

PROGRAM TESTING

	<u>Page</u>
1. CONTENTS	i
2. PROGRAM TESTING	1
A. GENERAL	1
B. DIAGNOSTIC ROUTINES	1
1. DUMP and SNAP	1
a. DUMP Macro	1
b. SNAP Macro	2
c. Examples	3
2. PCALL and PRIN	3
a. PCALL Macro	3
b. PRIN Macro	4
c. Examples	5
3. REPL	5

A. GENERAL

The purpose of program testing is to ensure that a program, once it has been written and assembled, actually performs the task it was designed to accomplish. Program testing basically involves the evaluation of actual output values produced by a program in terms of expected output, and the correction of those portions of a program which produce any variations from expected performance.

Program testing may be further refined to include the evaluation of intermediate values generated by a program which never appear on output media but which may critically affect output.

B. DIAGNOSTIC ROUTINES

A number of diagnostic procedures, which a program may call at any point, are incorporated into the PAL Assembler to record on the printer intermediate and final results of computation or other program output. These procedures are predefined and do not require definition within the program using them.

These program-testing procedures enable the programmer to evaluate the results produced by each logical subdivision or subroutine of his program. They provide the facility of testing each subroutine independently or in conjunction with the rest of the program.

When all portions of the program have been tested and found to function as intended, the corrected source program is reassembled without the program-testing procedure calls.

1. DUMP and SNAP

a. DUMP Macro

This instruction includes in the object program a routine to print a portion of memory. This routine can be entered only by the SNAP macro.*

The DUMP instruction may have any of the following forms:

LABEL	OPERATION	OPERANDS
	DUMP	
	DUMP	a
	DUMP	a, b
	DUMP	a, , c
	DUMP	a, b, c
	DUMP	, b
	DUMP	, b, c
	DUMP	, , c

*Performing a transfer of control on the console to the label of the DUMP call will print all of memory and stop the computer.

- A blank operand field states that there is no standard print routine as part of the worker program, and the DUMP routine will generate areas for printing and temporary storage.
- Parameter "a" is PRINT if the standard I/O print routine is part of the worker program and full lines are printed.
- Parameter "b" is an expression that represents an area of 128 characters in the worker program which may be used by the DUMP routine for printing.
- Parameter "c" is an expression that represents an area of 74 characters in the worker program which may be used by the DUMP routine for temporary storage. This may be a combined expression.

The tetrad area is restored after DUMP is used. When the standard print routine is not used, the printer base address, line counter, and interrupt entry of channel 0 are not printed correctly.

b. SNAP Macro

This instruction provides entry to the DUMP routine in order to print a specified portion of memory in a specified mode.

The SNAP instruction may have any of the following forms:

LABEL	OPERATION	OPERANDS
	SNAP	
	SNAP	a, b
	SNAP	a, b, A
	SNAP	a, b, T
	SNAP	a, b, R
	SNAP	a, b, T, A
	SNAP	a, b, R, A

- A blank operation field will cause the whole memory, including the tetrads, to be printed in octal.
- Parameter "a" and "b" represent, respectively, the lowest and highest addresses to be included in the printing. All locations from "a" to "b" inclusive will be represented in octal or alphanumeric format. Parameters "a" and "b" may have an implied index register. Parameter "b" must be greater than "a".
- Parameter "T" will cause the whole tetrad area to be printed in octal in addition to the information in locations from "a" to "b".
- Parameter "R" will just print the arithmetic and index registers of the tetrad area in addition to the information in locations from "a" to "b".
- Parameter "A" will cause the area from "a" to "b" inclusive to be printed in alphanumeric format.

To switch off SNAP, depress the STOP button and manually insert 00 at memory location XDUMP + 9.

LABEL	OPERATION	OPERANDS
	PCALL	a, b
	PCALL	a, , c
	PCALL	, b
	PCALL	, b, c
	PCALL	, , c

- Parameter "a" is as mentioned above.
- Parameter "b" is an expression that addresses an area of 128 characters or more in the worker program that can be used by the PCALL routine for printing.
- Parameter "c" is an expression that addresses an area of 65 characters or more in the worker program that can be used by the PCALL routine for temporary storage.

b. PRIN Macro ~~✗~~

This instruction, which must be used in conjunction with the PCALL instruction, causes the value of a field to be printed unconditionally or only when a specified condition exists.

The PRIN instruction may have any of the following forms:

LABEL	OPERATION	OPERANDS
	PRIN	a, b, c, d } Conditional Print
	PRIN	a, b, c } Conditional Print
	PRIN	AL, b, d } Unconditional Print
	PRIN	AL, b } Unconditional Print

- Parameter "a" specifies unconditional or conditional printing. If conditional printing is desired, parameter "a" is either EB, UB, GB, or SB*, if binary comparisons are to be made; it is ED, UD, GD, or SD,** if decimal comparisons are to be made. If unconditional printing is desired, parameter "a" is AL and parameter "c" is omitted. Printing will occur for:

* Equal Binary; Unequal Binary; Greater Binary; Smaller Binary.

**Equal Decimal; Unequal Decimal; Greater Decimal; Smaller Decimal.

EB, ED when the value of $b = c$
 UB, UD when the value of $b \neq c$
 GB, GD when the value of $b > c$
 SB, SD when the value of $b < c$

- Parameter "b" is the label of the field to be compared; it is also the field that is printed if the condition is met. If this field exceeds 16 characters, only the 16 least significant positions are used. An index register may also be implied.
- Parameter "c" is the label of the constant used in the comparison. It must be one of the data generation forms.
- Parameter "d" is an expression that allows the user to identify the lines being printed. It is normally the label specified by parameter b enclosed in quotes, used as a constant to print on the line. If this parameter is omitted, the word PRINT will be printed.

The PRINT format follows:

<u>LABEL</u>	<u>ADDRESS</u>	<u>VALUE</u>	<u>VALUE</u>	<u>CONSTANT</u>	<u>CONSTANT</u>
"d" or PRINT	of the value	Alpha- numeric	Octal	Alpha- numeric	Octal

c. Examples

It is desired to print the results of the inventory amount contained in a field addressed by INV3 whenever it exceeds the maximum amount contained in a constant labeled K3. It is also desired to identify this amount when printed. In addition, regardless of whether INV3 is printed or not, it is desired to print the contents of INV7. The following coding will accomplish the desired results.

— For BEGIN only.

SEQUENCE		LABEL	OPERATION		OPERANDS			
PAGE	LINE	INS	7	11	13	18	19	30
1	3	4	5	6				
					PCALL	PRINT	WSTR4	
					PRIN	GD	INV3	K3
					PRIN	AL	INV7	INV7
			K3		+5			'783'

The printed result will be

LABEL	ADDRESS	VALUE (ALPHANUMERIC)	VALUE (OCTAL)	CONSTANT (ALPHANUMERIC)	CONSTANT (OCTAL)
INV3	704	00785	01421	00783	01417

3. REPL

This instruction enables the programmer to replace certain computed results with predicted results and thus continue to test portions of a program beyond one that is not functioning properly. This enables one program test run to uncover several program errors and provides more efficient use of the computer for debugging. The form of the REPL instruction follows:

LABEL	OPERATION	OPERANDS
	REPL	$L_1, V_1, L_2, \dots, L_n, V_n$

where L_i is the label addressing the field that receives the constant V_i . A maximum of 5 fields ($n \leq 5$) is allowed for each REPL line.

NOTE:

L_i must be a label with an implied length.

V_i must be ≤ 16 characters in length and must follow the rules for data generation.

Example:

It is desired to have the value of a field labeled RATE preset to the value of 78 prior to entering a section of a program. The present contents of the field RATE are the results of previous computations in the program and may be in error if those sections contain programming errors. Normally it would be desirable to see the present contents of RATE before it is changed by the REPL macro. The coding to do this might appear as follows:

SEQUENCE			LABEL	OPERATION	OPERANDS					
1	PAGE	3	LINE	INS	7	11	13	18	19	30
				PRIN	AL, RATE, 'RATE'					
				REPL	RATE, K 7					
			K 7	+ 2	7 8					

For BEGIN only.

Execution of these macros will cause the printing of the value contained in the field RATE and will also change this value to 78. RATE must be a label with a defined field length.