUTINES	5				
FOLLOW."	1				
	1				
R1 prints the list L1.	1				
	1				
Q=3 says "TRACE THIS ROUTINE."		R1	13	L1	
				J151	0
P or Q = 0 can be left blank.					
Type-5, SYMB=R1 says execute R1	5			R1	

See pages 214-223 of the IPL Manual for a detailed discussion of the deck setup for running a program. Page 221 is a convenient summary of the requirements. The important things to note at this point are that you

must 1) define the regional symbols you use by type-2 cards, and 2) precede your blocks of <u>data</u> with a type-5 card, Q = 1 and your <u>routines</u> with a type-5, Q = 0 or blank.

5. L2 is a list structure which represents string 1A as a list of alphanumeric data terms, one data term for each character in the string.

COMMENTS		TYPE	NAME	PQ	SYMB	LINK
Data Header		5		01		
(AI(A ∨ B)) in data terms.			L2		0	
	(				9-1	
	A				9-2	
	I				9-3	
	(				9-4	
	A				9-5	
	∨				9-6	
	B				9-7	
	)				9-8	
	)				9-9	0
			9-1	21	(	
			9-2	21	A	
			9-3	21	I	
			9-4	21	(	
			9-5	21	A	
			9-6	21	∨	
			9-7	21	B	
			9-8	21	)	
			9-9	21	)	

Note that the characters in the data terms of L2 are <u>not</u> regional symbols, and hence you would only have

to define the L-region in order to load L2. This representation of the string 1A takes 2 cells per character.

SECTION II PROBLEMS

The following problems are based on and refer to the data structures and routines in Section I. Solutions to all problems are given in Section III. The student is urged to attempt hiw own solution and then compare it with the solution provided in Section III.

1. Code a list named D1 which represents the string in Example 1B as a list of regional symbols. (Use Example 2 as a model.)

2. Code a list structure named D2 which represents the string in Example 1B as a list of local data terms. (Use Example 5 as a model.)

3. Code R2--"Print L2 as a List Structure." (Use Example 3 as a model.)

4. Produce all the coding necessary to run R2 of Problem 3. (Use Example 4 as a model.)

5. Code P1--"Reverse the order of the characters in string (0)." Assume that the string named in H0 is like L1, Example 2.

6. If we assume that H0 names a string like L2 in Example 5, will the P1 which you coded in problem 05 still serve to reverse the order of the characters in the string? If not, code P2 to do so.

7. Code P3--"Delete all occurrences of character (0) from string (1). Set H5 minus if the character did not occur in the string, otherwise set H5 plus." P3 assumes that (0) is a regional symbol and (1) is a list like L1 of Example 2.

8. Code the routine P4, having the same definition as P3 in Problem 7, except that P4 assumes (0) names an alphanumeric data term and (1) is like L2 in Example 5.

9. Code P5 and P6--"Locate character (0) in string (1). Set H5 plus and output the location if the character is found. Otherwise set H5 minus and output the location of the last character in the string." P5 assumes (0) is a regional symbol and (1) is like L1. P6 assumes (0) names an alphanumeric data term and (1) is like L2. (In both P5 and P6, the "location" that is wanted is the name of one of the cells on the main list of L1 or L2, not the name of the regional symbol nor of the data term.)

10. Code P7--"Delete from string (1) all occurrences of the characters in string (0). H5 is not changed by P7." P7 assumes string (0) and string (1) are like L1 except a given character appears only once on (0) and may occur several times on string (1).

11. Code P8--P8 has the same definition as P7 but assumes strings (0) and (1) have a format like L2. Thus, to delete a character in this case means to remove the name of the local data term from the main list of L2 and then to erase that data term.

12. Code P9 and P10--"Create the list of operators that are common to string (0) and string (1), disregarding order. Set H5 minus and create no list if there are no common operators. Otherwise set H5 plus and output the list of common operators." P9 assumes the strings to be like L1, while P10 assumes them to be like L2.

Both assume that the characters "I," "∨," "*," and "-" are the only operators. An operator should appear on the output list only once, regardless of how often it appears in common on the input strings.

SECTION III SOLUTIONS TO PROBLEMS

The solutions presented here are not to be construed as the solution, but an attempt has been made to demonstrate standard IPL techniques.

B.A. 214
Spring 1961
F. M. Tonge

Notes on the Internal Representation of Data in the Logic Theorist (IPL-V version)

Total Expressions.

Logic expressions are represented as describable list structures, the sublists of which are non-describable. On sublists, representing subexpressions, the connective of the subexpression is the symbol in the head of the sublist, the left subexpression is the first symbol on the sublist, and the right subexpression (if any) is the second symbol on the sublist.

For example, the internal representation of

(PVQ)I-P

would be:

```
....  0
      9-1  0
9-1   I
      9-2
      9-3  0
9-2   V
      P
      Q    0
9-3   -
      P    0
```

The subexpression corresponding to the entire logic expression (here, 9-1) is called the main expression or main segment. All sublist structures are called segments.

Copy for Substitution.

The match process (M20) takes as inputs total expressions. Often, however, we wish to match subexpressions (as, for example, the right sides of two expressions.) The subexpressions must then be set up as separate total expressions. The match process makes use of substitution (M21) in the expressions to keep track of correspondences made between free variables and constants. However, the substitution should be made throughout the entire original total expressions from which the expressions for matching were derived.

We handle this total substitution by using the segment on which we desire to match (rather than a copy of that segment) as the main expression of a derived total expression, associating with the new expression the original total expression from which it was derived. Substitution into the original total expression thus is also substitution into the derived expression. (We must be careful not to erase the derived total expression as a list structure, since its main segment, which may be a local sublist, in fact belongs to the original total expression.)

Working lists.

There are four major lists on which logic expressions in various states of processing are kept.

The problem list (L0) is a linear list of the problems to be proven during the run.

The theorem list (L1) is a linear list of true logic expressions (at first just the axioms.)

The untried subproblems list (L10) is a list of pairs. The first of each pair is an integer data term. The second of each pair is a list of those untried subproblems having that integer as their "number of levels." The data terms are arranged in ascending order. Thus, the "simplest" untried subproblem can be found on the first sublist.

The tried subproblems list (L11) is also a list of pairs, the first of each pair being an integer data term, "number of levels." However, the associated list is also a list of pairs, the first member of each pair being an integer data term, "number of distinct variables." The associated lists are again lists of pairs, the first of each pair being an integer data term, "number of variable places." The associated list is a list of all tried subproblems (expressions) having the indicated "number of levels," "number of distinct variables," and "number of variable places." Thus, to determine if a given subproblem has already been tried, it is only necessary to compare it with tried subproblems having the same description.

B. A. 214
Spring, 1961
F. M. Tonge

Major Errata for IPL-V Manual, Section I (RAND P-1897) (May 16, 1960 version)

pg. 26, line 1: '9-1': the name of the next cell on the list.

pg. 29, line 7:
```
      9-1  30 W0          Pop up W0
```

pg. 42: all regions should be of size 2

pg. 52, line 11:
```
          32 H0  9-2     Pop it up, ...
```

pg. 52: an end of routine R3, add:
```
          12 H0
             9-10
          70 9-11  J8
```

pg. 73, line 4: 9-10 works in the same context as J77; ...

pg. 82, line -4: the cells of each sublist will be generated immediately after

pg. 83: code for R10 is wrong

pg. 95, line 15: Fig. 10.

pg. 125, line 17:
```
          ..............  11 Y1
```

IDENTIFICATION

RS IPLV -- IPL-V Interpretive System
C. L. Baker, H. S. Kelly
The RAND Corporation, Santa Monica, California
November 25, 1960

PURPOSE

To interpret and execute programs written in the IPL-V language, as described in AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V, and INFORMATION PROCESSING LANGUAGE V MANUAL, Sections I and II, The RAND Corporation papers P-1929, P-1897, and P-1918, 1960.

RESTRICTIONS

The program is in the form of a closed subroutine, designed for compatability with the MockDonald Operating System for the 709/7090. The calling sequence is in the same format as in the BCS routine within MD. The individual installation using this system may incorporate IPL as a file on the MD SYSTAP (as has been done at RAND), or execute the IPL program in phase I of the system, or integrate with any other system with a minimum of effort. The program may be run on any 709/7090 of at least 8K storage capacity and having at least three tape units. No on-line card reading or printing capability is provided.

METHOD

The IPL system is read into core, at which time it saves the rest of core (which would contain the operating system in use) on tape, then reads the IPL-V program into core and assembles it. Control is then passed to the IPL-V interpreter. Error detection and tracing capability is provided, and a Post-Mortem in IPL-V format is printed at the conclusion of execution. Core is then restored from tape, and control returned to the calling routine. Execution speed is approximately 210,000 IPL-V instructions per minute.

USE

See the papers mentioned above for a basic description of the system and language; deviations from this description are described in the attached write-up. Details of the calling sequence for the routine, etc., are also included.

NOTES

While this distribution is to be considered a field-test trial of the system, it is in use at The RAND Corporation for a number of complex problems, including the proof of logic

theorems using the General Problem Solver coded in IPL-V. This program contains more than 6,000 IPL-V instructions and takes more than 500,000 interpretation cycles for a complete run. Numerous other large problems have also been interpreted and executed. The inclusion of IPL-V in the MockDonald System at RAND permits IPL-V jobs to be stacked with other jobs and executed with all of core available to the IPL-V programmer.

Any questions regarding the operation or correctness of the IPL-V system should be directed to:

C. L. Baker
The RAND Corporation
1700 Main Street
Santa Monica, California.

Questions concerning the incorporation of the IPL-V system into the MockDonald Monitor should be directed to:

G. E. Bryan
The RAND Corporation
1700 Main Street
Santa Monica, California.

12-2-60
FMT

CLB

LINE READ PRIMITIVES

The line read primitives are assembled into 25500_{10}. They may be loaded preceded by a Type 5, P=3, card. No space need be reserved with a Type 3, Q=2, card. The packet of line read binaries should immediately follow the Type 2 and 3 cards on loading. Until they are in memory, references to W30 will not be accepted by the loader (and the initial random number control will be violated).

Line read is controlled by the following symbols:

1W24 names the current line. (The symbol in cell 1W24 is the address of the first of a block of consecutive cells forming the line, one column per cell. The first column is stored in the cell after 2W24, that first cell being used to hold the spacing control in line printing. The link in cell 1W24 is the number of columns in the line, and thus one less than the number of consecutive cells in the block. Until the Type 3, Q=1, facility for automatically setting up print (and read) lines is provided, the programmer must set these control words up himself, using consecutive space from the regions or primitive block.)

1W25 is an integer data term giving the left column of the current input field.

1W30 is an integer data term giving the size (number of columns) of the current input field.

The line read primitives are:

*J180 READ LINE. The next record on the input tape is read into line 1W24. (The record is assumed to be BCD.) Column 1 of the record is read into column 1 of the input line, and so forth. H5 is set +. If no record can be read (end of file condition), the line is not changed and H5 is set -.

*J181 INPUT LINE SYMBOL. The IPL symbol in the field starting in column 1W25 and of size 1W30, in line 1W24, is input to H0 and H5 is set +. The symbol is regional if the first column is a regional character; otherwise it is absolute internal. In inputting the symbol all non-numerical characters except in the first column are ignored. If the field is entirely blank, nothing is input to H0 and H5 is set -. In either case, 1W25 is incremented by the amount 1W30.

*J182 INPUT LINE DATA TERM (0). The field as specified in J181 is taken as the value of a data term, and input data term (0) is set to that value and left as output (0). H5 is set +. (At present only integer and BCD data terms are recognized; any input (0) not integer is taken as BCD. In composing BCD data terms, characters are entered from the left and the full data term completed with blanks on the right if necessary. In composing integer data terms, non-numerical characters are treated as blanks and ignored.) If the field is all blanks, (0) is cleared (to blanks for BCD, to zero for integer) and H5 is set -. 1W25 is incremented by the amount 1W30.

*J183 SET (0) TO NEXT BLANK. (0) is taken as the name of an integer data term. Line 1W24 is scanned, left to right, starting with column 1W25+1. A "1" is added to (0) for each column scanned, including that in which the scanned-for character (blank for J183) is found. (0) is left as output (0). H5 is set + if the character is found, and - if it is not.

(Thus, if (0)=1W25, after scanning it will indicate the column holding the scanned-for character. If (0)=0, after scanning it will be the size of a field delimited on the right by the next occurrence of the scanned-for character.)

*J184 SET (0) TO NEXT NON-BLANK. Same as J183, except scans for any non-blank character.

*J185 SET (1) TO NEXT OCCURRENCE OF CHARACTER (0). Same as J183, except scans for character (0), tallying into input (1). Input (1) is left as output (0). If input (0) is regional, its region is the character scanned for; if internal, its last digit is the character scanned for.

*J186 INPUT LINE CHARACTER. The character in column 1W25 of line 1W24 is input to H0 and H5 is set +. If the character is a blank, there is no input and H5 is set -. In either case, 1W25 is not advanced.

*J189 TRANSFER FIELD. The field in line 1W24, starting in column 1W25 and of size 1W30 is transferred to line (0), starting in column 1W21, and H5 is set +. If the entire field cannot be transferred (line (0) too short), as much is transferred as can be, and H5 is set -. In either case, 1W25 is set to the last column transferred + 1.

709/7090 IPL-V

Program Description

This write-up is in two parts; the first describing the nature of the IPL-V system as a subroutine, the second including deviations from and additions to the basic IPL-V manual.

PART I

Calling Sequence:

```
CS - 1 = LDI CNTRL
CS     = TSX AO,4
CS + 1 = ...
CS + 2 = PZE SYSPIT,,SYSMIT
CS + 3 = PZE SYSPOT,,,...
CS + 4 = PZE SYSES1,,SYSES2
CS + 5 = ...
CS + 6 = PZE ...,,SYSERR
                .
                .
                .
                .
CS + 13= PZE SYSORG,,...
                .
                .
                .
                .
CS + 29= Return
                .
                .
                .
                .
CS + 37= PZE SYSTSL,,...
```

The above calling sequence is that used within the MockDonald Operating System; hence its strange format. Words or parts of words indicated by "..." are not significant.

Control bits:

Bits 31-35 of CNTRL are placed in the Sense Indicator Register for conveying information to IPL-V as to the nature of the run being made. Immediately below is given the meaning of these bits; their interpretation is explained in Part 2.

Bit 35	0	LOAD
" "	1	RELOAD
Bit 34	0	NOTAPE
" "	1	TAPE

Bit 33	0	LIST
" "	1	NOLIST
Bit 32	0	NTRACE
"	1	TRACE
Bit 31	0	NODUMP
" "	1	DUMP

The program which includes the calling sequence may interpret a control card similar to that used to call SCAT, LOAD, etc., and set the bits accordingly.

Calling Sequence Information:

SYSPIT, SYSMIT, SYSMOT, SYSES1, SYSES2 are the locations of tape control words. These words must contain the address of the physical unit associated with the symbolic tape. (E.g., if SYSPIT is on physical unit B2, the control word would have an octal address of 02202.) Only the first three tapes are necessary for using the system. If the address is zero, the unit is interpreted as being unavailable.

SYSTSL is the location of the Tape Status list for reserved tapes. It is arranged such that

```
AXT  CHANNEL,2     (1 for Ch. A, 2 for Ch. B, etc.)
AXT  UNIT,1        (3 for unit 3, etc.)
CLA* SYSTSL,2
```

will put the address of the corresponding tape control word into the decrement of the accumulator. Tapes SYSAR1,2,3, SYSBR1,2, and 3 are interrogated. Again, a zero address is interpreted as meaning that the corresponding unit is unavailable.

SYSERR is the location to which the operator is to transfer in case of program error, loop, too much time, etc. Both SYSERR and SYSERR-2 are replaced with a transfer to the recovery routine for IPL-V; the contents of these cells is saved and restored. They must lie in the range $00001--00077_8$.

SYSORG is the name of a cell containing the location of the first word following the system in control. Memory up to location C(SYSORG) is left undisturbed by the execution of IPL-V (with the exception of cell zero.)

Operation of the IPL-V executive routine, AO:

1) All traps are disabled.
2) The tape control words are interpreted and a table of tape information is constructed.
3) Memory is saved on tape SYSMIT from locations 100_8-C(SYSORG).

4) If the RELOAD option is called, Memory is reloaded from SYSAR2.
5) The remaining Sense indicator bits are interpreted.
6) Control is passed to the IPL-V system interpreter.
7) At the conclusion of the IPL-V run (Program finished, error, or manual transfer to SYSTEM or SYSERR), Memory is restored from tape SYSMIT, and control returned to word 29 of the calling sequence.

Notes:

Although the system has been organized so as to work within the MockDonald Monitor System, it should be emphasized that it is in no way dependent upon any of the routines of that system. Only the tape control words of that system need be reproduced (in the format explained above) to gain complete independence of that system.

All input and output routines are contained within the IPL-V system itself. Data is read, in BCD, from tape SYSPIT; output is written in BCD on tape SYSPOT. Neither tape is rewound, nor is an end of file ever written on SYSPOT. Tape SYSMIT is used for saving core only; if it is not desired to retain this feature, routines A2 and A3 may be replaced by TRA 1,4. SYSAR1, 2 and 3 may be used by the IPL-V interpreted program; see following section for details. SYSES1,2, SYSBR1,2, and 3 are never used, and have been included in the tape set-up routine only in anticipation of possible future expansion of the IPL-V system.

Example:

Following is a complete calling sequence requiring no information from the MockDonald System:

```
START   LDI     CNTRL
        TSX     AO,4
        PZE
        PZE     SYSPIT,,SYSMIT
        PZE     SYSPOT
        PZE     SYSES1,,SYSES2
        PZE
        PZE     0,0,3           Manual error transfer to 3(8) or 1(8).
        PZE
        PZE
        PZE
        PZE
        PZE
        PZE
        PZE     SYSORG
        PZE
        PZE
        PZE
```

```
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            TRA     out
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE     SYSTSL
CNTRL       OCT     23          RELOAD,TAPE,LIST,NTRACE,DUMP  (Example)
SYSPIT      OCT     2201        Unit B1 for SYSPIT
SYSMIT      OCT     2202        Unit B2 for SYSMIT
SYSPOT      OCT     2203        Unit B3 for SYSPOT
SYSES1      PZE                 SYSES1 not used.
SYSES2      PZE                 SYSES2 not used.
SYSAR1      OCT     1201        Unit A1 for SYSAR1
SYSAR2      OCT     1202        Unit A2 for SYSAR2
SYSAR3      OCT     1203        Unit A3 for SYSAR3
SYSBR1      PZE                 SYSBR1 not used.
SYSBR2      PZE                 SYSBR2 not used.
SYSBR3      PZE                 SYSBR3 not used.
SYSORG      PZE     16384       Save lower half of core.
            PZE     CHNLB,1
            PZE     CHNLA,1
SYSTSL      PZE                 Tape Status List.
            PZE     0,,SYSAR3
            PZE     0,,SYSAR2
            PZE     0,,SYSAR1
CHNLA       PZE
            PZE     0,,SYSBR3
            PZE     0,,SYSBR2
            PZE     0,,SYSBR1
CHNLB       PZE
```

PART II

This write-up includes only deviations from and additions to the basic IPL-V manual, INFORMATION PROCESSING LANGUAGE V MANUAL, Section II, Programmers' Reference Manual, A. Newell, et al, The RAND Corporation paper P-1918, March 31, 1960. Familiarity with this manual is assumed, and references are to sections and page numbers in this manual.

INITIAL LOADING, Page 72

This version is not set up to run multiple IPL-V jobs. Each run must begin with the execution of the calling sequence for IPL-V. References to bits 31-35 refer to the setting of the Sense Indicator register on entry to IPL-V. (See Part 1.) The IPL-V program, set up as described on page 79, is assumed to be on tape SYSPIT, in BCD. This tape is not rewound prior to reading.

All output is off-line, on tape SYSPOT. This tape is never rewound, nor is an end-of-file written. This permits stacking of the IPL-V output with other jobs adhering to the same output conventions.

A TYPE 9 CARD (page 78) is treated as a type 1 comment card, as no multiple job provisions are included.

TYPE 2 CARDS do not have the full flexibility described on pages 73-74. Only the NAME and LINK fields are read; the first character of the NAME field must be the region symbol; the LINK field must contain the extent of the region. The first example in page 74 is the only acceptable format for type 2 cards.

A TYPE 3 CARD, if it appears, must precede any type 2 cards. It may be used to reserve a block of storage for binary IPL-V routines written by the programmer. The block will start at location 100_8, and the extent of the block will be as given in LINK. No other uses for type 3 cards are permitted. A block of storage has already been reserved for line printing; other print lines may be reserved by other means (see below under line printing).

TYPE 5 CARDS do not have the full flexibility as specified on pages 77-78. P must be zero or blank for standard IPL format, or 3 for absolute column binary format. Q is interpreted as described; however, there is a maximum of 150 local symbols and a maximum of 100 internal symbolics that may be used per routine or data list structure.

SYMB does not control the input unit, and is disregarded unless regional.

LINK character d is ignored; the only output from the assembly process is the listing on SYSPOT, as controlled by character c.

TYPE 6 AND 7 CARDS are treated as type 5 headers as no auxiliary storage capability is included.

NOTE: TYPE 1 CARDS should not appear in the middle of a program or data list following a program or data card with a blank SYMB or LINK. This will lead to an error which is not detected at assembly time. (The blank LINK or SYMB is treated as 0.)

ABSOLUTE BINARY INPUT. A type 5 card with P = 3 indicates that a block of absolute, column binary, cards follows. These must load into the space reserved by the single type 3 card at the beginning of the program. The last binary card must be a transfer card; the address is ignored, however. After the binary cards have been read, the symbols HO, JO, and WO are stored in the address parts of location 100_8, 101_8, and 102_8, respectively. This permits the programmer to refer to these regions conveniently when coding machine language programs.

ALTERNATIVE INPUT UNIT. Provision has been included for reading input from tape SYSAR1 in BCD format if desired. Sense indicator bit 34 controls this function. If bit 34 is TAPE when the program is first loaded, the IPL-V loader will initially read from tape SYSAR1 in BCD. The order and format of cards on this tape must be identical to that used for on-line reading of cards. When an end-of-file is encountered on this tape, the IPL-V loader will attempt to read from tape SYSPIT. Printing of the assembly of tape SYSAR1 may be suppressed by setting bit 33 to NOLIST; in this case only the header cards from SYSAR1 will print. If bit 33 is LIST, printing control is from character c of the header cards on the tape.

EXAMPLE. A short IPL-V program is to be loaded and executed. The deck should be set up exactly as described, and written on SYSPIT in BCD. Bit 34 should be NOTAPE. Execution of the IPL-V calling sequence will cause the program to be loaded and executed as described.

EXAMPLE. A long IPL-V program is to be loaded and executed. A BCD tape should be prepared off-line; the deck set-up should be exactly as described, except that the final type five card should not be written on the tape. The data is followed by an end-of-file on this tape. To load and execute, this tape is mounted on unit SYSAR1. The final type five card is loaded on SYSPIT, with bit 34 = TAPE, bit 35 = LOAD. Loading and execution will take place as described.

For subsequent runs, with routines to be changed or new data added, the same tape SYSAR1 may be mounted, and the corrected routines and data loaded on SYSPIT before the final type five card. Since the changed routines or data are read from SYSPIT after tape SYSAR1 has been read, the corrections will be made before interpretation begins.

INPROCESS LOADING, Page 80

The restrictions that apply to initial loading also apply to inprocess loading. Note that type 2 and type 3 cards may not be loaded. Regional symbols which were initially assigned by type 2 cards are still available, as unused regionals are NOT returned to available space. (This compensates somewhat for the lack of type 3 symbol reservation.)

SAVE FOR RESTART, Page 81

J166 should read SAVE ON TAPE SYSAR3 FOR RESTART. All of memory is written onto tape, and the program continues from that point. The tape is rewound after writing, so that multiple occurrences of J166 will cause the tape to be rewritten. Bit 31 should be DUMP if J166 is to be used.

J167 is not available, as no auxiliary storage is provided.

RELOADING. Reloading is from the tape written by J166. The tape should be mounted on tape SYSAR2; bit 35 = RELOAD is used to initiate the reload process. After reloading, control is returned to exactly the same place in the execution of J166 as when memory was saved.

EXAMPLE. A program is to be loaded, saved immediately for subsequent restart, and the program is then to be executed. On reloading, it is desired to make changes or corrections before execution.

Include a one-word routine in the program as follows:

NAME	SYMB	LINK
RO	J166	J165

Place two type 5 termination cards at the end of the deck; the first with a SYMB of RO, the second naming the start of the program; bit 35 = LOAD, bit 31 = DUMP. Operation is then as follows: The program is read in and assembled, and interpretation begins with instruction RO. J166 will save memory on SYSAR3, and the program will continue with the interpretation of the instruction, which links to J165. J165 will read more routines and data, including termination cards. Since the only card remaining is a termination card, interpretation will begin with the first instruction of the program.

To reload, place the corrected routines or data lists on SYSPIT, each preceded by the proper type 5 header card, and followed with the same two termination cards. The tape written on the first pass is mounted on tape SYSAR2. Bit 35 = RELOAD, bit 34 = NOTAPE, bit 31 = DUMP. Immediately after reload, interpretation will continue at RO; and J165 will read the corrections into

memory. The first termination card will repeat the save process on SYSAR3; the second will initiate the interpretation of the corrected program.

NOTE: The same procedure may be used when the initial program load is from SYSAR1 in BCD. Bit 34 should be NOTAPE on reload, however, so that J165 will be prepared to read corrections from SYSPIT rather than SYSAR1.

TAPE UNIT SUMMARY

TAPE SYSPIT. BCD Input tape (never rewound).

TAPE SYSPOT. BCD Output tape (never rewound; no end-of-files).

TAPE SYSAR1. Alternate BCD Input tape (under control of bit 34).

TAPE SYSAR2. Reload tape (under control of bit 35).

TAPE SYSAR3. Save for Restart tape (under control of bit 31 and J166).

OPERATION OF THE INTERPRETER

Interpretation begins within the system by interpreting the instruction

0 0 Xxx J7,

where Xxx is the routine named on the final type 5 termination card. Interpretation will continue until the J7 is linked to, at which time the program will print PROGRAM RAN TO COMPLETION on the output tape. Other terminations will occur if the programmer executes the routine J7, or if the interpreter or a primitive process detects any one of a number of errors. In case of such a termination, the reason will be printed on the output tape, followed by an IPL-V POST MORTEM. The post mortem consists of a print-out of the lists named on the list named by system cell W23. Initially this list names all the system cells, H0 through W29. The programmer may replace this list with another list, or add symbols to this list. Control is returned to 29,4 of the IPL-V calling sequence after execution.

ERROR TRAP, J170, Page 82

The error trap function is not provided; see above for error action.

SYSTEM CELLS, Page 35

Exceptions and additions are as follows:

H0 initially contains the symbol 0.

H2 will initially be about 26,000 cells for a 32K machine.

H3 is set to 1 by the initial loader.

H4 not used as no auxiliary storage is provided.

H5 is initially set to J4(+).

H6 is used by machine language recursive routines to hold index 4.

H7 is a data term which is set to the current level. It is set to 0 by the initial loader, and by J165 if cards are read.

H8 is the hide-out cell used by the generator processes.

H9, H10, and H11 are generator working cells.

W0....W9 initially contain the symbol 0.

W10 initially names a data term for random number generation.

W12....W15 initially contain J0.

W16....W20 are not used. See above.

W21 initially names a data term equal to 1.

W22 initially names a data term equal to 1.

W23 initially names a list of the system cells.

W24 initially names a 120 word internal buffer for line printing.

W25 initially names a data term equal to 1.

W26...W28 are not used. See above.

GENERATOR PROCESSES, Pages 40-43

The generator trace context and level context are not restored. Subprocesses may be marked to trace by use of J147..J149.

J102, Page 52

Not available. May be coded as described in part I of the manual, page 82, if desired.

AUXILIARY STORAGE PROCESSES, Page 52

Not available. J72 and J74 assume list structure is in memory.

ARITHMETIC PROCESSES, Pages 55-57

J110-J113 admit fixed and/or floating data terms only.

J115-J117 admit fixed and/or floating data terms only.

J125 admits fixed and/or floating data terms only.

J128 admits fixed and/or floating data terms only.

INPUT-OUTPUT CONVENTIONS, Pages 61-64

Input and output units are assigned as described above. Processes J140-J146 are not available.

MONITOR POINTS AND TRACING, Pages 64-67

No console signals are provided. Terminate for restart is not available. The program may be terminated by a manual TRA SYSERR or SYSTEM. Sense bit 32 = TRACE signals full trace. Trace speed is approximately 80 IPL-V instr/sec. Bit 32 = NTRACE gives normal tracing modes.

No provision is made for suppressing tracing.

On the printed trace line, if S or (0) is a local symbol, this is indicated by an asterisk following the symbol; e.g., 2765*. The designated symbol S is printed only if Q = 1 or 2.

PRINT PROCESSES, Pages 69-72

The output unit is always SYSPOT. When local symbols are printed (except in the case of those translated by J150) they are followed by an asterisk.

If a list structure with a local name is printed by J150, the converted name will be 9-0. Sublists of any list structure are always numbered consecutively starting with 9-1.

Data terms are always printed in the format given on page 71, with the exception of floating point numbers. The format for these is "-.123456-12"; plus signs are not printed. The field length is thus either 10 or 11 columns.

Cell W24 initially names an internal print buffer of 120 cells (one character per cell) length. Other print lines may be

assigned only by defining, via a type 2 card, a block of 120 consecutive regional cells. (E.G., 2 P0 120) P0 would then be a legitimate name for a print buffer line; the symbols P1....P119 would be unavailable for other use.

J158 and J159 leave 1W25 pointing to the first column to the left of the entered information.

Baker

20 October 1960 L-20983

Mr. Henry A. McCabe
Electronic Data Processing Department
Union Carbide Corporation
270 Park Avenue - 37th Floor
New York 17, New York

Dear Mr. McCabe:

An initial, field-test version of the IPL-V (Information Processing Language V) interpretive system for the 704 has been submitted to SDA for distribution to all interested SHARE members. While this is not a complete version of the system as specified in the Programmers' Reference Manual, it will suffice to run all but the largest IPL-V programs using a 32K machine, and may be used for small IPL-V programs on an 8K machine. It is more than adequate to indoctrinate the uninitiated in the use of this language.

We are also using, at RAND, a 7090 version operating within the MockDonald Monitor System (our version). Interested persons should first obtain the descriptive material from SDA, as this is essentially machine independent, and then contact me at RAND regarding the availability of the 7090 version.

Sincerely yours,

C. L. Baker

CLB:ahe

Baker

19 October 1960 L-20788

Mr. Donald C. Cashman
Applied Programming Department
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

Dear Don:

We are sending, under separate cover, the following material for distribution to SHARE members:

1. AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V (26 pages)
2. INFORMATION PROCESSING LANGUAGE V MANUAL, Section I, (138 pages)
3. INFORMATION PROCESSING LANGUAGE V MANUAL, Section II, (92 pages)
4. RS IPL5 Short Write-up (2 pages)
5. RS IPL5 Program Description (7 pages)
6. RS IPL5 Binary Deck (250 cards)
7. RS IPL5 Program Listing (114 pages)
8. RS IPL5 Remark Card Listing (28 pages)
9. SHARE Program Submittal Form (1 page)
10. RS IPL5 Index Cards (3 cards)
11. RS IPL5 Abstract Cards (4 cards)

This material constitutes the initial distribution of the 704 IPL5 interpretive system, and should be treated as a field-test version of the program.

Sincerely yours,

C. L. Baker (RS)

CLB:ahe

IDENTIFICATION

RS IPL5 -- IPL-V Interpretive System
C. L. Baker, H. S. Kelly
The RAND Corporation, Santa Monica, California
October 15, 1960

PURPOSE

To interpret and execute programs written in the IPL-V language, as described in AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V, and INFORMATION PROCESSING LANGUAGE V MANUAL, Sections I and II, The RAND Corporation papers P-1929, P-1897, and P-1918, 1960.

RESTRICTIONS

The program is in self-loading form, and may be run on any 704 of at least 8K storage capacity and having at least one tape unit. No on-line printing capability is provided. Other deviations of the program from the manuals are as described below.

METHOD

The IPL-V program is read into core and assembled, and control passed to the IPL-V interpreter. Error detection and tracing capability is provided, and a Post-Mortem in IPL-V format is printed at the conclusion of the execution phase. Execution speed is approximately 30,000 IPL-V instr/min.

USE

See the papers mentioned above for a basic description of the system and language; deviations from this description are described in the attached write-up.

NOTES

While this distribution is to be considered a field-test trial of the system, it has been used at The RAND Corporation for a number of complex problems, including the proof of logic theorems using the General Problem Solver coded in IPL-V. This program contains more than 6,000 IPL-V instructions, and takes more than 500,000 interpretation cycles for a complete run. Numerous other problems have also been interpreted and executed. The system has also been run on the 7090 using the compatibility package. A 7090 system, running within the MockDonald monitor, is currently being tested at RAND.

Any questions regarding the operation or correctness of the system should be directed to:

C. L. Baker
The RAND Corporation
1700 Main Street
Santa Monica, California.

704 IPL-V

Program Description

This write-up includes only deviations from and additions to the basic IPL-V manual, INFORMATION PROCESSING LANGUAGE V MANUAL, Section II, Programmers' Reference Manual, A. Newell, et al, The RAND Corporation paper P-1918, March 31, 1960. Familiarity with this manual is assumed, and references are to sections and page numbers in this manual.

INITIAL LOADING, Page 72

This version is not set up to run multiple IPL-V jobs. Each run must begin with the loading, on line, of the IPL-V system in binary form, followed by the IPL-V program, set up as described on page 79.

Since all output is off-line, on tape 10, this unit must be in ready status. This tape is never rewound, nor is an end-of-file written. This permits stacking of the IPL-V output with other jobs adhering to the same output conventions.

A TYPE 9 CARD (page 78) is treated as a type 1 comment card, as no multiple job provisions are included.

TYPE 2 CARDS do not have the full flexibility described on pages 73-74. Only the NAME and LINK fields are read; the first character of the NAME field must be the region symbol; the LINK field must contain the extent of the region. The first example in page 74 is the only acceptable format for type 2 cards.

TYPE 3 CARDS are not recognized. A block of storage has already been reserved for line printing; other print lines may be reserved by other means (see below under line printing). If a type 3 card is present, it will be treated as a type 2 card.

TYPE 5 CARDS do not have the full flexibility as specified on pages 77-78. P is disregarded; all input is assumed to be in IPL standard format. Q is interpreted as described; however, there is a maximum of 150 local symbols and a maximum of 100 internal symbolics that may be used per routine or data list structure.

SYMB does not control the input unit, and is disregarded unless regional.

LINK character d is ignored; the only output from the assembly process is the listing on tape 10, as controlled by character c.

TYPE 6 AND 7 CARDS are treated as type 5 headers as no auxiliary storage capability is included.

NOTE: TYPE 1 CARDS should not appear in the middle of a program or data list following a program or data card with a blank SYMB or LINK. This will lead to an error which is not detected at assembly time. (The blank LINK or SYMB is treated as 0.)

ALTERNATIVE INPUT UNIT. Provision has been included for reading input from tape 9 in BCD format if desired. Sense switch 6 controls this function. If switch 6 is ON when the program is first loaded, the IPL-V leader will initially read from tape 9 in BCD. The order and format of cards on this tape must be identical to that used for on-line reading of cards. When an end-of-file is encountered on this tape, the IPL-V loader will attempt to read from the on-line reader.

NOTE: If an end-of-file is not sensed before loading from tape 9 has been terminated by a type 5 card with a regional symbol, the input unit from which J165 (Load more routines and data) will attempt to read will be tape 9. Also, if memory is saved for restart, the reloading procedure will ignore the setting of switch 6, and the unit from which J165 will read will be that which was current when the memory was saved. (See Save for Restart and Reloading.)

EXAMPLE. A short IPL-V program is to be loaded and executed. The deck should be set up exactly as described, and placed behind the IPL-V binary deck in the on-line reader. Switch 6 should be up. Clear, load cards. Loading and execution will take place as described.

EXAMPLE. A long IPL-V program is to be loaded and executed. A BCD tape should be prepared off-line; the deck set-up should be exactly as described, except that the final type five card should not be written on the tape. The data is followed by an end-of-file on this tape. To load and execute, this tape is mounted on unit 9. The IPL-V binary deck is loaded on-line, followed by the final type five card, with switch 6 ON. Loading and execution will take place as described.

For subsequent runs, with routines to be changed or new data added, the same tape may be mounted, and the corrected routines and data loaded behind the IPL-V binary deck before the final type five card. Since the changed routines or data are read on-line after the tape has been read, the corrections will be made before interpretation begins.

INPROCESS LOADING, Page 80

The restrictions that apply to initial loading also apply to inprocess loading. Note that type 2 and type 3 cards may not be loaded. Regional symbols which were initially assigned by

type 2 cards are still available, as unused regionals are NOT returned to available space. (This compensates somewhat for the lack of type 3 symbol reservation.)

SAVE FOR RESTART, Page 81

J166 should read SAVE ON TAPE UNIT 2 FOR RESTART. All of memory is written onto tape 2 in self-loading form, and the machine stops. Pushing START will continue with the program from that point. The tape is rewound after writing, so that multiple occurrences of J166 will cause the tape to be re-written.

J167 is not available, as no auxiliary storage is provided.

RELOADING. Reloading is from the tape written by J166. The tape should be mounted on tape unit 1; LOAD TAPE is used to initiate the reload process. After reloading, control is returned to exactly the same place in the execution of J166 as when memory was saved, except that no halt occurs after the reload.

EXAMPLE. A program is to be loaded, saved immediately for subsequent restart, and the program is then to be executed. On reloading, it is desired to make changes or corrections before execution.

Include a one-word routine in the program as follows:

NAME	SYMB	LINK
RO	J166	J165

Place two type 5 termination cards at the end of the deck; the first with a SYMB of RO, the second naming the start of the program. Operation is then as follows: The program is read in and assembled, and interpretation begins with instruction RO. J166 will save memory on tape 2, and the program will stop. Pressing START will continue with the interpretation of the instruction, which links to J165. J165 will read more routines and data, including termination cards. Since the only card in the reader is a termination card, interpretation will begin with the first instruction of the program.

To reload, place the corrected routines or data lists in the card reader, each preceded by the proper type 5 header card, and followed with the same two termination cards. The tape written on the first pass is mounted on tape unit 1, and LOAD TAPE starts the reload process. Immediately after reload, interpretation will continue at RO; and J165 will read the corrections into memory. The first termination card will repeat

the save process on tape 2; the second will initiate the interpretation of the corrected program.

NOTE: The same procedure may be used when the initial program load is from tape 9 in BCD. The termination cards should not be on the tape, however, so that J165 will be prepared to read corrections from the on-line reader rather than tape 9. (See note above under Alternative Input Unit.)

TAPE UNIT SUMMARY

TAPE 1. Reload tape (self-loading).

TAPE 2. Save for Restart tape.

TAPE 9. BCD Input tape (under control of Switch 6).

TAPE 10. BCD Output tape (never rewound; no end-of-files).

OPERATION OF THE INTERPRETER

Interpretation begins within the system by interpreting the instruction

0 0 Xxx J7,

where Xxx is the routine named on the final type 5 termination card. Interpretation will continue until the J7 is linked to, at which time the program will print PROGRAM RAN TO COMPLETION on the output tape. Other stops will occur if the programmer executes the routine J7, or if the interpreter or a primitive process detects any one of a number of errors. In case of such a stop, the reason will be printed on the output tape, followed by an IPL-V POST MORTEM. The post mortem consists of a print out of the lists named on the list named by system cell W23. Initially this list names all the system cells, H0 through W29. The programmer may replace this list with another list, or add symbols to this list. The normal stop following a post mortem is 00000. Other stops may occur due to machine malfunction or for other undetectable IPL-V programming errors. Consult the program listing in case of such a stop.

ERROR TRAP, J170, Page 82

The error trap function is not provided; see above for error action.

SYSTEM CELLS, Page 35

Exseptions and additions are as follows:

H0 initially contains the symbol 0.

H2 will initially be about 26,000 cells for a 32K machine.

H3 is set to 1 by the initial loader.

H4 not used as no auxiliary storage is provided.

H5 is initially set to J4 (+).

H6 is used by machine language recursive routines to hold index 4.

H7 is a data term which is set to the current level. It is set to 0 by the initial leader, and by J165 if cards are read.

H8 is the hide-out cell used by the generator processes.

H9, H10, and H11 are generator working cells.

W0....W9 initially contain the symbol 0.

W10 initially names a data term for random number generation.

W12....W15 initially contain J0.

W16....W20 are not used. See above.

W21 initially names a data term equal to 1.

W22 initially names a data term equal to 1.

W23 initially names a list of the system cells.

W24 initially names a 120 word internal buffer for line printing.

W25 initially names a data term equal to 1.

W26...W28 are not used. See above.

GENERATOR PROCESSES, Pages 40-43

The generator trace context and level context are not restored. Subprocesses may be marked to trace by use of J147..J149.

J102, Page 52

Not available. May be coded as described in part I of the manual, page 82, if desired.

AUXILIARY STORAGE PROCESSES, Page 52

Not available. J72 and J74 assume list structure is in memory.

ARITHMETIC PROCESSES, Pages 55-57

J110-J113 admit fixed and/or floating data terms only.

J115-J117 admit fixed and/or floating data terms only.

J125 admits fixed and/or floating data terms only.

J128 admits fixed and/or floating data terms only.

INPUT-OUTPUT CONVENTIONS, Pages 61-64

Input and output units are assigned as described above. Processes J140-J146 are not available.

MONITOR POINTS AND TRACING, Pages 64-67

Sense switch 1 on is the signal for terminate for restart.

Sense switch 2 on is the signal for terminate.

Sense switch 3 on for full trace. Trace speed is approximately 40 IPL-V instr/sec.

No provision is made for suppressing tracing.

On the printed trace line, if S or (0) is a local symbol, this is indicated by an asterisk following the symbol; e.g., 2765*. The designated symbol S is printed only if Q = 1 or 2.

PRINT PROCESSES, Pages 69-72

The output unit is always tape 10. When local symbols are printed (except in the case of those translated by J150) they are followed by an asterisk.

If a list structure with a local name is printed by J150, the converted name will be 9-0. Sublists of any list structure are always numbered consecutively starting with 9-1.

Data terms are always printed in the format given on page 71, with the exception of floating point numbers. The format for these is "-.123456-12"; plus signs are not printed. The field length is thus either 10 or 11 columns.

Cell W24 initially names an internal print buffer of 120 cells (one character per cell) length. Other print lines may be assigned only by defining, via a type 2 card, a block of 120 consecutive regional cells. (E.g., 2 P0 120) P0 would then be a legitimate name for a print buffer line; the symbols P1....P119 would be unavailable for other use.

J158 and J159 leave 1W25 pointing to the first column to the left of the entered information.

SHARE PROGRAM SUBMITTAL

(Complete in Black Ink)

Dist.	No.			Date

To be added by SDA

Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor, SHARE or IBM as to the accuracy and functioning of the program and related program material and no responsibility is assumed by the contributor, SHARE or IBM in connection therewith.

Author's Submission Date

1.

Subj.	Code	Prog. Id.	Machine	Mach. Lang.	Core	Drum	Tapes
02	RS	IPL5-	704	SAP	8KC	00KD	1T
1. 3.	4. 6.	7. 13.		20.	27.	32.	37.

I/O	Subroutine	Primary Subj. Code	Secondary Subject Codes	* if Inactive	SDA Code	Local Use
-CH	CD	02	L2			
41.	45.	48.	51. 54. 57. 60. 63.	66.	68.	73. 80.

Filled in By SDA Only

2. **Program Title** RS IPL-V Interpretive System

3. **Author** C. L. Baker, H. S. Kelly **SHARE Sponsor Code** RS

Author Organization (if different from sponsor)

4. **Direct Inquiries to** C. L. Baker **Installation Code** RS

(Send copy to SHARE Sponsor)

Address The RAND Corp., 1700 Main St., Santa Monica, Cal. **Phone** EX 3-0411

5. **New Program? Yes** x **No**

If NO give previous distributions

Previous programs obsoleted

6. Have any changes whatsoever been made to the decks submitted since they were last checked out?

Yes ____ No x If YES, describe changes

7. Program Material Submitted:

		No. of Pages	Numbered From - To (inclusive)
7A. Paper:	Short write-up	2	
	Complete write-up (PA)	9+26+138+92	
	Listing (LS)	114+28 (~114+28)	
		No. of Cards	
7B. Cards:	Index cards	3	
	Abstract cards	4	0 - 3
	Symbolic (SY)	—	0001 -
	Binary (BI)	250	0001 - 0250
Check One:	☒ Row ☐ Column ☒ Absolute ☐ Relocatable		
	Fortran (FO)		-
	Squoze (SQ)		-
			-

8. General Comments: This initial distribution should be considered a field-test version of the 704 IPL-V system

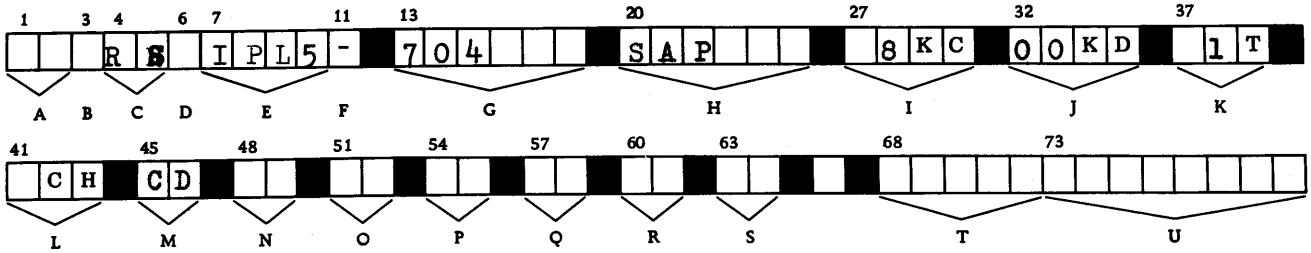
IBM Form No. 328-1238

SHARE PROGRAM DISTRIBUTION

Index Card Preparation

(Complete in Black Ink)

One set of index cards must accompany each contributed program



Explanation

A. Subject classification for this card. One card for each subject given in cols. 48-64. (Share Manual 03.21-04)

B. Machine Code: X for 7090, *for 709

C. Installation code. (Share Manual 03.21-11)

D. Fortran Code: F for all Fortran Programs

E. Program Identification (left justified). (Share Manual 03.21-04, 04.01-01-4, Standards)

F. Card serial number must be–(11 punch). (Share Manual 03.10 Catalog Card Format)

G. Machine for which program is written. Punch left justified as [7][0][4][ ]

[7][0][9][0] etc.

H. Program source language (left justified)

Use one of

F	1				
F	2				
S	A	P			
S	C	A	T		
S	A	P	-	F	

Fortran I

Fortran II

SAP for Fortran use

I. Minimum core required (1000's of words) ie [ ][4][K][C]

J. Minimum drum storage required (1000's of words) ie [0][0][K][D]

K. Minimum number of tape units required by program for its own use. ie [ ][4][T]

L. Number of I/O channels required by the program. ie [2][C][H]

M. Subroutine [S][R] or Independent routine [I][R] or Console Deck [C][D]

N. Primary subject classification code. (Share Manual 03.21.04, 04.01-01-4, Standards).

O.-S. Secondary subject classification codes (Share Manual 03.21.04, 04.01-01-4, Standards)

T. SHARE Distribution Number. (Entered by SDA only)

U. Available for local use.

Chuck Baker

23 November 1960 L-23579

Mr. Henry McCabe
Electronic Data Processing Department
Union Carbide Corporation
300 Madison Avenue, First Floor
New York 17, New York

Dear Mr. McCabe:

An initial, field-test version of the IPL-V (Information Processing Language V) interpretive system for the 709/7090 has been submitted to SDA for distribution to all interested SHARE members. While this is not a complete version of the system as specified in the Programmers' Reference Manual, it will suffice to run all but the largest IPL-V problems using a 32K machine, and may be used for small IPL-V programs on an 8K machine. It is more than adequate to indoctrinate the uninitiated in the use of this language.

This version is completely compatible with the 704 version previously submitted to SDA. Users who wish to switch from the use of the 704 system to the 709/7090 system at a later date may do so with little if any change in the IPL-V codes themselves.

The program has been incorporated as a file on the MockDonald system tape in use at The RAND Corporation. Interested persons should first obtain the material from SDA; we at RAND will be happy to aid users in incorporating the IPL-V system into their own monitors.

Sincerely,

C. L. Baker

CLB:ahe

Baker

28 November 1960 L-23612

Mr. Donald C. Cashman
Applied Programming Department
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

Dear Don:

We enclose the following material for distribution to SHARE members:

1. AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V (7 pages),
2. INFORMATION PROCESSING LANGUAGE V MANUAL, Section I, (138 pages),
3. INFORMATION PROCESSING LANGUAGE V MANUAL, Section II (92 pages),
4. RS IPLV Short Write-up (2 pages,
5. RS IPLV Program Description (11 pages),
6. SHARE Program Submittal Form (1 page).

Under separate cover we are mailing to you:

1. RS IPLV SQUOZE Deck (1193 cards),
2. RS IPLV Index Cards (3 cards),
3. RS IPLV Abstract Cards (4 cards).

This material constitutes the initial distribution of the 709/7090 IPLV interpretive system, and should be treated as a field-test version of the program.

Sincerely yours,

C. L. Baker (RS)

CLB:ahe

IDENTIFICATION

RS IPLV -- IPL-V Interpretive System
C. L. Baker, H. S. Kelly
The RAND Corporation, Santa Monica, California
November 25, 1960

PURPOSE

To interpret and execute programs written in the IPL-V language, as described in AN INTRODUCTION TO INFORMATION PROCESSING LANGUAGE V, and INFORMATION PROCESSING LANGUAGE V MANUAL, Sections I and II, The RAND Corporation papers P-1929, P-1897, and P-1918, 1960.

RESTRICTIONS

The program is in the form of a closed subroutine, designed for compatability with the MockDonald Operating System for the 709/7090. The calling sequence is in the same format as in the BCS routine within MD. The individual installation using this system may incorporate IPL as a file on the MD SYSTAP (as has been done at RAND), or execute the IPL program in phase I of the system, or integrate with any other system with a minimum of effort. The program may be run on any 709/7090 of at least 8K storage capacity and having at least three tape units. No on-line card reading or printing capability is provided.

METHOD

The IPL system is read into core, at which time it saves the rest of core (which would contain the operating system in use) on tape, then reads the IPL-V program into core and assembles it. Control is then passed to the IPL-V interpreter. Error detection and tracing capability is provided, and a Post-Mortem in IPL-V format is printed at the conclusion of execution. Core is then restored from tape, and control returned to the calling routine. Execution speed is approximately 210,000 IPL-V instructions per minute.

USE

See the papers mentioned above for a basic description of the system and language; deviations from this description are described in the attached write-up. Details of the calling sequence for the routine, etc., are also included.

NOTES

While this distribution is to be considered a field-test trial of the system, it is in use at The RAND Corporation for a number of complex problems, including the proof of logic

theorems using the General Problem Solver coded in IPL-V. This program contains more than 6,000 IPL-V instructions and takes more than 500,000 interpretation cycles for a complete run. Numerous other large problems have also been interpreted and executed. The inclusion of IPL-V in the MockDonald System at RAND permits IPL-V jobs to be stacked with other jobs and executed with all of core available to the IPL-V programmer.

Any questions regarding the operation or correctness of the IPL-V system should be directed to:

C. L. Baker
The RAND Corporation
1700 Main Street
Santa Monica, California.

Questions concerning the incorporation of the IPL-V system into the MockDonald Monitor should be directed to:

G. E. Bryan
The RAND Corporation
1700 Main Street
Santa Monica, California.

709/7090 IPL-V

Program Description

This write-up is in two parts; the first describing the nature of the IPL-V system as a subroutine, the second including deviations from and additions to the basic IPL-V manual.

PART I

Calling Sequence:

```
CS - 1 = LDI CNTRL
CS     = TSX AO,4
CS + 1 = ...
CS + 2 = PZE SYSPIT,,SYSMIT
CS + 3 = PZE SYSPOT,,,...
CS + 4 = PZE SYSES1,,SYSES2
CS + 5 = ...
CS + 6 = PZE ...,,SYSERR
           .
           .
           .
           .
CS + 13= PZE SYSORG,,...
           .
           .
           .
           .
CS + 29= Return
           .
           .
           .
           .
CS + 37= PZE SYSTSL,,...
```

The above calling sequence is that used within the MockDonald Operating System; hence its strange format. Words or parts of words indicated by "..." are not significant.

Control bits:

Bits 31-35 of CNTRL are placed in the Sense Indicator Register for conveying information to IPL-V as to the nature of the run being made. Immediately below is given the meaning of these bits; their interpretation is explained in Part 2.

```
Bit 35   0    LOAD
 "  "    1    RELOAD

Bit 34   0    NOTAPE
 "  "    1    TAPE
```

Bit 33	0	LIST
" "	1	NOLIST
Bit 32	0	NTRACE
"	1	TRACE
Bit 31	0	NODUMP
" "	1	DUMP

The program which includes the calling sequence may interpret a control card similar to that used to call SCAT, LOAD, etc., and set the bits accordingly.

Calling Sequence Information:

SYSPIT, SYSMIT, SYSMOT, SYSES1, SYSES2 are the locations of tape control words. These words must contain the address of the physical unit associated with the symbolic tape. (E.g., if SYSPIT is on physical unit B2, the control word would have an octal address of 02202.) Only the first three tapes are necessary for using the system. If the address is zero, the unit is interpreted as being unavailable.

SYSTSL is the location of the Tape Status list for reserved tapes. It is arranged such that

```
AXT  CHANNEL,2      (1 for Ch. A, 2 for Ch. B, etc.)
AXT  UNIT,1         (3 for unit 3, etc.)
CLA* SYSTSL,2
```

will put the address of the corresponding tape control word into the decrement of the accumulator. Tapes SYSAR1,2,3, SYSBR1,2, and 3 are interrogated. Again, a zero address is interpreted as meaning that the corresponding unit is unavailable.

SYSERR is the location to which the operator is to transfer in case of program error, loop, too much time, etc. Both SYSERR and SYSERR-2 are replaced with a transfer to the recovery routine for IPL-V; the contents of these cells is saved and restored. They must lie in the range 00001--00077$_8$.

SYSORG is the name of a cell containing the location of the first word following the system in control. Memory up to location C(SYSORG) is left undisturbed by the execution of IPL-V (with the exception of cell zero.)

Operation of the IPL-V executive routine, AO:

1) All traps are disabled.
2) The tape control words are interpreted and a table of tape information is constructed.
3) Memory is saved on tape SYSMIT from locations 100$_8$-C(SYSORG).

4) If the RELOAD option is called, Memory is reloaded from SYSAR2.
5) The remaining Sense indicator bits are interpreted.
6) Control is passed to the IPL-V system interpreter.
7) At the conclusion of the IPL-V run (Program finished, error, or manual transfer to SYSTEM or SYSERR), Memory is restored from tape SYSMIT, and control returned to word 29 of the calling sequence.

Notes:

Although the system has been organized so as to work within the MockDonald Monitor System, it should be emphasized that it is in no way dependent upon any of the routines of that system. Only the tape control words of that system need be reproduced (in the format explained above) to gain complete independence of that system.

All input and output routines are contained within the IPL-V system itself. Data is read, in BCD, from tape SYSPIT; output is written in BCD on tape SYSPOT. Neither tape is rewound, nor is an end of file ever written on SYSPOT. Tape SYSMIT is used for saving core only; if it is not desired to retain this feature, routines A2 and A3 may be replaced by TRA 1,4. SYSAR1, 2 and 3 may be used by the IPL-V interpreted program; see following section for details. SYSES1,2, SYSBR1,2, and 3 are never used, and have been included in the tape set-up routine only in anticipation of possible future expansion of the IPL-V system.

Example:

Following is a complete calling sequence requiring no information from the MockDonald System:

```
START    LDI     CNTRL
         TSX     A0,4
         PZE
         PZE     SYSPIT,,SYSMIT
         PZE     SYSPOT,,
         PZE     SYSES1,,SYSES2
         PZE
         PZE     0,0,3            Manual error transfer to 3(8) or 1(8).
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE     SYSORG
         PZE
         PZE
         PZE
```

```
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            TRA     out
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE
            PZE     SYSTSL
CNTRL       OCT     23          RELOAD,TAPE,LIST,NTRACE,DUMP  (Example)
SYSPIT      OCT     2201        Unit B1 for SYSPIT
SYSMIT      OCT     2202        Unit B2 for SYSMIT
SYSPOT      OCT     2203        Unit B3 for SYSPOT
SYSES1      PZE                 SYSES1 not used.
SYSES2      PZE                 SYSES2 not used.
SYSAR1      OCT     1201        Unit A1 for SYSAR1
SYSAR2      OCT     1202        Unit A2 for SYSAR2
SYSAR3      OCT     1203        Unit A3 for SYSAR3
SYSBR1      PZE                 SYSBR1 not used.
SYSBR2      PZE                 SYSBR2 not used.
SYSBR3      PZE                 SYSBR3 not used.
SYSORG      PZE     16384       Save lower half of core.
            PZE     CHNLB,1
            PZE     CHNLA,1
SYSTSL      PZE                 Tape Status List.
            PZE     0,,SYSAR3
            PZE     0,,SYSAR2
            PZE     0,,SYSAR1
CHNLA       PZE
            PZE     0,,SYSBR3
            PZE     0,,SYSBR2
            PZE     0,,SYSBR1
CHNLB       PZE
```

PART II

This write-up includes only deviations from and additions to the basic IPL-V manual, INFORMATION PROCESSING LANGUAGE V MANUAL, Section II, Programmers' Reference Manual, A. Newell, et al, The RAND Corporation paper P-1918, March 31, 1960. Familiarity with this manual is assumed, and references are to sections and page numbers in this manual.

INITIAL LOADING, Page 72

This version is not set up to run multiple IPL-V jobs. Each run must begin with the execution of the calling sequence for IPL-V. References to bits 31-35 refer to the setting of the Sense Indicator register on entry to IPL-V. (See Part 1.) The IPL-V program, set up as described on page 79, is assumed to be on tape SYSPIT, in BCD. This tape is not rewound prior to reading.

All output is off-line, on tape SYSPOT. This tape is never rewound, nor is an end-of-file written. This permits stacking of the IPL-V output with other jobs adhering to the same output conventions.

A TYPE 9 CARD (page 78) is treated as a type 1 comment card, as no multiple job provisions are included.

TYPE 2 CARDS do not have the full flexibility described on pages 73-74. Only the NAME and LINK fields are read; the first character of the NAME field must be the region symbol; the LINK field must contain the extent of the region. The first example in page 74 is the only acceptable format for type 2 cards.

A TYPE 3 CARD, if it appears, must precede any type 2 cards. It may be used to reserve a block of storage for binary IPL-V routines written by the programmer. The block will start at location 100_8, and the extent of the block will be as given in LINK. No other uses for type 3 cards are permitted. A block of storage has already been reserved for line printing; other print lines may be reserved by other means (see below under line printing).

TYPE 5 CARDS do not have the full flexibility as specified on pages 77-78. P must be zero or blank for standard IPL format, or 3 for absolute column binary format. Q is interpreted as described; however, there is a maximum of 150 local symbols and a maximum of 100 internal symbolics that may be used per routine or data list structure.

SYMB does not control the input unit, and is disregarded unless regional.

LINK character d is ignored; the only output from the assembly process is the listing on SYSPOT, as controlled by character c.

TYPE 6 AND 7 CARDS are treated as type 5 headers as no auxiliary storage capability is included.

NOTE: TYPE 1 CARDS should not appear in the middle of a program or data list following a program or data card with a blank SYMB or LINK. This will lead to an error which is not detected at assembly time. (The blank LINK or SYMB is treated as 0.)

ABSOLUTE BINARY INPUT. A type 5 card with P = 3 indicates that a block of absolute, column binary, cards follows. These must load into the space reserved by the single type 3 card at the beginning of the program. The last binary card must be a transfer card; the address is ignored, however. After the binary cards have been read, the symbols HO, JO, and WO are stored in the address parts of location 100_8, 101_8, and 102_8, respectively. This permits the programmer to refer to these regions conveniently when coding machine language programs.

ALTERNATIVE INPUT UNIT. Provision has been included for reading input from tape SYSAR1 in BCD format if desired. Sense indicator bit 34 controls this function. If bit 34 is TAPE when the program is first loaded, the IPL-V loader will initially read from tape SYSAR1 in BCD. The order and format of cards on this tape must be identical to that used for on-line reading of cards. When an end-of-file is encountered on this tape, the IPL-V loader will attempt to read from tape SYSPIT. Printing of the assembly of tape SYSAR1 may be suppressed by setting bit 33 to NOLIST; in this case only the header cards from SYSAR1 will print. If bit 33 is LIST, printing control is from character c of the header cards on the tape.

EXAMPLE. A short IPL-V program is to be loaded and executed. The deck should be set up exactly as described, and written on SYSPIT in BCD. Bit 34 should be NOTAPE. Execution of the IPL-V calling sequence will cause the program to be loaded and executed as described.

EXAMPLE. A long IPL-V program is to be loaded and executed. A BCD tape should be prepared off-line; the deck set-up should be exactly as described, except that the final type five card should not be written on the tape. The data is followed by an end-of-file on this tape. To load and execute, this tape is mounted on unit SYSAR1. The final type five card is loaded on SYSPIT, with bit 34 = TAPE, bit 35 = LOAD. Loading and execution will take place as described.

For subsequent runs, with routines to be changed or new data added, the same tape SYSAR1 may be mounted, and the corrected routines and data loaded on SYSPIT before the final type five card. Since the changed routines or data are read from SYSPIT after tape SYSAR1 has been read, the corrections will be made before interpretation begins.

INPROCESS LOADING, Page 80

The restrictions that apply to initial loading also apply to inprocess loading. Note that type 2 and type 3 cards may not be loaded. Regional symbols which were initially assigned by type 2 cards are still available, as unused regionals are NOT returned to available space. (This compensates somewhat for the lack of type 3 symbol reservation.)

SAVE FOR RESTART, Page 81

J166 should read SAVE ON TAPE SYSAR3 FOR RESTART. All of memory is written onto tape, and the program continues from that point. The tape is rewound after writing, so that multiple occurrences of J166 will cause the tape to be rewritten. Bit 31 should be DUMP if J166 is to be used.

J167 is not available, as no auxiliary storage is provided.

RELOADING. Reloading is from the tape written by J166. The tape should be mounted on tape SYSAR2; bit 35 = RELOAD is used to initiate the reload process. After reloading, control is returned to exactly the same place in the execution of J166 as when memory was saved.

EXAMPLE. A program is to be loaded, saved immediately for subsequent restart, and the program is then to be executed. On reloading, it is desired to make changes or corrections before execution.

Include a one-word routine in the program as follows:

NAME	SYMB	LINK
R0	J166	J165

Place two type 5 termination cards at the end of the deck; the first with a SYMB of R0, the second naming the start of the program; bit 35 = LOAD, bit 31 = DUMP. Operation is then as follows: The program is read in and assembled, and interpretation begins with instruction R0. J166 will save memory on SYSAR3, and the program will continue with the interpretation of the instruction, which links to J165. J165 will read more routines and data, including termination cards. Since the only card remaining is a termination card, interpretation will begin with the first instruction of the program.

To reload, place the corrected routines or data lists on SYSPIT, each preceded by the proper type 5 header card, and followed with the same two termination cards. The tape written on the first pass is mounted on tape SYSAR2. Bit 35 = RELOAD, bit 34 = NOTAPE, bit 31 = DUMP. Immediately after reload, interpretation will continue at R0; and J165 will read the corrections into

memory. The first termination card will repeat the save process on SYSAR3; the second will initiate the interpretation of the corrected program.

NOTE: The same procedure may be used when the initial program load is from SYSAR1 in BCD. Bit 34 should be NOTAPE on reload, however, so that J165 will be prepared to read corrections from SYSPIT rather than SYSAR1.

TAPE UNIT SUMMARY

TAPE SYSPIT. BCD Input tape (never rewound).

TAPE SYSPOT. BCD Output tape (never rewound; no end-of-files).

TAPE SYSAR1. Alternate BCD Input tape (under control of bit 34).

TAPE SYSAR2. Reload tape (under control of bit 35).

TAPE SYSAR3. Save for Restart tape (under control of bit 31 and J166).

OPERATION OF THE INTERPRETER

Interpretation begins within the system by interpreting the instruction

0 0 Xxx J7,

where Xxx is the routine named on the final type 5 termination card. Interpretation will continue until the J7 is linked to, at which time the program will print PROGRAM RAN TO COMPLETION on the output tape. Other terminations will occur if the programmer executes the routine J7, or if the interpreter or a primitive process detects any one of a number of errors. In case of such a termination, the reason will be printed on the output tape, followed by an IPL-V POST MORTEM. The post mortem consists of a print-out of the lists named on the list named by system cell W23. Initially this list names all the system cells, H0 through W29. The programmer may replace this list with another list, or add symbols to this list. Control is returned to 29,4 of the IPL-V calling sequence after execution.

ERROR TRAP, J170, Page 82

The error trap function is not provided; see above for error action.

SYSTEM CELLS, Page 35

Exceptions and additions are as follows:

H0 initially contains the symbol 0.

H2 will initially be about 26,000 cells for a 32K machine.

H3 is set to 1 by the initial loader.

H4 not used as no auxiliary storage is provided.

H5 is initially set to J4(+).

H6 is used by machine language recursive routines to hold index 4.

H7 is a data term which is set to the current level. It is set to 0 by the initial loader, and by J165 if cards are read.

H8 is the hide-out cell used by the generator processes.

H9, H10, and H11 are generator working cells.

W0....W9 initially contain the symbol 0.

W10 initially names a data term for random number generation.

W12....W15 initially contain J0.

W16....W20 are not used. See above.

W21 initially names a data term equal to 1.

W22 initially names a data term equal to 1.

W23 initially names a list of the system cells.

W24 initially names a 120 word internal buffer for line printing.

W25 initially names a data term equal to 1.

W26...W28 are not used. See above.

GENERATOR PROCESSES, Pages 40-43

The generator trace context and level context are not restored. Subprocesses may be marked to trace by use of J147..J149.

J102, Page 52

Not available. May be coded as described in part I of the manual, page 82, if desired.

AUXILIARY STORAGE PROCESSES, Page 52

Not available. J72 and J74 assume list structure is in memory.

ARITHMETIC PROCESSES, Pages 55-57

J110-J113 admit fixed and/or floating data terms only.

J115-J117 admit fixed and/or floating data terms only.

J125 admits fixed and/or floating data terms only.

J128 admits fixed and/or floating data terms only.

INPUT-OUTPUT CONVENTIONS, Pages 61-64

Input and output units are assigned as described above. Processes J140-J146 are not available.

MONITOR POINTS AND TRACING, Pages 64-67

No console signals are provided. Terminate for restart is not available. The program may be terminated by a manual TRA SYSERR or SYSTEM. Sense bit 32 = TRACE signals full trace. Trace speed is approximately 80 IPL-V instr/sec. Bit 32 = NTRACE gives normal tracing modes.

No provision is made for suppressing tracing.

On the printed trace line, if S or (0) is a local symbol, this is indicated by an asterisk following the symbol; e.g., 2765*. The designated symbol S is printed only if Q = 1 or 2.

PRINT PROCESSES, Pages 69-72

The output unit is always SYSPOT. When local symbols are printed (except in the case of those translated by J150) they are followed by an asterisk.

If a list structure with a local name is printed by J150, the converted name will be 9-0. Sublists of any list structure are always numbered consecutively starting with 9-1.

Data terms are always printed in the format given on page 71, with the exception of floating point numbers. The format for these is "-.123456-12"; plus signs are not printed. The field length is thus either 10 or 11 columns.

Cell W24 initially names an internal print buffer of 120 cells (one character per cell) length. Other print lines may be

assigned only by defining, via a type 2 card, a block of 120 consecutive regional cells. (E.G., 2 P0 120)
P0 would then be a legitimate name for a print buffer line; the symbols P1....P119 would be unavailable for other use.

J158 and J159 leave 1W25 pointing to the first column to the left of the entered information.

SHARE PROGRAM SUBMITTAL

(Complete in Black Ink)

Dist.	No.			Date

To be added by SDA

Author's Submission Date

1.

Subj. Code	Prog. Id.	Machine	Mach. Lang.	Core	Drum	Tapes
ϕ2XRS	IPLV-	709/90	SCRT	8KC	00KD	3T
1. 3. 4.	6. 7.	13.	20.	27.	32.	37.

I/O	Subroutine	Primary Subj. Code	Secondary Subject Codes	* if Inactive	SDA Code	Local Use
1CH	FR	ϕ2	L2			
41.	45.	48.	51. 54. 57. 60. 63.	66.	68.	73. 80.

Filled in By SDA Only

2. Program Title RS IPL-V Interpretive System

3. Author C.L. Baker, H.S. Kelly SHARE Sponsor Code RS

Author Organization (if different from sponsor) _____

4. Direct Inquiries to C.L. Baker Installation Code RS

(Send copy to SHARE Sponsor)

Address The Rand Corp. 1700 Main St. Phone Ex 30411

5. New Program? Yes X No _____

If NO give previous distributions _____

Previous programs obsoleted _____

6. Have any changes whatsoever been made to the decks submitted since they were last checked out?

Yes _____ No X If YES, describe changes _____

7. Program Material Submitted:

		No. of Pages / No. of Cards	Numbered From - To (inclusive)
7A. Paper:	Short write-up	2	
	Complete write-up (PA)	11+7+138+92	
	Listing (LS)	—	
7B. Cards:	Index cards	3	
	Abstract cards	4	0 - 3
	Symbolic (SY)		0001 -
	Binary (BI)		-
Check One:	☐ Row ☒ Column ☐ Absolute ☐ Relocatable		
	Fortran (FO)		-
	Squoze (SQ)	1193	0001 - 1193
			-

8. General Comments: This initial version should be considered a field-test version of the 709/7090 IPL-V System

IBM Form No. 328-1238

6 January 1961 L-314

Mrs. Clare L. Winter
Program Librarian
Bell Telephone Laboratories
Murray Hill, New Jersey

Dear Mrs. Winter:

Thank you for your inquiry regarding RS IPL V for the 7090. The distribution of the material you requested has been made via SDA; it is distribution number 1027.

However, we are not sending your tape back blank. It now contains the current RAND version of the MD9 MacDonald Monitor. I understand that Bell is not using the SOS system, so a few words about the tape are in order. A short deck of cards (enclosed) is loaded on-line to initiate the system. The tape assignments are made as shown on the attached sheet. Jobs will be read from SYSPIT, processed, and the output written on SYSPOT. If you intend to use this tape for IPL V jobs only, the following information plus the enclosed write-up (part of SDA 1027) should suffice.

Cards on SYSPIT:

Col:	1	8-14	16.....	
1)	7/8/9	JOB	Identifying info for job.	
2)	7/8/9	ASSIGN	A5-SYSAR1	Tape assignment cards as needed. See write-up for tape usage.
		⋮		
3)	7/8/9	IPL	A,B,C,D,E	IPL control card. A= LOAD or RELOAD B= TAPE or NOTAPE C= LIST or NOLIST D= TRACE or NTRACE E= DUMP or NODUMP The normal case is as underlined.

Col: 1 8-14 16.....

4) Cards in IPL V format.

5) Repeat of above sequence for remaining jobs.

6) 7/8/9 END

I suggest that you request distribution number 1027 from SDA in order to obtain the SQUOZE deck for IPL V. This may be listed using the MD9 tape if desired. We do not have a symbolic deck available.

I hope that this information will enable you to run IPL jobs on your 7090. If you have trouble with the MackDonald Monitor operation, I suggest you talk with Owen Hansen. I will be happy to help out if this doesn't work. For any questions regarding the operation of IPL V itself, please feel free to contact me at any time.

Sincerely yours,

C. L. Baker

CLB:ls

Enclosures:
1. Magnetic Tape
2. IBM Card Deck
3. "System Tapes"
4. "RS MD9A SYSTAP"
5. IPL V Program Description
6. IPL V Short Write-Up

Ref. Inc. No. 238

6 January 1961 L-314

Mrs. Clare L. Winter
Program Librarian
Bell Telephone Laboratories
Murray Hill, New Jersey

Dear Mrs. Winter:

Thank you for your inquiry regarding RS IPL V for the 7090. The distribution of the material you requested has been made via SDA; it is distribution number 1027.

However, we are not sending your tape back blank. It now contains the current RAND version of the MD9 MacDonald Monitor. I understand that Bell is not using the SOS system, so a few words about the tape are in order. A short deck of cards (enclosed) is loaded on-line to initiate the system. The tape assignments are made as shown on the attached sheet. Jobs will be read from SYSPIT, processed, and the output written on SYSPOT. If you intend to use this tape for IPL V jobs only, the following information plus the enclosed write-up (part of SDA 1027) should suffice.

Cards on SYSPIT:

Col:	1	8-14	16.....	
1)	7/8/9	JOB	Identifying info for job.	
2)	7/8/9	ASSIGN	A5=SYSAR1	Tape assignment cards as needed. See write-up for tape usage.
		:		
3)	7/8/9	IPL	A,B,C,D,E	IPL control card. A= LOAD or RELOAD B= TAPE or NOTAPE C= LIST or NOLIST D= TRACE or NTRACE E= DUMP or NODUMP The normal case is as underlined.

Col: 1 8-14 16.....

4) Cards in IPL V format.

5) Repeat of above sequence for remaining jobs.

6) 7/8/9 END

I suggest that you request distribution number 1027 from SDA in order to obtain the SQUOZE deck for IPL V. This may be listed using the MD9 tape if desired. We do not have a symbolic deck available.

I hope that this information will enable you to run IPL jobs on your 7090. If you have trouble with the MacDonald Moniter operation, I suggest you talk with Owen Hansen. I will be happy to help out if this doesn't work. For any questions regarding the operation of IPL V itself, please feel free to contact me at any time.

Sincerely yours,

C. L. Baker

CLB:ls

Enclosures:
1. Magnetic Tape
2. IBM Card Deck
3. "System Tapes"
4. "RS MD9A SYSTAP"
5. IPL V Program Description
6. IPL V Short Write-Up

Ref. Inc. No. 238

MAJOR DIFFERENCES BETWEEN IPL-V'S

OF 11/22/60 AND 6/22/61

1. An extra word is referenced in the calling sequence. The decrement of 3,4 contains the address of a cell containing the number of pages of output expected. This number is multiplied by 64 to give a line count. If this is zero, a count of 6000 lines is assumed. When the line count is exceeded, and error message and post mortem are printed.

2. On a manual transfer to SYSERR, an attempt is made to print a post mortem. On a transfer to SYSERR-2, no attempt is made, and the system cleans up immediately.

3. The interrogation of the tape status list now includes SYSAR4 and 5, and SYSBR4 and 5. Any or all of these ten reserved tapes may be used by J15n and/or J166.

4. Type 3 cards are all handled but may not use absolute locations for reserving space. The corresponding type 2 card must have already appeared. When P=1, Q=1, the BCD info is read from columns 1-72 of the next card; and if the line length is greater than 72, from columns 1-47 of the second card. The maximum line length is 119 characters.

5. Type 6 or 7 header cards may be used to put routines on auxiliary tape storage. (The back end of SYSMIT is used for this.) If a type 6 or 7 header is used for data, type

5 is assumed. A buffer longer than the longest routine on auxiliary must have been specified by a type 3 card. The head of the routine is set to Q=7, LINK=no. of routine on auxiliary.

✓6. A block of storage for absolute routines need not have been reserved before loading these. If none was reserved, H, J, and W locations are stored in -3, -2, and -1. This is useful for loading routines into available space not yet used, by patching av. sp. around the area loaded. (See the line read primitives for an example.)

✓7. J165 will also load routines to auxiliary storage, including new copies of old routines.

✓8. J166 now takes the name of an integer data term as an input. The value of this data determines the unit used for saving. 0 denotes the normal unit, SYSAR3. 1 thru 5 denote SYSAR1 thru 5; 6 thru 10 denote SYSBR1 thru 5. Reloading is still from SYSAR2, always.

The routine previously written as:	R99		J166	J165
would now be written as:	R99	10	9-1	
			J166	J165
	9-1	+01		0

9. Thirteen extra lists are printed following W29 on the post mortem. These contain information about J105-J107 usage. (See listing for details.) The contents of the current print line (1W24) is also printed.

10. H2 has about 24,000 cells initially. H12 is a cell used by the generator processes. W20 controls the output unit for J15n.

11. The arithmetic processes now handle all types of data terms. Conversion of BCD and octal to other types is primarily accomplished by changing q only.

12. The data term named in W20 (set to 0 initially) controls the output unit and mode for J150, J152, J153, and J155. 0 denotes the normal print mode on unit SYSPOT. 1 thru 5 denote print mode on units SYSAR1 thru 5; 6 thru 10 denote print mode on units SYSBR1 thru 5; 11 thru 15 denote punch mode (first 80 characters only, no spacing character) on units SYSAR1 thru 5; and 16 thru 20 denote punch mode on units SYSBR1 thru 5. An end-of-file is written on any units so used, before exit from IPL.

13. The generator trace modes and levels will operate as in the manual provided generators are executed from the left.

14. Processes J105, J107, and J108 are operating. Tapes SYSES1 and SYSES2 are used for data filed in auxiliary. J72, J74, J100, and J101 operate on data list structures in auxiliary.

15. All routines and data on auxiliary storage are saved and reloaded by J166 and the reload options.

✓16. Unused regional symbols are returned to the end of av. space at the end of loading; if new regionals are required on a subsequent reload, they should be reserved with a type 3 card at initial load. The unused regionals are not removed from the symbol table, however, and references to them at reload time will really be references to cells on available space -- probably causing trouble.

C. L. Baker

Revised 7/1/61

709/7090 IPL-V

PROGRAM DESCRIPTION

This write-up is in two parts: the first describing the nature of the IPL-V system as a subroutine, the second including deviations from and additions to the basic IPL-V manual.

PART I

CALLING SEQUENCE

```
CS - 1    =  LDI CNTRL
CS        =  TSX AO,4
CS + 1    =  ...
CS + 2    =  PZE SYSPIT,,SYSMIT
CS + 3    =  PZE SYSPOT,,PAGES
CS + 4    =  PZE SYSES1,,SYSES2
CS + 5    =  ...
CS + 6    =  PZE ...,,SYSERR
               .
               .
               .
CS + 13   =  PZE SYSORG,,...
               .
               .
               .
CS + 29   =  Return
               .
               .
               .
CS + 37   =  PZE SYSTSL,,...
```

The above calling sequence is that used within the MockDonald operating system; hence its strange format. Words or parts of words indicated by "..." are not significant.

CONTROL BITS

Bits 31-35 of CNTRL are placed in the sense indicator register for conveying information to IPL-V as to the nature of the run being made. Immediately below is given the meaning of these bits; their interpretation is explained in Part 2.

```
Bit 35   0    LOAD
Bit 35   1    RELOAD
```

Bit 34	0	NOTAPE
Bit 34	1	TAPE
Bit 33	0	LIST
Bit 33	1	NOLIST
Bit 32	0	NTRACE
Bit 32	1	TRACE
Bit 31	0	NODUMP
Bit 31	1	DUMP

The program which includes the calling sequence may interpret a control card similar to that used to call SCAT, LOAD, etc., and set the bits accordingly.

CALLING SEQUENCE INFORMATION

SYSPIT, SYSMIT, SYSMOT, SYSES1, SYSES2 are the locations of tape control words. These words must contain the address of the physical unit associated with the symbolic tape. (E.g., if SYSPIT is on physical unit B2, the control word would have an octal address of 02202.) Only the first three tapes are necessary for using the system. If the address is zero, the unit is interpreted as being unavailable.

SYSTSL is the location of the tape status list for reserved tapes. It is arranged such that

```
AXT   CHANNEL,2     (1 for Ch. A, 2 for Ch. B, etc.)
AXT   UNIT,1        (3 for Unit 3, etc.)
CLA*  SYSTSL,2
```

will put the address of the corresponding tape control word into the decrement of the accumulator. Tapes SYSAR1, 2, 3, 4, 5, SYSBR1, 2, 3, 4, and 5 are interrogated. Again, a zero address is interpreted as meaning that the corresponding unit is unavailable.

SYSERR and SYSERR-2 are the locations to which the operator is to transfer in case of program error, loop, too much time, etc. Both SYSERR and SYSERR-2 are replaced with a transfer to the recovery routine for IPL-V; the contents of these cells is saved and restored. They must lie in the range 00001 - 00077_8. The entry from SYSERR attempts to give an IPL-V post mortem before exit; the entry from SYSERR-2 causes an immediate exit from IPL-V.

SYSORG is the name of a cell containing the location of the first word following the system in control. Memory up to location C(SYSORG) is left undisturbed by the execution of IPL-V (with the exception of cell zero).

PAGES is the location of a cell containing the approximate number of pages of output expected. This number is multiplied by 64 to give a line count. (A line count of zero is replaced with a count of 6000.) An error condition is sensed when the output (from assembly, tracing, J150, J151, J152, J153, or J155, printing or punching) exceeds the expected amount. The line count is then reset so that a post mortem may be printed.

OPERATION OF THE IPL-V EXECUTIVE ROUTINE, AO

1. All traps are disabled.
2. The tape control words are interpreted, and a table of tape information is constructed.
3. Memory is saved on tape SYSMIT from locations 100_8-C(SYSØRG).
4. If the RELOAD option is called, memory is reloaded from SYSAR2.
5. The remaining sense indicator bits are interpreted.
6. Control is passed to the IPL-V system interpreter.
7. At the conclusion of the IPL-V run (program finished, error, or manual transfer to SYSERR-2 or SYSERR), memory is restored from tape SYSMIT and control returned to word 29 of the calling sequence.

NOTES

Although the system has been organized so as to work within the MockDonald monitor system, it should be emphasized that it is in no way dependent upon any of the routines of that system. Only the tape control words of that system need be reproduced (in the format explained above) to gain complete independence of that system.

All input and output routines are contained within the IPL-V system itself. Data is read, in BCD, from tape SYSPIT; output is written in BCD on tape SYSPOT. Neither tape is rewound, nor is an end-of-file ever written on SYSPOT. Tape SYSMIT is used for saving core and for auxiliary routine storage. SYSAR1, 2, 3, 4, 5, SYSBR1, 2, 3, 4, and 5 may be used by the IPL-V interpreted program. See following section for details.

EXAMPLE

Following is a complete calling sequence requiring no information from the MockDonald system:

```
START    LDI  CNTRL
         TSX  AO,4
         PZE
         PZE  SYSPIT,,SYSMIT
         PZE  SYSPOT,,PAGES
         PZE  SYSES1,,SYSES2
         PZE
         PZE  0,0,3  (Manual Error Transfer to 3₈ or 1₈.)
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE  SYSORG
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         TRA  Out
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE  SYSTSL
CNTRL    OCT  23     RELOAD,TAPE,LIST,NTRACE,DUMP  (Example)
SYSPIT   OCT  2201   Unit B1 for SYSPIT
SYSMIT   OCT  2202   Unit B2 for SYSMIT
SYSPOT   OCT  2203   Unit B3 for SYSPOT
SYSES1   OCT  1204   Unit A4 for SYSES1 (If Aux. Data Storage Used)
SYSES2   OCT  1205   Unit A5 for SYSES2 (If Aux. Data Storage Used)
SYSAR1   OCT  1201   Unit A1 for SYSAR1
SYSAR2   OCT  1202   Unit A2 for SYSAR2
```

```
SYSAR3    OCT  1203    Unit A3 for SYSAR3
SYSAR4    PZE           SYSAR4 Not Used
SYSAR5    PZE           SYSAR5 Not Used
SYSBR1    PZE           SYSBR1 Not Used
SYSBR2    PZE           SYSBR2 Not Used
SYSBR3    PZE           SYSBR3 Not Used
SYSBR4    OCT  2204    Unit B4 for SYSBR4 (If Used)
SYSBR5    OCT  2205    Unit B5 for SYSBR5 (If Used)
SYSORG    PZE  16384   Save lower half of core.
          PZE  CHNLB,1
          PZE  CHNLA,1
SYSTSL    PZE           Tape Status List
          PZE  0,,SYSAR5
          PZE  0,,SYSAR4
          PZE  0,,SYSAR3
          PZE  0,,SYSAR2
          PZE  0,,SYSAR1
CHNLA     PZE
          PZE  0,,SYSBR5
          PZE  0,,SYSBR4
          PZE  0,,SYSBR3
          PZE  0,,SYSBR2
          PZE  0,,SYSBR1
CHNLB     PZE
PAGES     DEC  200     Approximate Number of Pages (Example)
```

PART II

This write-up includes only deviations from and additions to the basic IPL-V manual, INFORMATION PROCESSING LANGUAGE V MANUAL, Section II, Programmers' Reference Manual, A. Newell, et al, The RAND Corporation paper P-1918, March 31, 1960. Familiarity with this manual is assumed, and references are to sections and page numbers in this manual.

INITIAL LOADING, Page 72

This version is not set up to run multiple IPL-V jobs. Each run must begin with the execution of the calling sequence for IPL-V. References to bits 31-35 refer to the setting of the sense indicator register on entry to IPL-V. (See Part I.) The IPL-V program, set up as described on page 79, is assumed to be on tape SYSPIT, in BCD. This tape is not rewound prior to reading.

All assembly, tracing, error, and post mortem output is off-line, on tape SYSPOT. This tape is never rewound, nor is an end-of-file written. This permits stacking of the IPL-V output with other jobs adhering to the same output conventions. Output from J150, J151, J152, J153, and J155 may be on reserved tapes if so specified (see below). An end-of-file is written on reserved tapes so used, at the conclusion of the program.

A TYPE 9 CARD (page 78) is treated as a type 1 comment card, as no multiple job provisions are included.

TYPE 2 CARDS do not have the full flexibility described on pages 73-74. Only the NAME and LINK fields are read; the first character of the NAME field must be the region symbol; the LINK field must contain the extent of the region. The first example on page 74 is the only acceptable format for type 2 cards.

TYPE 3 CARDS (page 75), if they appear, must follow any type 2 cards to which they refer. For Q=1, the number of words set aside for a print line may vary from 1 to 119. One word is required for each position in the print line. The word at SYMB is set to P=0,Q=0,SYMB=location of line,LINK=length of line. If, in addition, P is not zero or blank, the line is filled from the next one or two cards. The first 72 (or less) positions are filled from cols. 1-72 of the next card; if the line length is greater than 72 positions, the remaining positions are filled from cols. 1-47 of the second following card. Q=2

is used to reserve a block of storage for binary IPL-V routines written in machine language. LINK specifies the extent of the block; the starting position of the block will depend upon the number of locations previously assigned by type 2 or type 3 cards. If this is the first card to reserve space, the origin will be at 100_8. Q=3 is used to reserve a buffer for routines to be stored in auxiliary storage. LINK specifies the extent of the block; the starting position will again depend upon previous assignments.

TYPE 5 CARDS do not have the full flexibility as specified on pages 77-78. P must be zero or blank for standard IPL format, or 3 for absolute column binary format. Q is interpreted as described; however, there is a maximum of 150 local symbols and a maximum of 100 internal symbolics that may be used per routine or data list structure.

SYMB does not control the input unit, and is disregarded unless regional.

LINK character d is ignored; the only output from the assembly process is the listing on SYSPOT, as controlled by character c.

TYPE 6 AND 7 CARDS are treated identically, as only slow auxiliary routine storage is provided. If either P or Q is not zero or blank, they are treated as type 5 cards (i.e., data and binary programs may not be loaded to auxiliary storage.) If Q is zero or blank, the following routines are loaded into the auxiliary buffer, which must have been reserved by a type 3, Q=3 card. Each routine is then written on tape SYSMIT (following the record which was used to save memory). The head of a routine loaded to auxiliary is set to P=0,Q=7,SYMB= number of this routine on the auxiliary tape. Routines placed on auxiliary storage are automatically called in by the interpreter for execution and are automatically saved and reloaded by J166 and the reload processes.

NOTE: TYPE 1 CARDS should not appear in the middle of a program or data list following a program or data card with a blank SYMB or LINK. This will lead to an error which is not detected at assembly time. (The blank LINK or SYMB is treated as zero.)

ABSOLUTE BINARY INPUT. A type 5 card with P=3 indicates that a block of absolute, column binary cards follows. These must load into the space reserved by the single type 3, Q=2 card at the beginning of the program. The last binary card must be a transfer card; the address is ignored, however. After

the binary cards have been read, the symbols HO, JO, and WO are stored in the address parts of the first three cells of the reserved block. This permits the programmer to refer to these regions conveniently when coding machine language routines. If no space has been reserved for machine language routines, the cards will still be loaded. This feature may be used to load into unused available space; care must be taken to terminate available space or to link in around the routines so loaded. When this option is used, the symbols HO, JO, and WO are placed in the address parts of cells -3, -2, and -1.

ALTERNATIVE INPUT UNIT. Provision has been included for reading input from tape SYSAR1 in BCD format if desired. Sense indicator bit 34 controls this function. If bit 34 is TAPE when the program is first loaded, the IPL-V loader will initially read from tape SYSAR1 in BCD. The order and format of cards on this tape must be identical to that used for on-line reading of cards. When an end-of-file is encountered on this tape, the IPL-V loader will attempt to read from tape SYSPIT. Printing of the assembly of tape SYSAR1 may be suppressed by setting bit 33 to NOLIST; in this case only the header cards from SYSAR1 will print. If bit 33 is LIST, printing control is from character c of the header cards on the tape.

EXAMPLE. A short IPL-V program is to be loaded and executed. The deck should be set up exactly as described and written on SYSPIT in BCD. Bit 34 should be NOTAPE. Execution of the IPL-V calling sequence will cause the program to be loaded and executed as described.

EXAMPLE. A long IPL-V program is to be loaded and executed. A BCD tape should be prepared off-line; the deck set-up should be exactly as described, except that the final type 5 card should not be written on the tape. The data is followed by an end-of-file on this tape. To load and execute, this tape is mounted on unit SYSAR1. The final type 5 card is loaded on SYSPIT, with bit 34=TAPE, bit 35=LOAD. Loading and execution will take place as described.

For subsequent runs, with routines to be changed or new data added, the same tape SYSAR1 may be mounted, and the corrected routines and data loaded on SYSPIT before the final type 5 card. Since the changed routines or data are read from SYSPIT after tape SYSAR1 has been read, the corrections will be made before interpretation begins.

INPROCESS LOADING (J165), Page 80

Any restrictions that apply to initial loading also apply to inprocess loading. Note that type 2 and type 3 cards may not be loaded. New routines may be loaded to auxiliary storage, as may new copies of old routines.

Unused regionals were returned to the end of available space at the end of the initial loading process. If new regionals are required when loading more routines and data with J165, they should be reserved by type 3 cards at initial load. The locations of unused regional symbols returned to available space are still referenced by the internal symbol table, however, and an error will not be detected if J165 loads into them. All such references will be to cells on available space; this will result in an improperly constructed available space list and will usually lead to errors when the program is running.

SAVE FOR RESTART, Page 81

J166 saves on the tape specified by the integer data term named (0). If this data term is 0, memory is saved on tape SYSAR3. Other allowable tapes are specified as follows:

1=SYSAR1	6=SYSBR1
2=SYSAR2	7=SYSBR2
3=SYSAR3	8=SYSBR3
4=SYSAR4	9=SYSBR4
5=SYSAR5	10=SYSBR5

Other values of (0) will cause an error, as will reference to an unassigned tape. All of memory, plus any auxiliary routines or auxiliary data used, is saved on tape, and the program continues from that point. The tape is rewound after writing, so that multiple occurrences of J166 with the same input will cause the tape to be rewritten. Bit 31 of CNTRL (above) should be DUMP if J166 is to be used.

J167 is not available.

RELOADING. Reloading is from the tape written by J166. The tape should be mounted on tape SYSAR2; bit 35=RELOAD is used to initiate the reload process. After reloading, control is returned to exactly the same place in the execution of J166 as when memory was saved.

EXAMPLE. A program is to be loaded, saved immediately for subsequent restart, and the program is then to be executed.

On reloading, it is desired to make changes or corrections before execution.

Include a one-word routine in the program as follows:

```
NAME      P Q   SYMB  LINK
RO        1 0   9-1
                J166  J165
9-1     + 0 1                0
```

Place two type 5 termination cards at the end of the deck; the first with a SYMB of RO, the second naming the start of the program; bit 35=LOAD, bit 31=DUMP. Operation is then as follows: The program is read in and assembled, and interpretation begins with instruction RO. J166 will save memory on SYSAR3, and the program will continue with the interpretation of the instruction, which links to J165. J165 will read more routines and data, including termination cards. Since the only card remaining is a termination card, interpretation will begin with the first instruction of the program.

To reload, place the corrected routines (including any corrections to routines on auxiliary storage) or data lists on SYSPIT, preceded by the proper type 5, 6, or 7 header cards, and followed with the same two termination cards. The tape written on the first pass is mounted on tape SYSAR2. Bit 35=RELOAD, bit 34=NOTAPE, bit 31=DUMP. Immediately after reload, interpretation will continue at RO; and J165 will read the corrections into memory. The first termination card will repeat the save process on SYSAR3; the second will initiate the interpretation of the corrected program.

NOTE: The same procedure may be used when the initial program load is from SYSAR1 in BCD. Bit 34 should be NOTAPE on reload, however, so that J165 will be prepared to read corrections from SYSPIT rather than SYSAR1.

TAPE UNIT SUMMARY

TAPE SYSPIT. BCD input tape (never rewound).

TAPE SYSPOT. BCD output tape (never rewound; no end-of-files).

TAPES SYSES1-2. Auxiliary storage for data list structures (J105, J107, J108).

TAPE SYSAR1. Alternate BCD input tape (under control of bit 34).

TAPE SYSAR2. Reload tape (under control of bit 35).

TAPE SYSMIT. Auxiliary routines, if loaded, are written following the record which is used to save memory on entry to IPL-V.

TAPE SYSAR3. Normal save-for-restart tape (written by J166).

TAPES SYSAR1-5, SYSBR1-5. As used by J150, J151, J152, J153, J155, J166. If used by other than J166, and end-of-file is written at end of program.

OPERATION OF THE INTERPRETER

Interpretation begins within the system by interpreting the instruction

000 Xxx J7,

where Xxx is the routine named on the final type 5 termination card. Interpretation will continue until the J7 is linked to, at which time the program will print PROGRAM RAN TO COMPLETION on the output tape. Other terminations will occur if the programmer executes the routine J7, or if the interpreter or a primitive process detects any one of a number of errors. In case of such a termination, the reason will be printed on the output tape, followed by an IPL-V POST MORTEM. The post mortem consists of a print-out of the lists named on the list named by system cell W23. Initially this list names all the system cells, H0 through W29, plus 13 cells containing information about J105-J107 usage (see auxiliary data storage). In addition, the current print line, 1W24, is printed. The programmer may replace this list with another list, or add symbols to this list. Control is returned to 29,4 of the IPL-V calling sequence after execution of an IPL-V program.

ERROR TRAP, J170, Page 82

The error trap function is not provided; see above for error action.

SYSTEM CELLS, Page 35

Exceptions and additions are as follows:

H0 initially contains the symbol 0.

H2 will initially be about 24,000 cells for a 32K machine.

H3 is set to 1 by the initial loader.

H4 is not used, as no auxiliary storage is provided.

H5 is initially set to J4(+).

H6 is used by machine language recursive routines to hold index 4.

H7 is a data term which is set to the current level. It is set to 0 by the initial loader, and by J165 if cards are read.

H8 is the hide-out cell used by the generator processes.

H9, H10, H11, and H12 are generator-working cells.

W0 ... W9 initially contain the symbol 0.

W10 initially names a data term for random number generation.

W12 ... W15 initially contain J0.

W16 ... W19 are not used.

W20 initially names a data term equal to 0.

W21 initially names a data term equal to 1.

W22 initially names a data term equal to 1.

W23 initially names a list of the system cells.

W24 initially names a 120-word internal buffer for line printing.

W25 initially names a data term equal to 1.

W26 ... W28 are not used. See above.

W29 initially contains the symbol 0.

GENERATOR PROCESSES, Pages 40-43

The generator trace context and level context are handled as described, provided that generators are always executed from the left. Both generators and subprocesses may be on auxiliary storage; subprocesses must have regional names if on auxiliary.

J102, Page 52

Not available. May be coded as described in Part I of the manual, page 82, if desired.

AUXILIARY STORAGE FOR DATA LIST STRUCTURES, Page 52

J106 is not available, as only slow storage is provided.

A data list structure is filed as a series of binary records on the current auxiliary tape unit; the address of this unit (SYSES1 or SYSES2) is controlled automatically. The head of a list structure filed in auxiliary storage contains this address in SYMB, the number of the first record of the structure on the tape appears in LINK, and Q=7. When a structure is moved to memory from tape, the records that it occupied on tape are no longer available. A tally of these dead records is kept; and when the ratio of inactive to active records exceeds 1/4, the tape is automatically compacted onto the other unit used for auxiliary storage, eliminating the inactive structures. The new unit becomes the current auxiliary tape unit. Storage for data list structures thus alternates between tapes SYSES1 and SYSES2. (Note: Inactive data list structures at the end of the tape are written over by J107, and are not counted in the active/inactive ratio.)

The last 13 cells printed by the post mortem contain pertinent information about auxiliary data storage usage. These cells have internal addresses and are, in order:

1. Number of times compacting occurred.
2. Total number of records eliminated by compacting.
3. Number of times next write position was updated.
4. Number of records eliminated by updating.
5. Number of J72 usages involving auxiliary.
6. Number of J74 usages involving auxiliary.
7. Number of J105 usages (J100 and J101 use J105 automatically, if necessary).
8. Number of J107 usages.
9. Total number of words written by J107.
10. Total number of records written by J107. (Records vary in length up to 125 words.)
11. Record number of next write position.
12. Current position of auxiliary tape.

13. List of structures currently on auxiliary, each followed by the number of records occupied by the structure. The first entry represents the dead records and is followed by the total number of such records.

A new primitive, J139, has been added to enable the programmer to detect when available space has dropped below a specified minimum without counting the entire H2 list:

J139 TEST IF THE NUMBER OF CELLS ON LIST (1) EXCEEDS (0). (0) is assumed to be an integer data term. List (1) is counted until the count exceeds (0), at which point H5 is set +. If list (1) is exhausted before this happens, H5 is set -.

ARITHMETIC PROCESSES, Pages 55-57

J110, J111, J112, J113, J115, and J116 admit integer, floating, and octal data terms only. The type of the result (0) of J110-J113 is as follows:

(1)	(2)	(0)
Floating	Any	Floating
Any	Floating	Floating
Integer	Integer	Integer
Integer	Octal	Integer
Octal	Integer	Octal
Octal	Octal	Octal

J125 admits integer, floating, and octal data terms only.

J128 will translate integer to floating, floating to integer, octal to integer, and integer to octal. In addition, floating and BCD data terms are "translated" to octal by setting P=3, and octal is "translated" to BCD by setting P=2. All other translations (BCD to integer, octal or BCD to floating, and integer or floating to BCD) are illegal.

INPUT-OUTPUT CONVENTIONS, Pages 61-64

Input and output units are assigned as described above. Processes J140-J146 are not available.

MONITOR POINTS AND TRACING, Pages 64-67

No console signals are provided. Terminate for restart is not available. The program may be terminated by a manual TRA SYSERR or SYSERR-2. Sense bit 32=TRACE signals full trace.

Trace speed is approximately 80 IPL-V instructions per second. Bit 32=NTRACE gives normal tracing modes.

No provision is made for suppressing tracing.

On the printed trace line, if S or (0) is a local symbol, this is indicated by an asterisk following the symbol; e.g., 2765*. The designated symbol S is printed only if Q=1 or 2.

PRINT PROCESSES, Pages 69-72

The output unit for J150, J151, J152, J153, and J155 is controlled by the integer data term named in W20. If this data term is 0, the output unit is SYSPOT. Other units are specified as follows:

1 or 11 = SYSAR1	6 or 16 = SYSBR1
2 or 12 = SYSAR2	7 or 17 = SYSBR2
3 or 13 = SYSAR3	8 or 18 = SYSBR3
4 or 14 = SYSAR4	9 or 19 = SYSBR4
5 or 15 = SYSAR5	10 or 20 = SYSBR5

If the data term is 10 or less, the print line 1W24 is written on the specified tape for subsequent off-line printing; i.e., the first character is the spacing character as determined from 1W22. If the data term is greater than 10, the referenced line is written on the specified tape for subsequent off-line punching; i.e., the first 80 characters are written out without a spacing control character. All tapes written by these processes are written with an end-of-file at the end of the IPL-V run.

If a list structure with a local name is printed by J150, the converted name will be 9-0. Sublists of any list structure are always numbered consecutively starting with 9-1.

Data terms are always printed in the format given on page 71, with the exception of floating point numbers. The format for these is "-.123456-12"; plus signs are not printed. The field length is thus either 10 or 11 columns.

When local symbols are printed (except in the case of those translated by J150), they are followed by an asterisk.

Cell W24 initially names an internal print buffer of 120 cells (one character per cell) length. Other print lines may be assigned by type 3, Q=1 cards at initial loading.

J158 and J159 leave 1W25 pointing to the first column to the left of the entered information.

Revised 11/17/61

709/7090 IPL-V

PROGRAM DESCRIPTION

This write-up is in two parts: the first describing the nature of the IPL-V system as a subroutine, the second including deviations from and additions to the basic IPL-V manual.

PART I

CALLING SEQUENCE

```
CS - 1    =  LDI CNTRL
CS        =  TSX AO,4
CS + 1    =  ...
CS + 2    =  PZE SYSPIT,,SYSMIT
CS + 3    =  PZE SYSPOT,,PAGES
CS + 4    =  PZE SYSES1,,SYSES2
CS + 5    =  ...
CS + 6    =  PZE ...,,SYSERR
                 .
                 .
                 .
CS + 13   =  PZE SYSORG,,...
                 .
                 .
                 .
CS + 29   =  Return
                 .
                 .
                 .
CS + 37   =  PZE SYSTSL,,...
```

The above calling sequence is that used within the MockDonald operating system; hence its strange format. Words or parts of words indicated by "..." are not significant.

CONTROL BITS

Bits 30-35 of CNTRL are placed in the sense indicator register for conveying information to IPL-V as to the nature of the run being made. Immediately below is given the meaning of these bits; their interpretation is explained in Part 2.

```
Bit 35   0    LOAD
Bit 35   1    RELOAD
```

Bit 34	0	NOTAPE
Bit 34	1	TAPE
Bit 33	0	LIST
Bit 33	1	NOLIST
Bit 31	0	NODUMP
Bit 31	1	DUMP

Bit 30	Bit 32	
0	0	NTRACE (Suppress all tracing.)
0	1	TRACE (Trace everything.)
1	0	STRACE or blank (Selective trace ... normal.)
1	1	SNAP (Execute 1W12 and 1W13 instead of tracing marked routines.)

CALLING SEQUENCE INFORMATION

SYSPIT, SYSMIT, SYSMOT, SYSES1, SYSES2 are the locations of tape control words. These words must contain the address of the physical unit associated with the symbolic tape. (E.g., if SYSPIT is on physical unit B2, the control word would have an octal address of 02202.) Only the first three tapes are necessary for using the system. If the address is zero, the unit is interpreted as being unavailable.

SYSTSL is the location of the tape status list for reserved tapes. It is arranged such that

```
AXT   CHANNEL,2     (1 for Ch. A, 2 for Ch. B, etc.)
AXT   UNIT,1        (3 for Unit 3, etc.)
CLA*  SYSTSL,2
```

will put the address of the corresponding tape control word into the decrement of the accumulator. Tapes SYSAR1, 2, 3, 4, 5, SYSBR1, 2, 3, 4, and 5 are interrogated. Again, a zero address is interpreted as meaning that the corresponding unit is unavailable.

SYSERR and SYSERR-2 are the locations to which the operator is to transfer in case of program error, loop, too much time, etc. Both SYSERR and SYSERR-2 are replaced with a transfer to the recovery routine for IPL-V; the contents of these cells is saved and restored. They must lie in the range 00001 - 00077_8. The entry from SYSERR attempts to give an IPL-V post mortem before exit; the entry from SYSERR-2 causes an immediate exit from IPL-V.

SYSORG is the name of a cell containing the location of the first word following the system in control. Memory up to location C(SYSORG) is left undisturbed by the execution of IPL-V (with the exception of cell zero).

PAGES is the location of a cell containing the approximate number of pages of output expected. This number is multiplied by 64 to give a line count. (A line count of zero is replaced with a count of 6000.) An error condition is sensed when the output (from assembly, tracing, J150, J151, J152, J153, or J155, printing or punching) exceeds the expected amount. The line count is then reset so that a post mortem may be printed.

OPERATION OF THE IPL-V EXECUTIVE ROUTINE, AO

1. All traps are disabled.
2. The tape control words are interpreted, and a table of tape information is constructed.
3. Memory is saved on tape SYSMIT from locations 100_8-C(SYSØRG).
4. If the RELOAD option is called, memory is reloaded from SYSAR2.
5. The remaining sense indicator bits are interpreted.
6. Control is passed to the IPL-V system interpreter.
7. At the conclusion of the IPL-V run (program finished, error, or manual transfer to SYSERR-2 or SYSERR), memory is restored from tape SYSMIT and control returned to word 29 of the calling sequence.

NOTES

Although the system has been organized so as to work within the MockDonald monitor system, it should be emphasized that it is in no way dependent upon any of the routines of that system. Only the tape control words of that system need be reproduced (in the format explained above) to gain complete independence of that system.

All input and output routines are contained within the IPL-V system itself. Data is read, in BCD, from tape SYSPIT; output is written in BCD on tape SYSPOT. Neither tape is rewound, nor is an end-of-file ever written on SYSPOT. Tape SYSMIT is used for saving core and for auxiliary routine storage. SYSAR1, 2, 3, 4, 5, SYSBR1, 2, 3, 4, and 5 may be used by the IPL-V interpreted program. See following section for details.

EXAMPLE

Following is a complete calling sequence requiring no information from the MockDonald system:

```
START    LDI  CNTRL
         TSX  AO,4
         PZE
         PZE  SYSPIT,,SYSMIT
         PZE  SYSPOT,,PAGES
         PZE  SYSES1,,SYSES2
         PZE
         PZE  0,0,3  (Manual Error Transfer to 3₈ or 1₈.)
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE  SYSORG
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         TRA  Out
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE
         PZE  SYSTSL
CNTRL    OCT  23     RELOAD,TAPE,LIST,NTRACE,DUMP  (Example)
SYSPIT   OCT  2201   Unit B1 for SYSPIT
SYSMIT   OCT  2202   Unit B2 for SYSMIT
SYSPOT   OCT  2203   Unit B3 for SYSPOT
SYSES1   OCT  1204   Unit A4 for SYSES1 (If Aux. Data Storage Used)
SYSES2   OCT  1205   Unit A5 for SYSES2 (If Aux. Data Storage Used)
SYSAR1   OCT  1201   Unit A1 for SYSAR1
SYSAR2   OCT  1202   Unit A2 for SYSAR2
```

```
SYSAR3   OCT  1203    Unit A3 for SYSAR3
SYSAR4   PZE          SYSAR4 Not Used
SYSAR5   PZE          SYSAR5 Not Used
SYSBR1   PZE          SYSBR1 Not Used
SYSBR2   PZE          SYSBR2 Not Used
SYSBR3   PZE          SYSBR3 Not Used
SYSBR4   OCT  2204    Unit B4 for SYSBR4 (If Used)
SYSBR5   OCT  2205    Unit B5 for SYSBR5 (If Used)
SYSORG   PZE  16384   Save lower half of core.
         PZE  CHNLB,1
         PZE  CHNLA,1
SYSTSL   PZE          Tape Status List
         PZE  0,,SYSAR5
         PZE  0,,SYSAR4
         PZE  0,,SYSAR3
         PZE  0,,SYSAR2
         PZE  0,,SYSAR1
CHNLA    PZE
         PZE  0,,SYSBR5
         PZE  0,,SYSBR4
         PZE  0,,SYSBR3
         PZE  0,,SYSBR2
         PZE  0,,SYSBR1
CHNLB    PZE
PAGES    DEC  200     Approximate Number of Pages (Example)
```

PART II

This write-up includes only deviations from and additions to the basic IPL-V manual, INFORMATION PROCESSING LANGUAGE - V MANUAL, Section II, Programmers' Reference Manual, A. Newell, Editor, Prentice-Hall, Inc., 1961. Familiarity with this manual is assumed; references are to sections and page numbers in this manual.

INITIAL LOADING, Page 214

This version is not set up to run multiple IPL-V jobs. Each run must begin with the execution of the calling sequence for IPL-V. References to bits 30-35 refer to the setting of the sense indicator register on entry to IPL-V. (See Part I.) The IPL-V program, set up as described on page 221, is assumed to be on tape SYSPIT, in BCD. This tape is not rewound prior to reading.

All assembly, tracing, error, and post mortem output is off-line, on tape SYSPOT. This tape is never rewound, nor is an end-of-file written. This permits stacking of the IPL-V output with other jobs adhering to the same output conventions. Output from J150, J151, J152, J153, and J155 may be on reserved tapes if so specified (see below). An end-of-file is written on reserved tapes so used, at the conclusion of the program.

A TYPE 9 CARD (page 220) is treated as a type 1 comment card, as no multiple job provisions are included.

TYPE 2 CARDS do not have the full flexibility described on pages 215-216. Only the NAME and LINK fields are read; the first character of the NAME field must be the region symbol; the LINK field must contain the extent of the region. The first example on page 216 is the only acceptable format for type 2 cards.

TYPE 3 CARDS (page 216), if they appear, must follow any type 2 cards to which they refer. For Q=1, the number of words set aside for a print line may vary from 1 to 119. One word is required for each position in the print line. The word at SYMB is set to P=0,Q=0,SYMB=location of line,LINK=length of line. If, in addition, P is not zero or blank, the line is filled from the next one or two cards. The first 72 (or less) positions are filled from cols. 1-72 of the next card; if the line length is greater than 72 positions, the remaining positions are filled from cols. 1-47 of the second following card. Q=2

is used to reserve a block of storage for binary IPL-V routines written in machine language. LINK specifies the extent of the block; the starting position of the block will depend upon the number of locations previously assigned by type 2 or type 3 cards. If this is the first card to reserve space, the origin will be at 100$_8$. Q=3 is used to reserve a buffer for routines to be stored in auxiliary storage. LINK specifies the extent of the block; the starting position will again depend upon previous assignments.

TYPE 5 CARDS do not have the full flexibility as specified on pp. 217-220. P must be zero or blank for standard IPL format, or 3 for absolute column binary format. Q is interpreted as described; however, there is a maximum of 150 local symbols and a maximum of 100 internal symbolics that may be used per routine or data list structure.

SYMB does not control the input unit, and is disregarded unless regional.

LINK character d is ignored; the only output from the assembly process is the listing on SYSPOT, as controlled by character c.

TYPE 6 AND 7 CARDS are treated identically, as only slow auxiliary routine storage is provided. If either P or Q is not zero or blank, they are treated as type 5 cards (i.e., data and binary programs may not be loaded to auxiliary storage.) If Q is zero or blank, the following routines are loaded into the auxiliary buffer, which must have been reserved by a type 3, Q=3 card. Each routine is then written on tape SYSMIT (following the record which was used to save memory). The head of a routine loaded to auxiliary is set to P=0,Q=7,SYMB= number of this routine on the auxiliary tape. Routines placed on auxiliary storage are automatically called in by the interpreter for execution and are automatically saved and reloaded by J166 and the reload processes.

NOTE: TYPE 1 CARDS should not appear in the middle of a program or data list following a program or data card with a blank SYMB or LINK. This will lead to an error which is not detected at assembly time. (The blank LINK or SYMB is treated as zero.)

ABSOLUTE BINARY INPUT. A type 5 card with P=3 indicates that a block of absolute, column binary cards follows. These must load into the space reserved by the single type 3, Q=2 card at the beginning of the program. The last binary card must be a transfer card; the address is ignored, however. After

the binary cards have been read, the symbols HO, JO, and WO are stored in the address parts of the first three cells of the reserved block. This permits the programmer to refer to these regions conveniently when coding machine language routines. If no space has been reserved for machine language routines, the cards will still be loaded. This feature may be used to load into unused available space; care must be taken to terminate available space or to link in around the routines so loaded. When this option is used, the symbols HO, JO, and WO are placed in the address parts of cells -3, -2, and -1.

ALTERNATIVE INPUT UNIT. Provision has been included for reading input from tape SYSAR1 in BCD format if desired. Sense indicator bit 34 controls this function. If bit 34 is TAPE when the program is first loaded, the IPL-V loader will initially read from tape SYSAR1 in BCD. The order and format of cards on this tape must be identical to that used for on-line reading of cards. When an end-of-file is encountered on this tape, the IPL-V loader will attempt to read from tape SYSPIT. Printing of the assembly of tape SYSAR1 may be suppressed by setting bit 33 to NOLIST; in this case only the header cards from SYSAR1 will print. If bit 33 is LIST, printing control is from character c of the header cards on the tape.

EXAMPLE. A short IPL-V program is to be loaded and executed. The deck should be set up exactly as described and written on SYSPIT in BCD. Bit 34 should be NOTAPE. Execution of the IPL-V calling sequence will cause the program to be loaded and executed as described.

EXAMPLE. A long IPL-V program is to be loaded and executed. A BCD tape should be prepared off-line; the deck set-up should be exactly as described, except that the final type 5 card should not be written on the tape. The data is followed by an end-of-file on this tape. To load and execute, this tape is mounted on unit SYSAR1. The final type 5 card is loaded on SYSPIT, with bit 34=TAPE, bit 35=LOAD. Loading and execution will take place as described.

For subsequent runs, with routines to be changed or new data added, the same tape SYSAR1 may be mounted, and the corrected routines and data loaded on SYSPIT before the final type 5 card. Since the changed routines or data are read from SYSPIT after tape SYSAR1 has been read, the corrections will be made before interpretation begins.

INPROCESS LOADING (J165), Page 222

Any restrictions that apply to initial loading also apply to inprocess loading. Note that type 2 and type 3 cards may not be loaded. New routines may be loaded to auxiliary storage, as may new copies of old routines.

Unused regionals were returned to the end of available space at the end of the initial loading process. If new regionals are required when loading more routines and data with J165, they should be reserved by type 3 cards at initial load. The locations of unused regional symbols returned to available space are still referenced by the internal symbol table, however, and an error will not be detected if J165 loads into them. All such references will be to cells on available space; this will result in an improperly constructed available space list and will usually lead to errors when the program is running.

SAVE FOR RESTART, Page 223

J166 saves on the tape specified by the integer data term named (0). If this data term is 0, memory is saved on tape SYSAR3. Other allowable tapes are specified as follows:

1=SYSAR1	6=SYSBR1
2=SYSAR2	7=SYSBR2
3=SYSAR3	8=SYSBR3
4=SYSAR4	9=SYSBR4
5=SYSAR5	10=SYSBR5

Other values of (0) will cause an error, as will reference to an unassigned tape. All of memory, plus any auxiliary routines or auxiliary data used, is saved on tape, and the program continues from that point. The tape is rewound after writing, so that multiple occurrences of J166 with the same input will cause the tape to be rewritten. Bit 31 of CNTRL (above) should be DUMP if J166 is to be used.

J167 is not available.

RELOADING. Reloading is from the tape written by J166. The tape should be mounted on tape SYSAR2; bit 35=RELOAD is used to initiate the reload process. After reloading, control is returned to exactly the same place in the execution of J166 as when memory was saved.

EXAMPLE. A program is to be loaded, saved immediately for subsequent restart, and the program is then to be executed.

On reloading, it is desired to make changes or corrections before execution.

Include a one-word routine in the program as follows:

```
NAME      P Q   SYMB  LINK
RO        1 0   9-1
                J166  J165
9-1    + 0 1                0
```

Place two type 5 termination cards at the end of the deck; the first with a SYMB of RO, the second naming the start of the program; bit 35=LOAD, bit 31=DUMP. Operation is then as follows: The program is read in and assembled, and interpretation begins with instruction RO. J166 will save memory on SYSAR3, and the program will continue with the interpretation of the instruction, which links to J165. J165 will read more routines and data, including termination cards. Since the only card remaining is a termination card, interpretation will begin with the first instruction of the program.

To reload, place the corrected routines (including any corrections to routines on auxiliary storage) or data lists on SYSPIT, preceded by the proper type 5, 6, or 7 header cards, and followed with the same two termination cards. The tape written on the first pass is mounted on tape SYSAR2. Bit 35=RELOAD, bit 34=NOTAPE, bit 31=DUMP. Immediately after reload, interpretation will continue at RO; and J165 will read the corrections into memory. The first termination card will repeat the save process on SYSAR3; the second will initiate the interpretation of the corrected program.

NOTE: The same procedure may be used when the initial program load is from SYSAR1 in BCD. Bit 34 should be NOTAPE on reload, however, so that J165 will be prepared to read corrections from SYSPIT rather than SYSAR1.

TAPE UNIT SUMMARY

TAPE SYSPIT. BCD input tape (never rewound).

TAPE SYSPOT. BCD output tape (never rewound; no end-of-files).

TAPES SYSES1-2. Auxiliary storage for data list structures (J105, J107, J108).

TAPE SYSAR1. Alternate BCD input tape (under control of bit 34).

TAPE SYSAR2. Reload tape (under control of bit 35).

TAPE SYSMIT. Auxiliary routines, if loaded, are written following the record which is used to save memory on entry to IPL-V.

TAPE SYSAR3. Normal save-for-restart tape (written by J166).

TAPES SYSAR1-5, SYSBR1-5. As used by J150, J151, J152, J153, J155, J166. If used by other than J166, and end-of-file is written at end of program.

OPERATION OF THE INTERPRETER

Interpretation begins within the system by interpreting the instruction

```
000  Xxx  J7,
```

where Xxx is the routine named on the final type 5 termination card. Interpretation will continue until the J7 is linked to, at which time the program will print PROGRAM RAN TO COMPLETION on the output tape. Other terminations will occur if the programmer executes the routine J7, or if the interpreter or a primitive process detects any one of a number of errors. In case of such a termination, the reason will be printed on the output tape, followed by an IPL-V POST MORTEM. The post mortem consists of a print-out of the lists named on the list named by system cell W23. Initially this list names all the system cells, H0 through W29, plus 13 cells containing information about J105-J107 usage (see auxiliary data storage). In addition, the current print line, 1W24, is printed. The programmer may replace this list with another list, or add symbols to this list. Control is returned to 29,4 of the IPL-V calling sequence after execution of an IPL-V program.

ERROR TRAP, J170, Page 224

The error trap function is not provided; see above for error action.

SYSTEM CELLS, Page 177

Exceptions and additions are as follows:

H0 initially contains the symbol 0.

H2 will initially be about 24,000 cells for a 32K machine.

H3 is set to 1 by the initial loader.

H4 records current auxiliary routine.

H5 is initially set to J4(+).

H6 is used by machine language recursive routines to hold index 4.

H7 is a data term which is set to the current level. It is set to 0 by the initial loader, and by J165 if cards are read.

H8 is the hide-out cell used by the generator processes.

H9, H10, H11, and H12 are generator-working cells.

W0 ... W9 initially contain the symbol 0.

W10 initially names a data term for random number generation.

W12 ... W15 initially contain J0.

W16, W17, W19 are not used.

W18 controls the input unit; 0=SYSPIT, non-zero=SYSAR1.

W20 initially names a data term equal to 0.

W21 initially names a data term equal to 1.

W22 initially names a data term equal to 1.

W23 initially names a list of the system cells.

W24 initially names a 120-word internal buffer for line printing.

W25 initially names a data term equal to 1.

W26 ... W28 are not used. See above.

W29 initially contains the symbol 0.

GENERATOR PROCESSES, Pages 182-185

The generator trace context and level context are handled as described, provided that generators are always executed from the left. Both generators and subprocesses may be on auxiliary storage; subprocesses must have regional names if on auxiliary.

J102, Page 194

Not available. May be coded as described in Part I of the Manual, page 82, if desired.

AUXILIARY STORAGE FOR DATA LIST STRUCTURES, Page 194

J106 is not available, as only slow storage is provided.

A data list structure is filed as a series of binary records on the current auxiliary tape unit; the address of this unit (SYSES1 or SYSES2) is controlled automatically. The head of a list structure filed in auxiliary storage contains this address in SYMB, the number of the first record of the structure on the tape appears in LINK, and Q=7. When a structure is moved to memory from tape, the records that it occupied on tape are no longer available. A tally of these dead records is kept; and when the ratio of inactive to active records exceeds 1/4, the tape is automatically compacted onto the other unit used for auxiliary storage, eliminating the inactive structures. The new unit becomes the current auxiliary tape unit. Storage for data list structures thus alternates between tapes SYSES1 and SYSES2. (Note: Inactive data list structures at the end of the tape are written over by J107, and are not counted in the active/inactive ratio.)

The last 13 cells printed by the post mortem contain pertinent information about auxiliary data storage usage. These cells have internal addresses and are, in order:

1. Number of times compacting occurred.
2. Total number of records eliminated by compacting.
3. Number of times next write position was updated.
4. Number of records eliminated by updating.
5. Number of J72 usages involving auxiliary.
6. Number of J74 usages involving auxiliary.
7. Number of J105 usages (J100 and J101 use J105 automatically, if necessary).
8. Number of J107 usages.
9. Total number of words written by J107.
10. Total number of records written by J107. (Records vary in length up to 125 words.)
11. Record number of next write position.
12. Current position of auxiliary tape.

13. List of structures currently on auxiliary, each followed by the number of records occupied by the structure. The first entry represents the dead records and is followed by the total number of such records.

A new primitive, J139, has been added to enable the programmer to detect when available space has dropped below a specified minimum without counting the entire H2 list:

J139 TEST IF THE NUMBER OF CELLS ON LIST (1) EXCEEDS (0). (0) is assumed to be an integer data term. List (1) is counted until the count exceeds (0), at which point H5 is set +. If list (1) is exhausted before this happens, H5 is set -.

ARITHMETIC PROCESSES, Pages 197-199

J110, J111, J112, J113, J115, and J116 admit integer, floating, and octal data terms only. The type of the result (0) of J110-J113 is as follows:

(1)	(2)	(0)
Floating	Any	Floating
Any	Floating	Floating
Integer	Integer	Integer
Integer	Octal	Integer
Octal	Integer	Octal
Octal	Octal	Octal

J125 admits integer, floating, and octal data terms only.

J128 will translate integer to floating, floating to integer, octal to integer, and integer to octal. In addition, floating and BCD data terms are "translated" to octal by setting P=3, and octal is "translated" to BCD by setting P=2. All other translations (BCD to integer, octal or BCD to floating, and integer or floating to BCD) are illegal.

INPUT-OUTPUT CONVENTIONS, Pages 203-206

Input and output units are assigned as described above. Processes J140-J146 are not available.

MONITOR POINTS AND TRACING, Page 206-209

Sense switch 5 is provided as a console signal to terminate the job. When SW 5 is depressed, the interpreter immediately executes the routine named by W14, then gives a post-mortem dump and quits. The programmer may provide any program for

W14, including J166 as a means of terminating for restart. A manual TRA SYSERR also terminates a job with a post-mortem, while TRA SYSERR-2 terminates without one.

Indicator bits 30 and 32 control tracing. 00=NTRACE=suppress all tracing; 01=TRACE=trace everything; 10=STRACE (or blank)= selective tracing (normal); 11=SNAP=execute the programs in W12 and W13 for routines marked with Q=3 but omit the detailed trace printout.

PRINT PROCESSES, Pages 211-214

The output unit for J150, J151, J152, J153, and J155 is controlled by the integer data term named in W20. If this data term is 0, the output unit is SYSPOT. Other units are specified as follows:

1 or 11 = SYSAR1	6 or 16 = SYSBR1
2 or 12 = SYSAR2	7 or 17 = SYSBR2
3 or 13 = SYSAR3	8 or 18 = SYSBR3
4 or 14 = SYSAR4	9 or 19 = SYSBR4
5 or 15 = SYSAR5	10 or 20 = SYSBR5

If the data term is 10 or less, the print line 1W24 is written on the specified tape for subsequent off-line printing; i.e., the first character is the spacing character as determined from 1W22. If the data term is greater than 10, the referenced line is written on the specified tape for subsequent off-line punching; i.e., the first 80 characters are written out without a spacing control character. All tapes written by these processes are written with an end-of-file at the end of the IPL-V run.

If a list structure with a local name is printed by J150, the converted name will be 9-0. Sublists of any list structure are always numbered consecutively starting with 9-1.

Data terms are always printed in the format given on page 71, with the exception of floating point numbers. The format for these is "-.123456-12"; plus signs are not printed. The field length is thus either 10 or 11 columns.

When local symbols are printed (except in the case of those translated by J150), they are followed by an asterisk.

Cell W24 initially names an internal print buffer of 120 cells (one character per cell) length. Other print lines may be assigned by type 3, Q=1 cards at initial loading.

J158 and J159 leave 1W25 pointing to the first column to the left of the entered information.

IPL-TO-MACHINE-LANGUAGE COMPILER — OPERATING INSTRUCTIONS

The compiler is designed to take a complete IPL program deck (with the exception of absolute binary input cards) and convert to symbolic machine language for input to SCAT. The binary deck produced by SCAT becomes input to IPL for actual running. The deck set-up described below should be used to go from IPL to absolute binary in one 7090 run.

DECK SET-UP: COMPILE PASS

#	JOB	6331, etc.	
#	ASSIGN	A5=SYSAR1	(Optional. The program to be compiled may be on SYSPIT, in which case this assign card should be omitted. See below.)
#	ASSIGN	A6=SYSAR2	(Tape 579 is the reload tape for the compiler, IPL version; Tape 1291 is the reload tape for the compiler, compiled into machine language.)
#	ASSIGN	B5=SYSBR1	(Utility tape. The compiler output is on this tape, which becomes the input to SCAT.)
#	IPL	RELOAD	(IPL control card.)

Type	SYMB	
5	{B0 / X3}	(IPL start card. B0 for input from SYSPIT, X3 for input from SYSAR1.)

May be on SYSAR1 or SYSPIT	[Type 2 and 3 cards for IPL program.]	(Note: The order of these cards must be preserved. See below.)
	[IPL program and data, with usual type 5 cards.]	(Program to be compiled.)
	[Type 5 start card.]	(Terminate compilation.)

#	REWIND	SYSBR1	(Rewind output tape.)
#	SCAT	NOGO,SQZ,ABS,LIST,SYSBR1	(SCAT control card.)

[blank card]

The absolute program is designed to load into available space immediately following the space reserved by type 2 and 3 cards. Since these type 2 and 3 cards control the absolute addresses which will be assigned both by SCAT and by IPL during running phase, it is essential that this deck is identical for both compile and execution passes. However, the compiler does not recognize print line cards (which follow type 3, PQ= 11 cards), so these should **NOT** be present when compiling, unless they have a 1 in the **TYPE** col, (41).

The SCAT code output by the compiler is as follows:

ORG 64	(corresponds to IPL origin)
BSS n ⋮ BSS n	(one for each type 2 or 3 card that actually assigns space, two for print line reservations.)
XXX ⋮ XXX	(SCAT code corresponding to routines or data. The original IPL code appears as commentary.)
YYY ⋮ YYY	(Subroutines required by the compiled code, EQU cards for references to the IPL system, new head for H2 list.)
END 0	(SCAT END cards)

This code should give no SCAT errors except **ORIGIN IN MONITOR.**

Note that IPL code which is "aware" of the IPL system (e.g., makes references to H1, or treats data as program or vice versa, etc.) will probably not run correctly when compiled.

Cell H6 is used by the compiled code as the machine language CIA list.

There is no advantage to compiling data list structures other than the fewer cards required when the list structure is in binary.

The speed ratio obtained by compiling an IPL code is in the neighborhood of 1.8 to 1.

DECK SET-UP: RUNNING PASS

```
#    JOB    6331, etc.

#    IPL    etc.                    (IPL control card.)
```

[Type 2 and 3 cards from compile run, in identical order.]

Type	PQ	SYMB
5	30	

[Binary cards produced by the compile pass.]

[Other IPL routines, data, absolute binary, etc.]

```
5              Xxx    (IPL start card.)
```

LINE READ PRIMITIVES

The line read primitives provide a means of reading a BCD card under control of an IPL-V program and translating selected fields into IPL symbols or data terms.

CONTROL CELLS

1W24 names the current read line. ("Read lines" and "print lines" are identical and interchangeable. Lines for either or both purposes are specified by Type 3, Q=1 cards.)

1W25 is a decimal integer data term specifying the left column of the current input field.

1W30 is a decimal integer data term specifying the size (number of columns) of the current input field.

PRIMITIVE DEFINITIONS

*J180 READ LINE. The next input record is read to line 1W24. (The record is assumed to be BCD.) Column 1 of the record is read into column 1 of the read line, and so forth. H5 is set +. If no record can be read (end of file condition), the line is not changed and H5 is set -.

*J181 INPUT LINE SYMBOL. The IPL symbol in the field starting in column 1W25 and of size 1W30, in line 1W24, is input to H0; H5 is set +. The symbol is regional if the first (leftmost) column holds a regional character; otherwise it is absolute internal. All non-numerical characters except in the first column are ignored. If the field is entirely blank, or ignored, there is no input to H0, and H5 is set -. In either case, 1W25 is incremented by the amount 1W30.

*J182 INPUT LINE DATA TERM (0). The field specified as in J181 is taken as the value of a data term; input data term (0) is set to that value and left as output (0). H5 is set +. The data type of input (0) determines the data type of the output. If the input (0) is a decimal or octal integer or BCD the read line field is interpreted as that type. Any other data type is treated as BCD. In composing BCD data terms, the field is left-justified and the full data term completed with blanks on the right

if necessary. If the specified field exceeds five columns, the rightmost five columns are taken as the field. In composing decimal and octal integer data terms, non-numerical characters are ignored; if the resulting information exceeds the capacity of the data term, the rightmost digits are retained. If the read line field is entirely blank (or non-numerical, for integer data types), (0) is cleared (to blanks for BCD, to zero for integer) and H5 is set -. In either case, 1W25 is incremented by the amount 1W30.

*J183 SET (0) TO NEXT BLANK. (0) is taken as a decimal integer data term. Line 1W24 is scanned, left to right, starting with column 1W25+1, for a blank. One is added to (0) for each column scanned, including that in which the scanned-for character ("blank" in J183) is found. (0) is left as output (0). H5 is set + if the character is found in the line, and - if it is not.

(Thus, if input (0) = 1W25, after scanning output (0) will specify the column holding the scanned-for character. If input (0) = decimal integer 0, after scanning output (0) will be the size of a field beginning in column 1W25 and delimited on the right by the next occurrence of the scanned-for character.)

*J184 SET (0) TO NEXT NON-BLANK. Same as J183, except scans for any non-blank character.

*J185 SET (1) TO NEXT OCCURRENCE OF CHARACTER (0). Same as J183, except scans for character (0), counting into decimal integer data term (1). Input (1) is left as output (0). If input (0) is a regional symbol, its region character is the character scanned for; if input (0) is internal, its last (low-order) digit is the character scanned for.

*J186 INPUT LINE CHARACTER. The character in column 1W25 of line 1W24 is input to H0; H5 is set +. If the character is numerical that internal symbol is input; if the character is non-numerical, the zeroth symbol in the region designated by the character is input. If the character is a blank, there is no input and H5 is set -. In either case, 1W25 is not advanced.

*J189 TRANSFER FIELD. The field in line 1W24, starting in column 1W25 and of size 1W30, is transferred to line (0), starting in column 1W21; H5 is set +. If the entire field cannot be transferred (line (0) is too short), as much is transferred as can be,

and H5 is set -. In either case, 1W25 is set to the last column transferred plus one.

IMPLEMENTATION

The line read primitives are coded in SCAT for use with the RAND version of IPL-V for the 7090. Since the code uses parts of the IPL-V loader and interpreter as subroutines, equivalent addresses must be specified for several of the symbols used when line read is assembled separately from the rest of the system. (See the program listing for details.)

Because an interpreter subroutine is used in converting input, an error condition (UNDEFINED SYMBOL DURING LOADING) will occur if the symbol input by J181 or J186 does not lie within a defined region.

USE OF THESE PRIMITIVES

The current deck of line read absolute binaries is assembled into a block beginning at 23000_{10}. The binaries may be loaded during IPL loading following a Type 5, P=3 card; no Type 3 primitive block reservation card is required. (Of course, available space should not have been used as high as 23000_{10} prior to this loading.) The symbolic version of these primitives may be assembled into another absolute location if so desired.

```
*                          THE FOLLOWING EQUIVALENCE CARDS (WITH THE
*                          APPROPRIATE ABSOLUTE ADDRESS) MUST BE SUPPLIED
*                          WHEN LINE READ IS ASSEMBLED SEPARATELY FROM THE
*                          IPL-V 7090 SYSTEM.  WHEN LINE READ IS BEING
*                          ASSEMBLED WITH THE IPL-V SYSTEM, THEY MUST
*                          (REPEAT, MUST) BE OMITTED.
J         BOOL     63170
H0        BOOL     62705
J3        BOOL     70774
J4        BOOL     71000
L2        BOOL     65036                READS A CARD INTO L902 BUFFER
L6        BOOL     65207                CONVERTS 5 BCD COLS. TO REG. + REL.
L7        BOOL     65262                ASSIGN CELL TO SYMBOL FROM TABLES
L902      BOOL     64472                BUFFER FOR UNPACKED BCD IMAGE
L99       BOOL     65012                SIGNAL, ZERO IF INTERNALS ABSOLUTE
P3        BOOL     67146                LOOKS UP REGIONAL SYMBOL IN TABLE
S4        BOOL     57351                PRESERVES H0
S5        BOOL     57353                RESTORES H0
W         BOOL     62722
W21       EQU      W+21
W24       EQU      W+24
W25       EQU      W+25
*                          THE PRECEDING EQUIVALENCE CARDS (WITH THE
*                          APPROPRIATE ABSOLUTE ADDRESS) MUST BE SUPPLIED
*                          WHEN LINE READ IS ASSEMBLED SEPARATELY FROM THE
*                          IPL-V 7090 SYSTEM.  WHEN LINE READ IS BEING
*                          ASSEMBLED WITH THE IPL-V SYSTEM, THEY MUST
*                          (REPEAT, MUST) BE OMITTED.
          ORG      23000                ORIGIN OF LINE READ PRIMITIVES
          PZE      0,0,YLAST            REMOVE SECTION FROM AVAILABLE SPACE
*                                  Y0 READS A CARD INTO BUFFER L902 AND
*                                  THEN TRANSFERS THE INFORMATION TO THE
*                                  1W24 BUFFER.
*
 Y0       SXD      Y0003,4
          TSX      L2,4                 READ CARD
          TRA      Y0001                NO CARD
          LAC      W24,4
          LDQ      0,4                  TO BUFFER INFORMATION TO MQ
          CLA      Y0002                FROM BUFFER INFORMATION TO ACC
          AXC      1,3                  SET STARTING COLUMNS
          TSX      YY9,4                TRANSFER INFORMATION
          LXD      Y0003,4
          TRA      J4
 Y0001    LXD      Y0003,4
          TRA      J3
 Y0002    PZE      L902-1,0,80          ZEROTH WORD, 0, CARD SIZE
 Y0003    PZE      0,0,**               X4
*
*                                  Y1 INPUTS LINE SYMBOL TO H0 AND SETS H5
*
 Y1       SXD      Y1001,4
          CLA      W30
          STA      *+1
          LXA      **,1                 FIELD SIZE TO X1
```

```
        SXA     Y1002,1          SAVE FIELD SIZE
        TIX     Y1003,1,1        TRANSFER IF NOT SINGLE CHARACTER
        TSX     Y6,4             INPUT LINE CHARACTER
Y1004   LAC     W25,2
        CLA     0,2
        ADD     Y1002            ADVANCE 1W25
        STO     0,2
        LXD     Y1001,4
        TRA     1,4
Y1003   LAC     W24,2
        CLA     0,2
        LAC     W25,2
        ADD     0,2
        ANA     YI86
        COM
        PAX     0,2              1COM FIRST COLUMN TO X2
        TSX     L6+1,4           CONVERT SYMBOL
        TMI     Y1005            TRANSFER IF ALL BLANKS
        PDX     0,1
        TXH     *+2,1,0          TRANSFER IF REGIONAL
        ACL     YI90             Q=4
        STP     Y1001
        STZ     L99              SET INTERNALS ABSOLUTE
        TSX     L7,4             GET SYMBOL
        STA     Y1001
        TSX     S4,4             PSV H0
        TSX     J4,4
        CAL     Y1001
        STA     H0
        STP     H0
        TRA     Y1004
Y1005   TSX     J3,4
        TRA     Y1004
Y1001   PZE     **,0,**          SYMBOL,X4
Y1002   PZE     **               HOLDS FIELD SIZE
*
*                                Y2 INPUTS LINE DATA TERM (0), SETS H5.
*                                   (IF NOT INTEGER OR OCTAL, ASSUME BCD)
*
Y2      SXD     Y1001,4
        LAC     H0,2
        CAL     0,2
        ANA     Y2002
        TZE     Y2003            TRANSFER INTEGER
        SUB     Y2020
        TZE     Y2030            TRANSFER OCTAL
        TSX     YY2,4            2CFIRST TO X2, FIELD SIZE TO X1
        TXL     *+2,1,5
        AXT     5,1              CLEAR FIELD
        PXD     0,0              SET UP FULL FIELD SIZE
        AXT     5,4
Y2004   ALS     6                ADD IN CHARACTER
        ADD     0,2              REDUCE FIELD SIZE
        TXI     *+1,4,-1
```

```
        TXI     *+1,2,-1        MOVE TO NEXT COLUMN
        TIX     *-4,1,1         TEST IF DONE
        TXL     *+4,4,0         TEST IF FULL FIELD
        ALS     6
        ADD     YL696           ADD IN BLANK
        TIX     *-2,4,1
        ADD     Y2006           SET PQ
        LAC*    H0,2
        STO     0,2             SET NEW ALPHA DATA TERM
        SUB     Y2007
        TZE     *+3             TRANSFER IF ALL BLANKS
Y2014   TSX     J4,4
        TRA     Y1004           ADVANCE 1W25
Y2015   TSX     J3,4
        TRA     Y1004
Y2003   TSX     YY2,4           2C FIRST TO X2, FIELD SIZE TO X1
        STZ     Y2008           CLEAR SUM
        AXT     0,4             COUNT NON-BLANKS
Y2011   CLA     0,2
        CAS     YL694           TEST IF NON-NUMERIC
        TRA     Y2012
        TRA     Y2012
        LDQ     Y2008
        MPY     YL694           SUMX10
        XCA
        ADD     0,2             ADD NEXT DIGIT
        STO     Y2008
        TXI     *+1,4,1
Y2012   TXI     *+1,2,-1        MOVE TO NEXT COLUMN
        TIX     Y2011,1,1       TRANSFER IF NOT DONE
        CLA     Y2008
        ANA     Y2009
Y2022   ADD     YI87            SET PQ
        LAC     H0,2
        STO     0,2
        TXH     Y2014,4,0       TEST IF ANY NON-BLANKS
        TRA     Y2015
Y2030   TSX     YY2,4           ZCFIRST TO X2, FIELD SIZE TO X1
        STZ     Y2008           CLEAR SUM
        AXT     0,4             COUNT NON-BLANKS
Y2031   CLA     0,2
        CAS     Y2021           TEST IF NON-OCTAL
        TRA     Y2032
        TRA     Y2032
        CLA     Y2008
        ALS     3
        ADD     0,2             ADD NEXT DIGIT
        STO     Y2008
        TXI     *+1,4,1
Y2032   TXI     *+1,2,-1        MOVE TO NEXT COLUMN
        TIX     Y2031,1,1
        CLA     Y2008
        ANA     Y2009
        ADD     Y2020           SET Q
```

```
        TRA     Y2022
Y2020   OCT     030000000000        P=3
Y2021   PZE     8                   LIMIT ON OCTAL DIGITS
 Y2002  OCT     070000000000        MASK FOR P
 Y2006  OCT     120000000000        EMPTY ALPHA DATA TERM
 Y2007  OCT     126060606060        ALPHA ALL BLANKS
 Y2008  PZE     **                  HOLDS SUM FOR INTEGER CASE
 Y2009  OCT     7777777777
 YY2    CLA     W30
        STA     *+1
        LXA     **,1                FIELD SIZE TO X1
        SXA     Y1002,1             SAVE FIELD SIZE
        LAC     W24,2
        CLA     0,2                 BUFFER START TO H0
        LAC     W25,2
        ADD     0,2                 ADD FIRST COLUMN
        PAC     0,2                 2C FIRST ADDRESS TO X2
        TRA     1,4
*
*                                   Y3 SETS (0) TO NEXT BLANK BEGINNING WITH
*                                   COLUMN 1W25+1, AND SETS H5.
*
 Y3     SXD     Y3001,4
        TSX     YYY3,4              SET UP SCAN
 Y3006  SXA     Y3002,1             SAVE NO. OF COLUMNS
        LDQ     YL696               SET UP BLANK
 Y3007  TSX     YY3,4               SCAN
        TRA     Y3003
        TSX     J4,4                SET H5+
 Y3005  SXA     Y3004,1             SAVE NO. OF UNSCANNED COLUMNS
        LAC     H0,1
        CLA     0,1
        ADD     Y3002
        SUB     Y3004
        STO     0,1                 RESET (0)
        LXD     Y3001,4
        TRA     1,4
 Y3003  TSX     J3,4                SET H5-
        TRA     Y3005
Y3001   PZE     0,0,**              X4
 Y3002  PZE     **                  HOLDS INITIAL NO. OF COLUMNS
 Y3004  PZE     **                  HOLDS NO. OF COLUMNS UNSCANNED
*
*                                   Y4 SETS (0) TO NEXT NON-BLANK
*
 Y4     SXD     Y3001,4
        TSX     YYY3,4              SET UP SCAN
        SSM                         SET FOR NON-IDENTITY
        TRA     Y3006
*
*                                   Y5 SETS (1) TO NEXT OCCURRENCE OF (0),
*                                   LEAVES INPUT (1) AS OUTPUT (0)
*
 Y5     SXD     Y3001,4
```

```
        CLA     H0
        STA     Y5001
        CLA     Y5001
        TSX     P3,4                    GET REGIONAL SYMBOL
        TRA     Y5002                   NOT REGIONAL
 Y5004  STA     Y5001
        TSX     S5,4                    RESTORE H0
        TSX     YYY3,4                  SET UP SCAN
        SXA     Y3002,1
        LDQ     Y5001                   CHARACTER TO MQ
        TRA     Y3007
 Y5002  LDQ     Y5001
        PXD     0,0
        DVP     YL694                   GET LOW ORDER DIGIT
        TRA     Y5004
 Y5001  PZE     **                      HOLDS SEARCH CHARACTER
*
*                                       Y6 INPUTS LINE CHARACTER TO H0 AND SETS
*                                       H5.
*
*
 Y6     SXD     Y6001,4
        TSX     YY6,4
        STA     *+1
        CLA     **,2
        STO     Y6002                   SAVE CHARACTER
        SUB     YL696
        TNZ     *+3                     TRANSFER IF NOT BLANK
        LXD     Y6001,4
        TRA     J3                      CHARACTER BLANK, SET H5-
        TSX     S4,4                    PSV H0
        CLA     Y6002
        SUB     YL694
        TPL     Y6003                   TRANSFER IF NOT INTERNAL
        CLA     Y6002
        ACL     YI90                    Q=4
 Y6004  STA     H0
        STP     H0
        LXD     Y6001,4
        TRA     J4
 Y6003  LXA     Y6002,2
        PXD     0,2                     SET UP REGIONAL SYMBOL
        TSX     L7,4                    ASSIGN EQUIVALENT
        TRA     Y6004
Y6001   PZE     0,0,**                  X4
 Y6002  PZE     **                      HOLDS CHARACTER
*
*                                       Y9 TRANSFERS THE FIELD IN 1W24, STARTING
*                                       IN COLUMN 1W25, FIELD SIZE 1W30 TO LINE
*                                       (0), STARTING IN COLUMN 1W21
*
 Y9     SXD     Y9EXT,4
        LAC     H0,4
        LDQ     0,4                     TO BUFFER INFORMATION TO MQ
```

```
        CLA     W25
        STA     Y9002
        STA     *+1
        LAC     **,2            FROM START COLUMN TO X2
        CLA     W21
        STA     *+1
        LAC     **,1            TO START COLUMN TO X1
        CLA     W30
        STA     *+1
        LXA     **,4
        SXD     Y9001,4         NO. OF COLUMNS TO ACC. WORD
        LAC     W24,4
        CLA     0,4
        STA     Y9001           FROM BUFFER NAME TO ACC. WORD
        CLA     Y9001
        TSX     YY9,4           TRANSFER INFORMATION
        PXD     0,2
        PDC     0,2
Y9002   SXA     **,2            RESET 1W25
        LXD     Y9EXT,4
        TRA     S5                      RESTORE HO
Y9001   PZE     **,0,**                 FROM BUFFER CONTROL
Y9EXT   PZE     0,0,**                  X4
*
*                               YY3 SCANS BUFFER FIELD.
*                               ACC=BUFFER CONTROL WORD
*                               MQ=SYMBOL
*                               SIGN(ACC)=+ FOR IDENTITY,-FOR UNIDENTITY
*                               IR1=NO. OF COLUMNS TO BE SCANNED
*                               2CIR2=COLUMN BEFORE FIRST TO BE SCANNED
*                               ON EXIT--
*                               +1 IF NOT SUCCESSFUL, +2 IF SUCCESSFUL.
*                               IR1 = NO. OF COLUMNS NOT YET SCANNED
*                               2CIR2 = LAST COLUMN SCANNED
*
YY3     STQ     YY301
        STA     YY302
        XCA     0                       SIGN TO MQ
YY305   TXL     YY303,1,0               TRANSFER IF ALL SCANNED
        TXI     *+1,2,-1                MOVE TO NEXT COLUMN
        TXI     *+1,1,-1                REDUCE UNSCANNED COUNT
YY302   CLA     **,2
        CAS     YY301                   COMPARE NEXT WITH SYMBOL
        TRA     *+2
        TRA     YY304
        TQP     YY305                   NOT EQUAL, TEST IF IDENTITY REQUIRED
        TRA     2,4                     FOUND REQUIRED COLUMN
YY304   TQP     2,4                     EQUAL, TEST IF IDENTITY REQUIRED
        TRA     YY305
YY303   TRA     1,4
YY301           **                      CHARACTER BEING COMPARED
*
*                               YYY3 SETS UP FOR SCAN, EXCEPT FOR SYMBOL IN
*                               MQ AND SIGN OF ACC FOR IDENTITY OR NOT
```

```
*                                   (IS + NORMALLY)
*
 YYY3   SXD     YYY31,4
        TSX     YY6,4
        PDX     0,1                      LAST COLUMN TO X1
        SXD     *+1,2
        TXI     *+1,1,**                 SUBTRACT COLUMN BEFORE FIRST
        LXD     YYY31,4
        TRA     1,4
YYY31   PZE     0,0,**                   X4
*
*                                   YY6 LOADS 2C OF 1W25 (COLUMN NO.) IN IR2,
*                                   2C OF BUFFER SYMBOL IN IR1, AND
*                                   BUFFER CONTROL WORD IN ACC.
*
 YY6    LAC     W24,1
        CLA     W25
        STA     *+1
        LAC     **,2
        CLA     0,1
        TRA     1,4
*
*                                   YY9 TRANSFERS INFORMATION BETWEEN BUFFERS.
*                                   A(ACC)=ZEROTH WORD OF FROM BUFFER.
*                                   D(ACC)=NO. OF COLUMNS TO TRANSFER.
*                                   A(MQ)=ZEROTH WORD OF TO BUFFER.
*                                   D(MQ)=LAST COLUMN OF TO BUFFER.
*                                   X1=2COM OF STARTING COLUMN, TO BUFFER.
*                                   X2=2COM OF STARTING COLUMN, FROM BUFFER.
*                                   OUTPUT X2=2COM OF NEXT COLUMN, FROM BUF.
*                                    H5 SET WHETHER ALL TRANSFERRED.
*
 YY9    SXD     YY9XT,4
        STO     YY9FM
        STQ     YY9TO
        CLA     YY904
        STT     YY9TO                    SET TAG=1
        CLA     YY905
        STT     YY9FM                    SET TAG=2
        LDC     YY9TO,4
        TXI     *+1,4,-1
        SXD     YY902,4                  SET BUFFER LIMIT
        LXD     YY9FM,4
        TXI     *+1,4,1                  NO. OF COLUMNS TO X4
 YY901  TIX     YY902,4,1                TEST IF ALL TRANSFERRED
        LXD     YY9XT,4
        TRA     J4
 YY902  TXH     YY903,1,**               TEST IF BUFFER FULL
        LXD     YY9XT,4
        TRA     J3
 YY903  CLA*    YY9FM
        STO*    YY9TO
 YY904  TXI     *+1,1,-1                 (TAG USED AS CONSTANT)
 YY905  TXI     YY901,2,-1               (TAG USED AS CONSTANT)
```

```
 YY9FM  PZE     **              HOLDS FROM BUFFER CONTROL
 YY9TO  PZE     **              HOLDS TO BUFFER CONTROL
 YY9XT  PZE     0,0,**          X4
*
YI86    PZE     -1              SYMBOL MASK
YI87    PON                     Q=1
YI90    FOR                     Q=4
YL694   PZE     10
YL695   PZE     9
YL696   OCT     60              BLANK
YLAST   PZE     0,0,*+1
        ORG     W+30
W30     FOR                     INPUT FIELD CONTROL CELL
        ORG     J+180
        FVE     Y0              J180
        FVE     Y1              J181
        FVE     Y2              J182
        FVE     Y3              J183
        FVE     Y4              J184
        FVE     Y5              J185
        FVE     Y6              J186
        BSS     2
        FVE     Y9              J189
        END     0
```

000

000

BAKER B060 TAPE ASSIGNMENT AS OF 09-18-61

JOB	REEL	ASSIGNED	RELEASE	DESCRIPTION	COMMENTS
6331	280H	09-13-61	- -	IPL-V SQZ-7	
6331	543H	07-31-61	- -	COMPILER BCD	
6331	579H	09-12-61	- -	IPL COMPILER RESTART	
6331	596H	08-01-61	- -	COMPILER PPT	
6331	1061H	09-13-61	- -	IPL-5 SAP OUTPUT	
6331	1075H	09-12-61	- -	IPL-V SQZ-8	
6331	1193H	09-12-61	- -	IPL-5 SAP INPUT	
6331	1246H	09-12-61	- -	IPL-V SQZ-8 SOT	
6331	1291H	08-21-61	- -	ML COMPILER RESTART	
6331	1331H	09-12-61	- -	IPL-V SQZ-8 POT	

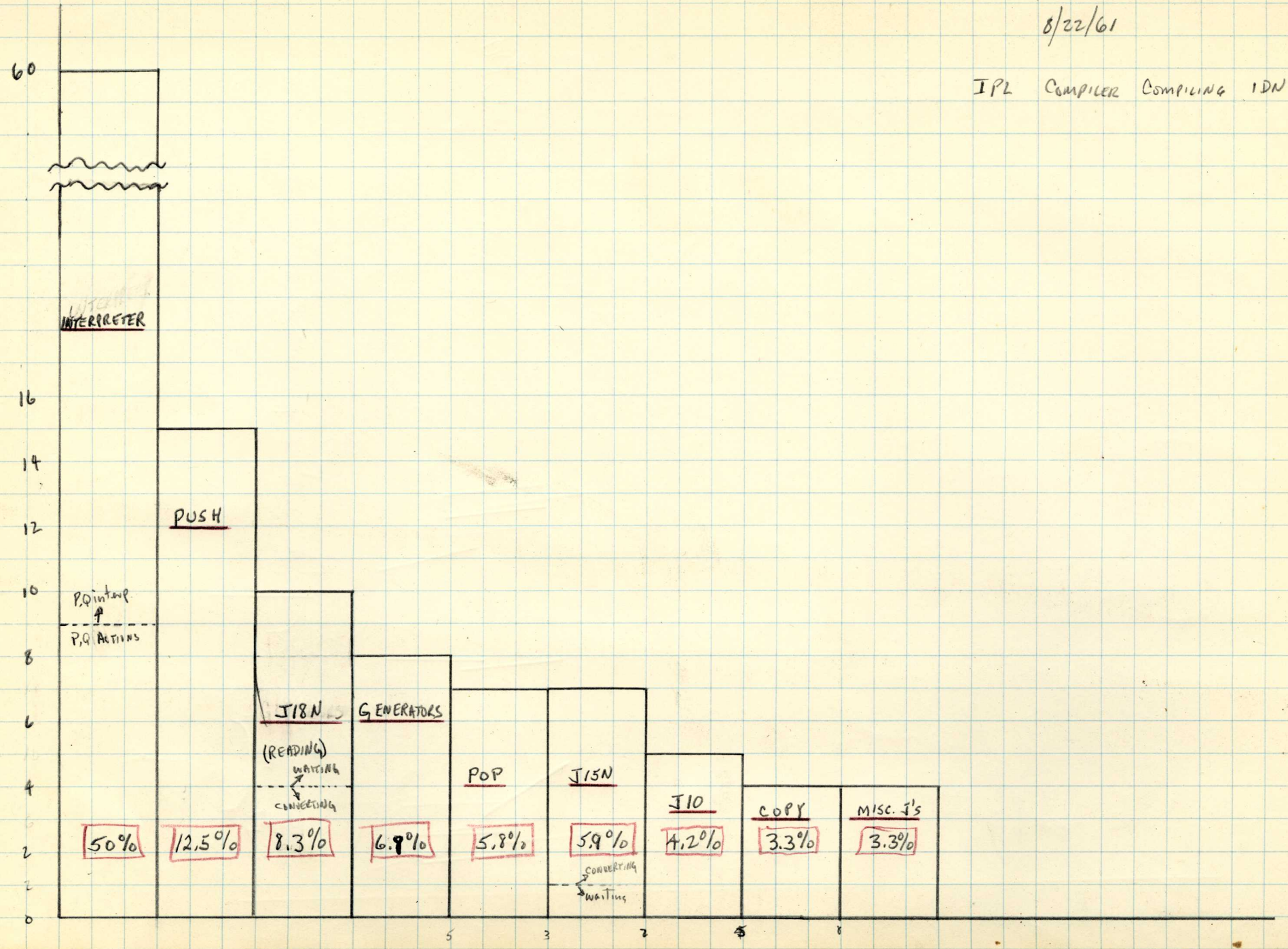
010 REELS

```
                     63466    1          ORG   J+190
63466 -1 CC000 0 56535        2          FVE   J190             J190 SET P OF (0)=P OF (1), LEAVE (0)
63467 -1 0C000 0 56540        3          FVE   J191             J191 SET Q OF (0)=Q OF (1)
63470 -1 CC000 0 56543        4          FVE   J192             J192 SET PQ OF (0)=PQ OF (1)
63471 -1 CC000 0 56546        5          FVE   J193             J193 SET SYMB CF (0)=SYMB OF (1)
63472 -1 CC000 0 56551        6          FVE   J194             J194 SET LINK OF (0)=LINK OF (1)
63473 -1 CC000 0 56554        7          FVE   J195             J195 TEST P OF (0)=P CF (1)
63474 -1 00000 0 56557        8          FVE   J196             J196 TEST Q OF (0)=Q CF (1)
63475 -1 CC000 0 56562        9          FVE   J197             J197 TEST PQ OF (0)=PQ OF (1)
63476 -1 CC000 0 56565       10          FVE   J198             J198 TEST SYMB OF (0)=SYMB OF (1)
63477 -1 00000 0 56570       11          FVE   J199             J199 TEST LINK OF (0)=LINK OF (1)
                     56534   12          ORG   2390C
56534  0 56661 0 CC000       13       -1 PZE   C,0,R298+1       LINK AV. SP. ARCUND J190'S
56535 -0 50000 0 56537       14 J190     CAL   *+2
56536  0 02000 0 56573       15       +1 TRA   R1
56537 -3 77777 0 77777       16       +2 SVN   0-1,0,-1         P MASK
56540 -0 50000 0 56542       17 J191     CAL   *+2
56541  0 02000 0 56573       18       +1 TRA   R1
56542  0 77777 7 77777       19       +2 PZE   0-1,7,-1         Q MASK
56543 -0 50000 0 56545       20 J192     CAL   *+2
56544  0 02000 0 56573       21       +1 TRA   R1
56545  0 77777 0 77777       22       +2 PZE   0-1,0,-1         PQ MASK
56546 -0 50000 0 56550       23 J193     CAL   *+2
56547  0 02000 0 56573       24       +1 TRA   R1
56550 -3 77777 7 CC000       25       +2 SVN   0,7,-1           SYMB MASK
56551 -0 50000 0 56553       26 J194     CAL   *+2
56552  0 02000 0 56573       27       +1 TRA   R1
56553 -3 CCOCO 7 77777       28       +2 SVN   0-1,7            LINK MASK
56554 -0 50000 0 56556       29 J195     CAL   *+2
56555  0 02000 0 56621       30       +1 TRA   R2
56556  0 C0000 7 C0000       31       +2 PZE   0,7              P MASK
56557 -0 50000 0 56561       32 J196     CAL   *+2
56560  0 02000 0 56621       33       +1 TRA   R2
56561 -3 00000 0 0C000       34       +2 SVN                    Q MASK
56562 -0 50000 0 56564       35 J197     CAL   *+2
56563  0 02000 0 56621       36       +1 TRA   R2
56564 -3 00000 7 CC000       37       +2 SVN   0,7              PQ MASK
56565 -0 50000 0 56567       38 J198     CAL   *+2
56566  0 02000 0 56621       39       +1 TRA   R2
56567  0 CC000 0 77777       40       +2 PZE   0-1              SYMB MASK
56570 -0 50000 0 56572       41 J199     CAL   *+2
56571  0 02000 0 56621       42       +1 TRA   R2
56572  0 77777 0 CC000       43       +2 PZE   0,0,-1           LINK MASK
                             44*
56573  0 60200 0 56620       45 R1       SLW   R199             SET (0) = (1) WITH MASK IN ACC.
56574  0 56000 0 62707       46       +1 LDQ   H+2
56575  0 50000 0 62705       47       +2 CLA   H
56576  0 62200 0 62707       48       +3 STD   H+2
56577  0 73700 1 CC000       49       +4 PAC   0,1              -(0) TO X1
56600  0 77100 0 00022       50       +5 ARS   18
56601  0 62100 0 56611       51       +6 STA   R101
56602  0 62100 0 56603       52       +7 STA   *+1
56603 -0 50000 0 00000       53       +8 CAL   **0
56604 -0 10000 0 56607       54       +9 TNZ   *+3
56605 -0 60000 0 62707       55      +10 STQ   H+2              FIX H2
56606  0 02000 0 70523       56      +11 TRA   E11              NCT ENOUGH INPUTS
```

```
56607  0 73700 2 CC000    57      +12  PAC  C,2      -(1) TO X2
5661C  0 62200 0 62705    58      +13  STD  H
56611 -0 60000 0 CC000    59  R101     STQ  **0
56612 -0 50000 0 56620    60      +1   CAL  R199
56613  0 32000 1 CC000    61      +2   ANS  C,1      ZERC PORTION TO CHANGE
56614  0 76000 0 CC006    62      +3   COM
56615 -0 32000 2 CC000    63      +4   ANA  C,2      GET NEW PORTION
56616 -0 60200 1 CC000    64      +5   ORS  C,1      INSERT
56617  0 02000 4 CC001    65      +6   TRA  1,4
5662C  0 00000 0 CC000    66  R199     PZE           MASK
                          67*
56621  0 60200 0 56657    68  R2       SLW  R299     TEST (0) = (1) WITH MASK IN ACC.
56622  0 5C000 0 62705    69      +1   CLA  H
56623  0 73700 1 CC000    70      +2   PAC  C,1      -(0) TO X1
56624  0 56000 0 62707    71      +3   LDQ  H+2
56625  0 62200 0 62707    72      +4   STD  H+2      LINK FOR H2
56626  0 77100 0 CC022    73      +5   ARS  18
56627  0 62100 0 56630    74      +6   STA  *+1
56630 -0 50000 0 CC000    75      +7   CAL  **0
56631 -0 10000 0 56634    76      +8   TNZ  *+3
56632 -0 60000 0 62707    77      +9   STQ  H+2      FIX H2
56633  0 02000 0 70523    78      +1C  TRA  E11      NOT ENOUGH INPUTS
56634  0 73700 2 CC000    79      +11  PAC  C,2      -(1) TO X2
56635  0 77100 0 CC022    80      +12  ARS  18
56636  0 62100 0 56640    81      +13  STA  *+2
56637  0 62100 0 56644    82      +14  STA  *+5
56640 -0 50000 0 CC000    83      +15  CAL  **0
56641 -0 10000 0 56644    84      +16  TNZ  *+3
56642 -0 60000 0 62707    85      +17  STQ  H+2      FIX H2
56643  0 02000 0 70523    86      +18  TRA  E11      NOT ENOUGH INPUTS
56644 -0 60000 0 CC000    87      +19  STQ  **0
56645  0 60200 0 62705    88      +20  SLW  H
56646 -0 50000 1 CC000    89      +21  CAL  C,1
56647 -0 32000 0 56657    90      +22  ANA  R299     MASK (0)
56650  0 60200 0 56660    91      +23  SLW  R298
56651 -0 50000 2 CC000    92      +24  CAL  C,2
56652 -0 32000 0 56657    93      +25  ANA  R299     MASK (1)
56653 -0 34000 0 56660    94      +26  LAS  R298     COMPARE
56654  0 02000 0 70774    95      +27  TRA  J3
56655  0 02000 0 71000    96      +28  TRA  J4
56656  0 02000 0 70774    97      +29  TRA  J3
56657  0 00000 0 CC000    98  R299     PZE           MASK
56660  0 00000 0 CC000    99  R298     PZE
                 63170   100  J        EQU  26232
                 62705   101  H        EQU  26053
                 70523   102  E11      EQU  29011
                 70774   103  J3       EQU  29180
                 71000   104  J4       EQU  29184
                 12715   105           END  0
```

8/22/61

IPL COMPILER COMPILING IDN



8/21/61

RANDOM (?) SAMPLE OF
INSTR. COUNTER IN IPL-V.
M.L. COMPILER COMPILING COMPILER.
SAMPLE SIZE = 120