



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-271
TITLE	LIP, LOGICAL "IF" PACKAGE
AUTHOR	Bryan D. Young
COMPANY	University Department of Medical Cardiology Royal Infirmary Glasgow, Scotland
DATE	February 17, 1970
SOURCE LANGUAGE	PALD



# LIP, LOGICAL "IF" PACKAGE

DECUS Program Library Write-up

DECUS No. 8-271

## ABSTRACT

LIP enables logical "IF" statements involving floating point variables to be written in a conventional manner within assembly language programs. The logical relations and operators available are:-

<u>Mnemonic</u>	<u>Meaning</u>
GT	Greater than
GE	Greater than or equal to
EQ	Equal to
LE	Less than or equal to
LT	Less than
NE	Not equal to
AN	Logical AND
OR	Logical OR

e. g. the FORTRAN statement:-

IF ((A. GT. B)OR(C. LT. D)) LOC

is written in machine code format as:-

IF;A;GT;B;OR;C;LT;D;GO;LOC where A, B, C and D are 3 word floating point variables.

### 3. Requirements

#### 3.1 Storage.

This package occupies locations 4, 5400-5577. The 3 word Floating Point package (Digital 8-5-S) which must also be in core occupies locations 5-7, 40-63, 5600-7577.

Therefore total storage requirements are 4-7, 40-63,  
5400-7577.

### 3.2 Equipment.

Standard PDP-8 , ASR-33 Teletype.

## 4. Usage

### 4.1 Loading.

This package is loaded with the binary loader (Digital 8-2-U).  
The binary tape supplied contains only the LIP package and  
hence the 3 word Floating Point package (Digital 8-5-S)  
must also be loaded.

### 4.2 Calling Sequence.

Location 4 is used as an indirect address for entry to the  
package. This is accomplished by use of the IF instruction  
which is equivalent to JMS I 4 (see Discussion).

### 4.3 Definitions.

Since the mnemonic instructions used in the "IF" statements  
are not held in the permanent symbol tables of the PAL and  
MACRO compilers they must be defined by the user's  
program as follows:-

IF	=	4404
GO	=	7777
OR	=	7776
AN	=	7775
GT	=	7774
GE	=	7773
EQ	=	7772
LE	=	7771
LT	=	7770
NE	=	7767

5. Restrictions

None.

6. Description

6.1 Discussion

LIP enables logical "IF" statements to be written within assembly language programs in a format similar in function and appearance to the corresponding FORTRAN and ALGOL statements. The general form is:-

IF;(logical expression);GO;LOC ..... (1)

The complete statement is normally written on one line with semicolons between each variable name and instruction.

However, if a statement is too long it may be continued on subsequent lines.

Each statement must include both the IF and GO instructions which initiate the entry to and exit from the package respectively.

The logical expression, indicated in (1) is evaluated by the package and if it is found to be true control is transferred to the location named after the GO instruction (LOC in (1)). If it is not true control is returned to the next location after the end of the IF statement. The logical expression is composed of floating point variable names combined using the following relations and logical operators.

<u>Mnemonic</u>	<u>Meaning</u>
GT	Greater than
GE	Greater than or equal to
EQ	Equal to ( = )
LE	Less than or equal to
LT	Less than
NE	Not equal to
AN	Logical AND operator
OR	Logical OR operator

Examples of logical expressions:-

A > B	A;GT;B
(A < B)OR(C > D)	A;LT;B;OR;C;GT;D
(A = B)AND (C = D)	A;EQ;B;AN;C;EQ;D
A > B > C	A;GT;B;AN;B;GT;C

All variable names must be 3 word floating point numbers.

Although there is no limit to the size of logical expressions, it is important to note that evaluation takes place from left to right.

e. g. A;GT;B;AN;C;LT;B;OR;C;LE;D is equivalent to:-

((A > B) AND (C < B)) OR (C < D))

## 6.2 Example.

Let A, B, C and D be floating point variables. It is required to know whether any of the following conditions hold:-

- (I) A=B=C=D
- (II) A > B > C > D
- (III) A < B < C < D

This would be programed as:-

```
IF;A;EQ;B;AN;B;EQ;C;AN;C;EQ;D;GO;L1
IF;A;GT;B;AN;B;GT;C;AN;C;GT;D;GO;L2
IF;A;LT;B;AN;B;LT;C;AN;C;LT;D;GO;L3
```

L4, ..... / (I), (II), (III) are not true

L1, ..... / (I) is true.

L2, ..... / (II) is true.

L3, ..... / (III) is true.

### 6.3 Errors.

Any format errors in an IF statement will cause the package to halt. Logical variables are not allowed and therefore a statement such as:- IF;A;GO;LOC is illegal and would produce a failure. The logical operators OR and AN may link only expressions of the type A;GT;B and hence the statement:-

IF;A;OR;B;GT;C;GO;LOC

is illegal. The correct version of this statement is:-

IF;A;GT;C;OR;B;GT;C;GO;LOC

A variable name may not be both preceded and followed by one of the relations GT, GE etc. Therefore the statement:-

IF;A;GT;B;GT;C;GO;LOC is illegal.

The correct version is:-

IF;A;GT;B;AN;B;GT;C;GO;LOC.

/LIP, A LOGICAL "IF" ROUTINE  
 /FOR ASSEMBLY LANGUAGE PROGRAMS

/DEFINITIONS:-

IF=4404  
 GO=7777  
 OR=7776  
 AN=7775  
 GT=7774  
 GE=7773  
 EQ=7772  
 LE=7771  
 LT=7770  
 NE=7767  
 FGET=5000  
 FSUB=2000  
 FEXT=0000

	*4			
0004	5400		5400	/INDIRECT ADDRESS FOR ROUTINE
	*5400			
5400	0000	POINTR,	0	/USED TO EXAMINE INSTRUCTIONS
5401	3237		DCA CLV	
5402	5204		JMP .+2	
5403	2200	LPI,	ISZ POINTR	
5404	2200		ISZ POINTR	
5405	4240	LP4,	JMS FUNC	
5406	7450		SNA	
5407	5226		JMP OUT	
5410	7700		SMA CLA	
5411	5215		JMP LP2	
5412	4277		JMS VAL	
5413	3237		DCA CLV	
5414	5203		JMP LPI	
5415	2200	LP2,	ISZ POINTR	
5416	2200		ISZ POINTR	
5417	4240		JMS FUNC	
5420	7700		SMA CLA	
5421	7402		HLT	/ILLEGAL FORMAT
5422	4277		JMS VAL	
5423	4261		JMS LOG	
5424	3237		DCA CLV	
5425	5203		JMP LPI	
5426	2200	OUT,	ISZ POINTR	/EXIT FROM PACKAGE
5427	1237		TAD CLV	
5430	7650		SNA CLA	
5431	5235		JMP LP6	
5432	1600		TAD I POINTR	
5433	3200		DCA POINTR	
5434	5600		JMP I POINTR	

5435	<del>2200</del>	LP6,	ISZ POINTR	
5436	<del>5600</del>		JMP I POINTR	
5437	<del>0000</del>	CLV,	Ø	/CURRENT LOGICAL VALUE
5440	<del>0000</del>	FUNC,	Ø	/DETERMINES INSTRUCTIONS
5441	<del>1600</del>		TAD I POINTR	
5442	<del>7500</del>		SMA	
5443	<del>7402</del>		HLT	/ILLEGAL INSTRUCTION
5444	<del>7001</del>		IAC	
5445	<del>7450</del>		SNA	
5446	<del>5640</del>		JMP I FUNC	
5447	<del>1377</del>		TAD (2)	
5450	<del>7510</del>		SPA	
5451	<del>5254</del>		JMP .+3	
5452	<del>7201</del>		CLA IAC	
5453	<del>5640</del>		JMP I FUNC	
5454	<del>1376</del>		TAD (6)	
5455	<del>7710</del>		SPA CLA	
5456	<del>7402</del>		HLT	/ILLEGAL INSTRUCTION
5457	<del>7040</del>		CMA	
5460	<del>5640</del>		JMP I FUNC	
5461	<del>0000</del>	LOG,	Ø	/EVALUATES 'AND', 'OR'
5462	<del>1237</del>		TAD CLV	
5463	<del>3275</del>		DCA RHS	
5464	<del>1200</del>		TAD POINTR	
5465	<del>1375</del>		TAD (-2)	
5466	<del>3276</del>		DCA LTEM	
5467	<del>1676</del>		TAD I LTEM	
5470	<del>7001</del>		IAC	
5471	<del>1275</del>		TAD RHS	
5472	<del>7700</del>		SMA CLA	
5473	<del>7001</del>		IAC	
5474	<del>5661</del>		JMP I LOG	
5475	<del>0000</del>	RHS,	Ø	
5476	<del>0000</del>	LTEM,	Ø	
5477	<del>0000</del>	VAL,	Ø	/EVALUATES 'A;GT;'B' ETC.
5500	<del>1200</del>		TAD POINTR	
5501	<del>1374</del>		TAD (-1)	
5502	<del>3352</del>		DCA VAL2	
5503	<del>1752</del>		TAD I VAL2	
5504	<del>3351</del>		DCA VAL1	
5505	<del>2352</del>		ISZ VAL2	
5506	<del>2352</del>		ISZ VAL2	
5507	<del>1752</del>		TAD I VAL2	
5510	<del>3352</del>		DCA VAL2	
5511	<del>4407</del>		JMS I 7	
5512	<del>5751</del>		FGET I VAL1	
5513	<del>2752</del>		FSUB I VAL2	
5514	<del>0000</del>		FEXT	

5515	1045		TAD 45
5516	7450		SNA
5517	5327		JMP EQUAL
5520	7700		SMA CLA
5521	5337		JMP GRET
5522	1600		TAD I POINTR
5523	1376		TAD (6)
5524	7710		SPA CLA
5525	7001		IAC
5526	5677		JMP I VAL
5527	1600	EQUAL,	TAD I POINTR
5530	1373		TAD (4)
5531	7500		SMA
5532	5677		JMP I VAL
5533	1372		TAD (3)
5534	7700		SMA CLA
5535	7001		IAC
5536	5677		JMP I VAL
5537	1600	GRET,	TAD I POINTR
5540	1371		TAD (5)
5541	7510		SPA
5542	5345		JMP .+3
5543	7201		CLA IAC
5544	5677		JMP I VAL
5545	1372		TAD (3)
5546	7710		SPA CLA
5547	7001		IAC
5550	5677		JMP I VAL
5551	0000	VAL1,	Ø
5552	0000	VAL2,	Ø
5571	0005		
5572	0003		
5573	0004		
5574	7777		
5575	7776		
5576	0006		
5577	0002		

AN	7775
CLV	5437
EQ	7772
EQUAL	5527
FEXT	<del>0000</del>
FGET	<del>5000</del>
FSUB	<del>2000</del>
FUNC	5440
GE	7773
GO	7777
GRET	5537
GT	7774
IF	<del>4404</del>
LE	7771
LOG	5461
LP1	<del>5403</del>
LP2	5415
LP4	<del>5405</del>
LP6	5435
LT	<del>7770</del>
LTEM	5476
NE	7767
OR	7776
OUT	5426
POINTR	<del>5400</del>
RHS	5476
VAL	5477
VAL1	5551
VAL2	5552

