



```

FFFFFFFFF 000000 RRRRRRRR LL EEEEEEEEEE XX XX
FFFFFFFFF 000000 RRRRRRRR LL EEEEEEEEEE XX XX
FF 00 00 RR RR LL EE XX XX
FF 00 00 RR RR LL EE XX XX
FF 00 00 RR RR LL EE XX XX
FFFFFFFF 00 00 RRRRRRRR LL EEEEEEEE XX XX
FFFFFFFF 00 00 RRRRRRRR LL EEEEEEEE XX XX
FF 00 00 RR RR LL EE XX XX
FF 00 00 RR RR LL EE XX XX
FF 00 00 RR RR LL EE XX XX
FF 00 00 RR RR LL EE XX XX
FF 000000 RR RR LLLLLLLLLL EEEEEEEEEE XX XX
FF 000000 RR RR LLLLLLLLLL EEEEEEEEEE XX XX

```

```

LL I I I I I S S S S S S S
LL I I I I I S S S S S S S
LL II SS
LL II SS
LL II SS
LL II S S S S S S
LL II S S S S S S
LL II SS
LL II SS
LL II SS
LL I I I I I S S S S S S S
LL I I I I I S S S S S S S

```



(2)	50
(3)	82
(4)	171

DECLARATIONS  
FORSLEX - Lexical comparison functions  
Local functions for lexical comparisons

```
0000 1 .TITLE FOR$LEX - Lexical comparison functions
0000 2 .IDENT /1-001/ ; File: FORLEX.MAR
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28
0000 29 :++
0000 30 : FACILITY: FORTRAN language support library
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 : This module contains routines to perform lexical comparison
0000 35 : of character strings passed by descriptor. They are primarily
0000 36 : for conformance with FORTRAN-77, but are useable from any
0000 37 : language. Note that most languages have these functions
0000 38 : built-in.
0000 39
0000 40 : ENVIRONMENT: User Mode, AST Reentrant
0000 41
0000 42 :--
0000 43 : AUTHOR: Steven B. Lionel, CREATION DATE: 30-Jan-79
0000 44
0000 45 : MODIFIED BY:
0000 46
0000 47 : EDIT HISTORY:
0000 48 : 1-001 - Original. SBL 30-Jan-79
```

```
0000 50      .SBTTL  DECLARATIONS
0000 51      :
0000 52      : INCLUDE FILES:
0000 53      :
0000 54      :
0000 55      :
0000 56      : EXTERNAL DECLARATIONS:
0000 57      :
0000 58      .DSABL  GBL                : Prevent undeclared
0000 59      :                               : symbols from being
0000 60      :                               : automatically global.
0000 61      :
0000 62      :
0000 63      : MACROS:
0000 64      :
0000 65      :
0000 66      :
0000 67      : EQUATED SYMBOLS:
0000 68      :
00000004 0000 69      string1 = 4      ; First string argument
00000008 0000 70      string2 = 8     ; Second string argument
0000 71      :
0000 72      : OWN STORAGE:
0000 73      :
0000 74      :
0000 75      :
0000 76      : PSECT DECLARATIONS:
0000 77      :
00000000 0000 78      .PSECT _FOR$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 79      EXE, RD, NOWRT, LONG
0000 80
```

```

0000 82      .SBTTL FOR$LEX - Lexical comparison functions
0000 83      :++
0000 84      : FUNCTIONAL DESCRIPTION:
0000 85      :
0000 86      : The four functions in this module compare two character
0000 87      : strings according to the ASCII collating sequence.
0000 88      :
0000 89      : Most VAX-11 high level languages already implement these
0000 90      : functions in-line, but ANSI FORTRAN-77 requires the
0000 91      : FORTRAN functions LLE, LGT, LGE and LLT to be available.
0000 92      :
0000 93      : In normal cases, FORTRAN IV-PLUS will generate the necessary
0000 94      : instructions in-line, but since a programmer can force an
0000 95      : external reference to be generated, these routines are
0000 96      : necessary.
0000 97      :
0000 98      : Each routine takes two arguments, the addresses of string
0000 99      : descriptors. The strings are compared, blank padding where
0000 100     : necessary, and a TRUE or FALSE value is returned according
0000 101     : to whether the appropriate condition is satisfied or not.
0000 102     :
0000 103     : TRUE is a longword of all 1 bits, FALSE is a longword of zero.
0000 104     :
0000 105     : CALLING SEQUENCE:
0000 106     :
0000 107     : condition.wlu.v = FOR$LLT (string1.rx.dx, string2.rx.dx)
0000 108     : TRUE if string1 < string2, FALSE otherwise
0000 109     :
0000 110     : condition.wlu.v = FOR$LLE (string1.rx.dx, string2.rx.dx)
0000 111     : TRUE if string1 <= string2, FALSE otherwise
0000 112     :
0000 113     : condition.wlu.v = FOR$LGE (string1.rx.dx, string2.rx.dx)
0000 114     : TRUE if string1 >= string2, FALSE otherwise
0000 115     :
0000 116     : condition.wlu.v = FOR$LGT (string1.rx.dx, string2.rx.dx)
0000 117     : TRUE if string1 > string2, FALSE otherwise
0000 118     :
0000 119     : INPUT PARAMETERS:
0000 120     :
0000 121     : string1 - Input string passed by descriptor. Only fields
0000 122     : in descriptor looked at are DSC$W_LENGTH and
0000 123     : DSC$A_POINTER.
0000 124     :
0000 125     : string2 - Same as string1.
0000 126     :
0000 127     : IMPLICIT INPUTS:
0000 128     :
0000 129     : NONE
0000 130     :
0000 131     : OUTPUT PARAMETERS:
0000 132     :
0000 133     : NONE
0000 134     :
0000 135     : IMPLICIT OUTPUTS:
0000 136     :
0000 137     : NONE
0000 138     :

```

```

0000 139 : FUNCTION VALUE:
0000 140 :
0000 141 :     TRUE (-1) if condition is satisfied,
0000 142 :     FALSE (0) if condition is not satisfied.
0000 143 :
0000 144 : SIDE EFFECTS:
0000 145 :
0000 146 :     NONE
0000 147 :
0000 148 :--
0000 149 :
1C 000C 0000 150 : .ENTRY FOR$LLT, ^M<R2, R3> : Lexical less than
29 10 0002 151 : BSBB COMPARE : Compare the strings
2B 1F 0004 152 : BLSSU TRUE : TRUE if less than
11 0006 153 : BRB FALSE : Otherwise FALSE
0008 154 :
14 000C 0008 155 : .ENTRY FOR$LLE, ^M<R2, R3> : Lexical less than or equal
21 10 000A 156 : BSBB COMPARE : Compare the strings
23 1B 000C 157 : BLEQU TRUE : TRUE if less than or equal
11 000E 158 : BRB FALSE : Otherwise FALSE
0010 159 :
0C 000C 0010 160 : .ENTRY FOR$LGE, ^M<R2, R3> : Lexical greater than or equal
19 10 0012 161 : BSBB COMPARE : Compare the strings
1B 1E 0014 162 : BGEQU TRUE : TRUE if greater than or equal
11 0016 163 : BRB FALSE : Otherwise FALSE
0018 164 :
04 000C 0018 165 : .ENTRY FOR$LGT, ^M<R2, R3> : Lexical greater than
11 10 001A 166 : BSBB COMPARE : Compare the strings
13 1A 001C 167 : BGTRU TRUE : TRUE if greater than or equal
11 001E 168 : BRB FALSE : Otherwise FALSE
0020 169 :

```

```

0020 171      .SBTTL Local functions for lexical comparisons
0020 172
0020 173      :++
0020 174      COMPARE
0020 175      :
0020 176      COMPARE loads the two string descriptors into R0-R3,
0020 177      does a compare on the strings and returns to the local
0020 178      caller. Since the condition codes are preserved, the
0020 179      local caller may do a conditional branch for whatever
0020 180      condition is appropriate.
0020 181      :--
0020 182
0020 183 COMPARE:
63 52 20 50 04 BC 7D 0020 184 MOVQ @string1(AP), R0 ; Get first descriptor
52 61 50 08 BC 7D 0024 185 MOVQ @string2(AP), R2 ; Get second descriptor
05 002E 186 CMPC5 R0, (R1), #^A/ /, R2, (R3) ; Compare strings
002F 187 RSB ; Return to local caller
002F 188
002F 189
50 01 CE 002F 190 TRUE:
04 0032 191 MNEGL #1, R0 ; Return TRUE
0033 192 RET ; Return
0033 193
50 D4 0033 194 FALSE:
04 0035 195 CLRL R0 ; Return FALSE
0036 196 RET ; Return
0036 197
0036 198
0036 199
0036 200 .END

```



FOR\$LEX  
Symbol table

- Lexical comparison functions

L 9

15-SEP-1984 23:54:56  
6-SEP-1984 10:57:14

VAX/VMS Macro V04-00  
[FORRTL.SRC]FORLEX.MAR;1

Page 6  
(4)

COMPARE	00000020	R	01
FALSE	00000033	R	01
FOR\$LGE	00000010	RG	01
FOR\$LGT	00000018	RG	01
FOR\$LLE	00000008	RG	01
FOR\$LLT	00000000	RG	01
STRING1	= 00000004		
STRING2	= 00000008		
TRUE	0000002F	R	01

-----  
! Psect synopsis !  
-----

PSECT name

Allocation

PSECT No.

Attributes

ABS  
\_FOR\$CODE

00000000 ( 0.)  
00000036 ( 54.)

00 ( 0.)  
01 ( 1.)

NOPIC USR  
PIC USR

CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE  
CON REL LCL SHR EXE RD NOWRT NOVEC LONG

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.06	00:00:01.57
Command processing	113	00:00:00.43	00:00:02.64
Pass 1	70	00:00:00.48	00:00:02.62
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	51	00:00:00.47	00:00:01.67
Symbol table output	2	00:00:00.03	00:00:00.03
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	270	00:00:01.49	00:00:08.55

The working set limit was 900 pages.  
2198 bytes (5 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 9 non-local and 0 local symbols.  
200 source lines were read in Pass 1, producing 19 object records in Pass 2.  
0 pages of virtual memory were used to define 0 macros.

-----  
! Macro library statistics !  
-----

Macro library name

Macros defined

\_\$255\$DUA28:[SYSLIB]STARLET.MLB;2

0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:FORLEX/OBJ=OBJ\$:FORLEX MSRC\$:FORLEX/UPDATE=(ENH\$:FORLEX)

FORINTLND LIS

FORMSG LIS

FORIOBEG LIS

FORTOEND LIS

FORLEX LIS

FORMLTAB LIS

FORINQUIR LIS

FORIOELEM LIS

FORDATE LIS

FORLIB LIS