


```
MM      MM      TTTTTTTTTT  HH      HH      FFFFFFFFFF  LL      000000  000000  RRRRRRRR  
MM      MM      TTTTTTTTTT  HH      HH      FFFFFFFFFF  LL      000000  000000  RRRRRRRR  
MMMM    MMMM    TT          HH      HH      FF          LL      00      00      RR      RR  
MMMM    MMMM    TT          HH      HH      FF          LL      00      00      RR      RR  
MM  MM  MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM  MM  MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HHHHHHHHHH  FFFFFFFF  LL      00      00      RRRRRRRR  
MM      MM      TT          HHHHHHHHHH  FFFFFFFF  LL      00      00      RRRRRRRR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LL      00      00      RR      RR  
MM      MM      TT          HH      HH      FF          LLLLLLLLLL  000000  000000  RR      RR  
MM      MM      TT          HH      HH      FF          LLLLLLLLLL  000000  000000  RR      RR
```

```
LL      IIIIII  SSSSSSSS  
LL      IIIIII  SSSSSSSS  
LL      II     SS  
LL      II     SS  
LL      II     SS  
LL      II     SS  
LL      II     SSSSSS  
LL      II     SSSSSS  
LL      II     SS  
LL      II     SS  
LL      II     SS  
LL      II     SS  
LLLLLLLL  IIIIII  SSSSSSSS  
LLLLLLLL  IIIIII  SSSSSSSS
```

(2) 53
(3) 83
(4) 140

DECLARATIONS
MTHSFLOOR - greatest integer floating routine
MTHSFLOOR_R1 - JSB entry point

```
0000 1 .TITLE MTH$FLOOR - Greatest integer floating routine
0000 2 .IDENT /1-006/ ; File: MTHFLOOR.MAR
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 9 * ALL RIGHTS RESERVED. *
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 16 * TRANSFERRED. *
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 20 * CORPORATION. *
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: Math Library
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 This routine finds the largest integer less than the input
0000 35 value, i.e. it truncates toward negative infinity
0000 36 for type float.
0000 37
0000 38 ENVIRONMENT: User Mode, AST Reentrant
0000 39
0000 40 --
0000 41 AUTHOR:R. Will, CREATION DATE: 1-Dec-78
0000 42
0000 43 MODIFIED BY:
0000 44
0000 45 VERSION 00
0000 46 1-001 - Original
0000 47 1-002 - Add "" to the PSECT directive. JBS 22-DEC-78
0000 48 1-003 - Put MTH$AINT code in line. RW 26-Mar-79
0000 49 1-004 - Correct bug for -1 < input < 0. RW 11-Jul-79
0000 50 1-005 - Add a JSB entry point. JBS 25-JUL-1979
0000 51 1-006 - Change name to MTH$FLOOR. JBS 27-JUL-1979
```

```
0000 53      .SBTTL  DECLARATIONS
0000 54      :
0000 55      : INCLUDE FILES:
0000 56      :
0000 57      :
0000 58      :
0000 59      : EXTERNAL DECLARATIONS:
0000 60      :
0000 61      :      .DSABL  GBL
0000 62      :
0000 63      :
0000 64      :
0000 65      : MACROS:
0000 66      :
0000 67      :
0000 68      :
0000 69      : EQUATED SYMBOLS:
0000 70      :
0000 71      :
0000 72      :
0000 73      : OWN STORAGE:
0000 74      :
0000 75      :
0000 76      :
0000 77      : PSECT DECLARATIONS:
0000 78      :
00000000 79      :      .PSECT  _MTH$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 80      :      EXE, RD, NOWRT, LONG
0000 81
```

```
: Prevent undeclared
: symbols from being
: automatically global.
```

```

0000 83      .SBTTL MTHSFLOOR - greatest integer floating routine
0000 84      :++
0000 85      : FUNCTIONAL DESCRIPTION:
0000 86      :
0000 87      :     This routine finds the floor by truncating, and then if the
0000 88      :     input value is negative and not an integer subtracting 1.
0000 89      :
0000 90      : CALLING SEQUENCE:
0000 91      :
0000 92      :     CALL result_int.wf.v = MTHSFLOOR (input.rf.r)
0000 93      :
0000 94      : INPUT PARAMETERS:
0000 95      :
00000004 0000 96      :     input_addr = 4
0000 97      :
0000 98      : IMPLICIT INPUTS:
0000 99      :
0000 100     :     NONE
0000 101     :
0000 102     : OUTPUT PARAMETERS:
0000 103     :
0000 104     :     NONE
0000 105     :
0000 106     : IMPLICIT OUTPUTS:
0000 107     :
0000 108     :     NONE
0000 109     :
0000 110     : FUNCTION VALUE:
0000 111     : COMPLETION CODES:
0000 112     :
0000 113     :     the floating value of the greatest integer
0000 114     :
0000 115     : SIDE EFFECTS:
0000 116     :
0000 117     :     NONE
0000 118     :
0000 119     :--
0000 120     :
0000 121     .ENTRY MTHSFLOOR, ^M<>           ; entry point
0002 122     :
51 51 08 50 04 BC 50 0002 123     MOVF @input_addr(AP), R0           ; R0 = arg
0006 124     EMOVF R0, #0, #1, R1, R1       ; R1 = fraction_part(R0)
000C 125     SUBF R1, R0
000F 126     :
000F 127     BGTR 40$                        ; if > 0, have correct answer
0011 128     :
0011 129     TSTF R1                          ; look at fraction part
0013 130     BGEQ 40$                        ; if > 0, 0 < input < 1 and
0015 131     :                               ; we have the correct answer
0015 132     :                               ; if = 0, input was integer and
0015 133     :                               ; we have the correct answer
0015 134     :
50 08 42 0015 135     SUBF2 #1, R0          ; subtract 1 from truncated
0018 136     :                               ; negative non-integer
0018 137     :
04 0018 138 40$: RET

```

```

0019 140 .SBTTL MTHSFLOOR_R1 - JSB entry point
0019 141 :++
0019 142 : FUNCTIONAL DESCRIPTION:
0019 143 :
0019 144 : This is the JSB entry point to MTHSFLOOR.
0019 145 :
0019 146 : CALLING SEQUENCE:
0019 147 :
0019 148 : JSB result_int.wf.v = MTHSFLOOR_R1 (input.rf.v)
0019 149 :
0019 150 : INPUT PARAMETERS:
0019 151 :
0019 152 : R0 contains the input value
0019 153 :
0019 154 : IMPLICIT INPUTS:
0019 155 :
0019 156 : NONE
0019 157 :
0019 158 : OUTPUT PARAMETERS:
0019 159 :
0019 160 : NONE
0019 161 :
0019 162 : IMPLICIT OUTPUTS:
0019 163 :
0019 164 : NONE
0019 165 :
0019 166 : FUNCTION VALUE:
0019 167 : COMPLETION CODES:
0019 168 :
0019 169 : the floating value of the greatest integer
0019 170 :
0019 171 : SIDE EFFECTS:
0019 172 :
0019 173 : NONE
0019 174 :
0019 175 :--
0019 176
0019 177 MTHSFLOOR_R1:: ; entry point
0019 178
51 51 08 00 50 54 0019 179 EMOVF R0, #0, #1, R1, R1 ; R1 = fraction_part(R0)
50 51 42 001F 180 SUBF R1, R0
0022 181
0022 182 BGTR 40$ ; if > 0, have correct answer
0024 183
0024 184 TSTF R1 ; look at fraction part
03 18 0026 185 BGEQ 40$ ; if > 0, 0 < input < 1 and
0028 186 ; we have the correct answer
0028 187 ; if = 0, input was integer and
0028 188 ; we have the correct answer
0028 189
50 08 42 0028 190 SUBF2 #1, R0 ; subtract 1 from truncated
0028 191 ; negative non-integer
0028 192
0028 193 40$: RSB
002C 194
002C 195 .END

```

MTHSFLOOR
Symbol table

- Greatest integer floating routine^{J 4}

16-SEP-1984 01:24:03
6-SEP-1984 11:23:13

VAX/VMS Macro V04-00
[MTHRTL.SRC]MTHFLOOR.MAR;1

Page 5
(4)

INPUT_ADDR = 00000004
MTHSFLOOR 00000000 RG 01
MTHSFLOOR_R1 00000019 RG 01

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_MTHSCODE	0000002C (44.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.48
Command processing	115	00:00:00.51	00:00:03.55
Pass 1	70	00:00:00.51	00:00:02.66
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	46	00:00:00.38	00:00:01.18
Symbol table output	2	00:00:00.00	00:00:00.01
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	266	00:00:01.52	00:00:07.92

The working set limit was 900 pages.
2069 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 3 non-local and 2 local symbols.
195 source lines were read in Pass 1, producing 11 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\$:MTHFLOOR/OBJ=OBJ\$:MTHFLOOR MSRC\$:MTHFLOOR/UPDATE=(ENH\$:MTHFLOOR)

This image displays a grid of 150 small, low-resolution screenshots of various software interfaces, likely from the VAX/VMS operating system. Each screenshot shows a different screen or menu, often with a title bar indicating the application name. The titles are as follows:

- MTHGCONJG LIS
- MTHGINT LIS
- MTHGMOD LIS
- MTHEXP LIS
- MTHFLOOR LIS
- MTHDTAN LIS
- MTHDTANH LIS
- MTHGEXP LIS
- MTHGFLOOR LIS
- MTHGMINI LIS
- MTHGCOSH LIS
- MTHGLOG LIS
- MTHGCOS LIS
- MTHGASIN LIS
- MTHGATAN LIS
- MTHGATANH LIS
- MTHGMAXI LIS
- MTHGNINT LIS

The screenshots contain various data visualizations, including text-based tables, bar charts, and menu structures, representing different functional areas of the software suite.