


```

MM      MM      TTTTTTTTTT  HH      HH  88888888  IIIIII  TTTTTTTTTT  000000  PPPPPPPP  SSSSSSSS
MM      MM      TTTTTTTTTT  HH      HH  88888888  IIIIII  TTTTTTTTTT  000000  PPPPPPPP  SSSSSSSS
MMMM    MMMM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MMMM    MMMM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HHHHHHHHHH  88888888  III      TT      00      00  PPPPPPPP  SSSSSS
MM      MM      TT      HHHHHHHHHH  88888888  III      TT      00      00  PPPPPPPP  SSSSSS
MM      MM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HH      HH  88      88      II      TT      00      00  PP      PP  SS
MM      MM      TT      HH      HH  88888888  IIIIII  TT      000000  PP      SSSSSSSS
MM      MM      TT      HH      HH  88888888  IIIIII  TT      000000  PP      SSSSSSSS

```

```

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
LLLLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLLLL  IIIIII  SSSSSSSS

```

(2)	50	HISTORY	: Detailed Current Edit History
(3)	67	DECLARATIONS	
(4)	98	MTH\$IIAND	
(5)	136	MTH\$JIAND	
(6)	175	MTH\$IIOR	
(7)	213	MTH\$JIOR	
(8)	251	MTH\$IIEOR	
(9)	289	MTH\$JIEOR	
(10)	327	MTH\$INOT	
(11)	365	MTH\$JNOT	
(12)	403	MTH\$IISHFT	
(13)	442	MTH\$JISHFT	

```
0000 1 .TITLE MTH$BITOPS ROUTINES FOR BIT OPERATIONS
0000 2 .IDENT /1-005/ ; File: MTHBITOPS.MAR Edit: JAW1005
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 : FACILITY: MATH LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 : This module contains routines for operations on individual
0000 33 : bits of arguments.
0000 34 :
0000 35 :
0000 36 :--
0000 37 :
0000 38 : VERSION: 0
0000 39 :
0000 40 : HISTORY:
0000 41 :
0000 42 : AUTHOR:
0000 43 : Jonathan M. Taylor, 14-JUL-77: Version 0
0000 44 :
0000 45 : MODIFIED BY:
0000 46 :
0000 47 :
0000 48 :
```

M1
S)
M1
M1
M1
M1
M1
M1
M1
M1
M1
PS
--
-M
Pr
--
Ir
Cc
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
47
Th
47
0
Ma
--
-1
0
Th
MA

```
000C 50 .SBTTL HISTORY ; Detailed Current Edit History
0000 51
0000 52
0000 53 ; Edit History for Version 0 of MTH$BITOPS
0000 54 :
0000 55 : 1-001 - Update version number and copyright notice. The last
0000 56 : 1-002 - Add "" to the PSECT directive. JBS 21-DEC-78
0000 57 : edit number for version 0 was 3. JBS 16-NOV-78
0000 58 : 1-003 - Fix MTH$IIAND, MTH$JIAND. SPR 11-25743 SBL 8-Aug-1979
0000 59 : 1-004 - Add routines to support additional functions defined in
0000 60 : MIL-STD 1753 for Fortran: MTH$IMVBITS, MTH$JMVBITS,
0000 61 : MTH$IIBITS, MTH$JIBITS, MTH$IISHFTC, MTH$JISHFTC, MTH$BITEST,
0000 62 : MTH$BJTEST, MTH$IIBSET, MTH$JIBSET, MTH$IIBCLR, MTH$JIBCLR.
0000 63 : JAW 21-May-1981
0000 64 : 1-005 - Remove the routines added in edit 1-004 and place them in
0000 65 : FOR$BITOPS as FOR$ routines. JAW 06-Jun-1981
```

```
0000 67          .SBTTL  DECLARATIONS
0000 68
0000 69 :
0000 70 : INCLUDE FILES:
0000 71 :         oerr.mar
0000 72 :
0000 73 :
0000 74 :
0000 75 : EXTERNAL SYMBOLS:
0000 76 :         NONE
0000 77 :
0000 78 :
0000 79 :
0000 80 : MACROS:
0000 81 :         NONE
0000 82 :
0000 83 :
0000 84 :
0000 85 : PSECT DECLARATIONS:
0000 86 :         .PSECT  _MTH$CODE          PIC, SHR, LONG, EXE, NOWRT
0000 87 :
0000 88 :
0000 89 : EQUATED SYMBOLS:
0000 90 :         NONE
0000 91 :
0000 92 :
0000 93 :
0000 94 : OWN STORAGE:
0000 95 :         NONE
0000 96 :
```

```

0000 98      .SBTTL MTH$IIAND
0000 99
0000 100    :++
0000 101    : FUNCTIONAL DESCRIPTION:
0000 102    :   Return the bitwise AND of two one-word arguments.
0000 103    :
0000 104    :
0000 105    : CALLING SEQUENCE:
0000 106    :   Bitwise_AND.wv.v = MTH$IIAND (arg1.rw.r, arg2.rw.r)
0000 107    :
0000 108    :
0000 109    :
0000 110    : INPUT PARAMETERS:
0000 111    :   Both arguments are one-word values and are call-by-reference.
0000 112    :
0000 113    :
0000 114    : IMPLICIT INPUTS:
0000 115    :   NONE
0000 116    :
0000 117    : OUTPUT PARAMETERS:
0000 118    :   NONE
0000 119    :
0000 120    : IMPLICIT OUTPUTS:
0000 121    :   NONE
0000 122    :
0000 123    : COMPLETION CODES:
0000 124    :   NONE
0000 125    :
0000 126    : SIDE EFFECTS:
0000 127    :   NONE
0000 128    :
0000 129    :--
0000 130
0000 131    .ENTRY MTH$IIAND, ^M<>
50   50   04 BC 0000 0002 132    MCOMW @4(AP), R0 ; R0 = complemented first arg
    08 BC 50  AB 0006 133    BICW3 R0, @8(AP), R0 ; R0 = ANDED args
    04 000B 134    RET

```

```

000C 136      .SBTTL  MTH$JIAND
000C 137
000C 138      :++
000C 139      : FUNCTIONAL DESCRIPTION:
000C 140      :   Return the bitwise AND of two longword arguments.
000C 141      :
000C 142      :
000C 143      : CALLING SEQUENCE:
000C 144      :   Bitwise_AND.wl.v = MTH$JIAND (arg1.rl.r, arg2.rl.r)
000C 145      :
000C 146      :
000C 147      :
000C 148      : INPUT PARAMETERS:
000C 149      :   The two parameters are longword values and are call-by-reference.
000C 150      :
000C 151      :
000C 152      : IMPLICIT INPUTS:
000C 153      :   NONE
000C 154      :
000C 155      : OUTPUT PARAMETERS:
000C 156      :   NONE
000C 157      :
000C 158      : IMPLICIT OUTPUTS:
000C 159      :   NONE
000C 160      :
000C 161      : COMPLETION CODES:
000C 162      :   NONE
000C 163      :
000C 164      : SIDE EFFECTS:
000C 165      :   NONE
000C 166      :
000C 167      :--
000C 168
000C 169
000C 170      .ENTRY  MTH$JIAND,      ^M<>
50   50   04  BC  0000 000C 171      MCOML  @4(AP), R0      ; R0 = complemented first arg
08   08  BC   50   D2  000E 172      BICL3  R0, @8(AP), R0    ; R0 = ANDED args
04   04          CB  0012 173      RET
04   04          04  0017

```



```

0018 175      .SBTTL MTH$I IOR
0018 176
0018 177      :++
0018 178      : FUNCTIONAL DESCRIPTION:
0018 179      : Return the bitwise inclusive OR of two one-word arguments.
0018 180      :
0018 181      :
0018 182      : CALLING SEQUENCE:
0018 183      : Bitwise_inclusive_OR.ww.v = MTH$I IOR (arg1.rw.r, arg2.rw.r)
0018 184      :
0018 185      :
0018 186      : INPUT PARAMETERS:
0018 187      : The two parameters are one-word values and are call-by-reference.
0018 188      :
0018 189      :
0018 190      : IMPLICIT INPUTS:
0018 191      : NONE
0018 192      :
0018 193      : OUTPUT PARAMETERS:
0018 194      : NONE
0018 195      :
0018 196      : IMPLICIT OUTPUTS:
0018 197      : NONE
0018 198      :
0018 199      : COMPLETION CODES:
0018 200      : NONE
0018 201      :
0018 202      : SIDE EFFECTS:
0018 203      : NONE
0018 204      :
0018 205      :
0018 206      :--
0018 207
0018 208
50 08 BC 04 BC 0000 0018 209      .ENTRY MTH$I IOR,      ^M<>
A9 001A 210      BISW3 @4(AP), @8(AP), R0 ; R0 = ORed args
04 0020 211      RET

```

```

0021 213      .SBTTL MTH$JIOR
0021 214
0021 215      :++
0021 216      : FUNCTIONAL DESCRIPTION:
0021 217      :   Return the bitwise inclusive OR of two longword arguments.
0021 218      :
0021 219      :
0021 220      : CALLING SEQUENCE:
0021 221      :   Bitwise_inclusive_OR.wl.v = MTH$JIOR (arg1.rl.r, arg2.rl.r)
0021 222      :
0021 223      :
0021 224      :
0021 225      : INPUT PARAMETERS:
0021 226      :   The two parameters are longword values and are call-by-reference.
0021 227      :
0021 228      :
0021 229      : IMPLICIT INPUTS:
0021 230      :   NONE
0021 231      :
0021 232      : OUTPUT PARAMETERS:
0021 233      :   NONE
0021 234      :
0021 235      : IMPLICIT OUTPUTS:
0021 236      :   NONE
0021 237      :
0021 238      : COMPLETION CODES:
0021 239      :   NONE
0021 240      :
0021 241      : SIDE EFFECTS:
0021 242      :   NONE
0021 243      :
0021 244      :--
0021 245
0021 246
50  08 BC  04 BC  0000 0021 247      .ENTRY MTH$JIOR,      ^M<>
          C9 0023 248      BISL3 @4(AP), @8(AP), R0 ; R0 = ORed args
          04 0029 249      RET

```

```

002A 251      .SBTTL MTH$IIEOR
002A 252
002A 253      :++
002A 254      : FUNCTIONAL DESCRIPTION:
002A 255      : Return the bitwise XOR of two one-word arguments.
002A 256      :
002A 257      :
002A 258      : CALLING SEQUENCE:
002A 259      : Bitwise_exclusive_OR.wv.v = MTH$IIEOR (arg1.rw.r, arg2.rw.r)
002A 260      :
002A 261      :
002A 262      :
002A 263      : INPUT PARAMETERS:
002A 264      : The two parameters are one-word values and are call-by-reference.
002A 265      :
002A 266      :
002A 267      : IMPLICIT INPUTS:
002A 268      : NONE
002A 269      :
002A 270      : OUTPUT PARAMETERS:
002A 271      : NONE
002A 272      :
002A 273      : IMPLICIT OUTPUTS:
002A 274      : NONE
002A 275      :
002A 276      : COMPLETION CODES:
002A 277      : NONE
002A 278      :
002A 279      : SIDE EFFECTS:
002A 280      : NONE
002A 281      :
002A 282      :--
002A 283
002A 284
002A 285      .ENTRY MTH$IIEOR, ^M<>
50 08 BC 04 BC 0000 002A 286      XORW3 @4(AP), @8(AP), R0 ; R0 = XORed args
AD 002C 287      RET
04 0032

```

MT
Sy
AR
MT
MT
MT
RE
ZE

PS
--
_M

Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
25
Th
21
1

Ma
--
_S
O
Th
MA

```
0033 289      .SBTTL MTH$JIEOR
0033 290
0033 291      :++
0033 292      : FUNCTIONAL DESCRIPTION:
0033 293      :   Return the bitwise XOR of two longword arguments.
0033 294      :
0033 295      :
0033 296      : CALLING SEQUENCE:
0033 297      :   Bitwise_exclusive_or.wl.v = MTH$JIEOR (arg1.rl.r, arg2.rl.r)
0033 298      :
0033 299      :
0033 300      :
0033 301      : INPUT PARAMETERS:
0033 302      :   The two arguments are longword values and are call-by-reference.
0033 303      :
0033 304      :
0033 305      : IMPLICIT INPUTS:
0033 306      :   NONE
0033 307      :
0033 308      : OUTPUT PARAMETERS:
0033 309      :   NONE
0033 310      :
0033 311      : IMPLICIT OUTPUTS:
0033 312      :   NONE
0033 313      :
0033 314      : COMPLETION CODES:
0033 315      :   NONE
0033 316      :
0033 317      : SIDE EFFECTS:
0033 318      :   NONE
0033 319      :
0033 320      :--
0033 321
0033 322
0033 323      .ENTRY MTH$JIEOR, ^M<>
50 08 BC 04 BC 0000 0033 324      XORL3 @4(AP), @8(AP), R0 ; R0 = XORed args
04 0038 325      RET
```

```

003C 327      .SBTTL MTH$INOT
003C 328
003C 329      :++
003C 330      : FUNCTIONAL DESCRIPTION:
003C 331      :   Return the bitwise NOT of a one-word argument.
003C 332      :
003C 333      :
003C 334      : CALLING SEQUENCE:
003C 335      :   Bitwise_complement.ww.v = MTH$INOT (arg.rw.r)
003C 336      :
003C 337      :
003C 338      : INPUT PARAMETERS:
003C 339      :   The one parameter is a one-word value and is call-by-reference.
003C 340      :
003C 341      :
003C 342      : IMPLICIT INPUTS:
003C 343      :   NONE
003C 344      :
003C 345      :
003C 346      : OUTPUT PARAMETERS:
003C 347      :   NONE
003C 348      :
003C 349      : IMPLICIT OUTPUTS:
003C 350      :   NONE
003C 351      :
003C 352      : COMPLETION CODES:
003C 353      :   NONE
003C 354      :
003C 355      : SIDE EFFECTS:
003C 356      :   NONE
003C 357      :
003C 358      :--
003C 359
003C 360
50  04 BC 0000 003C 361      .ENTRY MTH$INOT,      ^M<>
      B2  003E 362      MCOMW @4(AP), R0      ; R0 = complemented arg
      04  0042 363      RET

```

```

0043 365      .SBTTL MTH$JNOT
0043 366
0043 367 :++
0043 368 : FUNCTIONAL DESCRIPTION:
0043 369 :   Return the bitwise NOT of a longword argument.
0043 370 :
0043 371 :
0043 372 : CALLING SEQUENCE:
0043 373 :   Bitwise_complement.wl.v=MTH$JNOT (arg.rl.r)
0043 374 :
0043 375 :
0043 376 :
0043 377 : INPUT PARAMETERS:
0043 378 :   The one parameter is a longword value and is call-by-reference.
0043 379 :
0043 380 :
0043 381 : IMPLICIT INPUTS:
0043 382 :   NONE
0043 383 :
0043 384 : OUTPUT PARAMETERS:
0043 385 :   NONE
0043 386 :
0043 387 : IMPLICIT OUTPUTS:
0043 388 :   NONE
0043 389 :
0043 390 : COMPLETION CODES:
0043 391 :   NONE
0043 392 :
0043 393 : SIDE EFFECTS:
0043 394 :   NONE
0043 395 :
0043 396 :--
0043 397
0043 398
0043 399      .ENTRY MTH$JNOT,      ^M<>
50 04 BC 0000 0043 399      MCOML @4(AP), R0      ; R0 = complemented arg
0043 400      RET
0043 401

```

```

004A 403      .SBTTL MTH$IISHFT
004A 404
004A 405      :++
004A 406      : FUNCTIONAL DESCRIPTION:
004A 407      :   Return the result of shifting arg1 by arg2 places.
004A 408      :
004A 409      :
004A 410      : CALLING SEQUENCE:
004A 411      :   Bitwise_shift.wv.v = MTH$IISHFT (arg1.rwu.r, shiftcount.rw.r)
004A 412      :
004A 413      :
004A 414      :
004A 415      : INPUT PARAMETERS:
004A 416      :   Both arguments are one-word values and are call-by-reference.
004A 417      :
004A 418      :
004A 419      : IMPLICIT INPUTS:
004A 420      :   NONE
004A 421      :
004A 422      : OUTPUT PARAMETERS:
004A 423      :   NONE
004A 424      :
004A 425      : IMPLICIT OUTPUTS:
004A 426      :   NONE
004A 427      :
004A 428      : COMPLETION CODES:
004A 429      :   NONE
004A 430      :
004A 431      : SIDE EFFECTS:
004A 432      :   NONE
004A 433      :
004A 434      :--
004A 435
004A 436
004A 437      .ENTRY MTH$IISHFT,      ^M<>
50 50 04 BC 0000 004A 438      MOVZWL @4(AP), R0      ; R0 = zero-extended arg
004A 439      ASHL @8(AP), R0, R0 ; so right shift brings in zeroes
004A 440      RET

```

```

0056 442      .SBTTL MTH$JISHFT
0056 443
0056 444 :++
0056 445 : FUNCTIONAL DESCRIPTION:
0056 446 :
0056 447 :     Return the result of shifting arg1 by arg2 places
0056 448 : CALLING SEQUENCE:
0056 449 :     Bitwise-shift.wl.v = MTH$JISHFT (arg1.rlu.r, shiftcount.rl.r)
0056 450 :
0056 451 : INPUT PARAMETERS:
0056 452 :     Both arguments are longword values and are call-by-references.
0056 453 :
0056 454 : IMPLICIT INPUTS:
0056 455 :     NONE
0056 456 :
0056 457 : OUTPUT PARAMETERS:
0056 458 :     NONE
0056 459 :
0056 460 : IMPLICIT OUTPUTS:
0056 461 :     NONE
0056 462 :
0056 463 : COMPLETION CODES:
0056 464 :     NONE
0056 465 :
0056 466 : SIDE EFFECTS:
0056 467 :     NONE
0056 468 :
0056 469 :
0056 470 :--
0056 471
0056 472
0056 473      .ENTRY MTH$JISHFT,      ^M<>
50 04 BC 0000 0058 474      MOVL @4(AP), R0      ; R0 = arg to shift
50 50 08 BC 04 005C 475      CLRL R1              ; R1 = 0 so right shift brings in zeroes
005E 476      ASHQ @8(AP), R0, R0 ; shift R0 by arg2 bits
0063 477      RET
0064 478
0064 479      .END

```


MTH\$BITOPS
Symbol table

ROUTINES FOR BIT OPERATIONS

L 14

16-SEP-1984 01:05:22 VAX/VMS Macro V04-00 Page 14
6-SEP-1984 11:20:41 [MTHRTL.SRC]MTHBITOPS.MAR;1 (13)

MT
i-

MTH\$IAND	00000000	RG	01
MTH\$IIEOR	0000002A	RG	01
MTH\$IIR	00000018	RG	01
MTH\$IISHFT	0000004A	RG	01
MTH\$INOT	0000003C	RG	01
MTH\$JAND	0000000C	RG	01
MTH\$JIEOR	00000033	RG	01
MTH\$JIOR	00000021	RG	01
MTH\$JISHFT	00000056	RG	01
MTH\$JNOT	00000043	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
_MTH\$CODE	00000064 (100.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.06	00:00:01.93
Command processing	138	00:00:00.49	00:00:02.89
Pass 1	76	00:00:00.77	00:00:04.10
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	83	00:00:00.86	00:00:04.34
Symbol table output	2	00:00:00.01	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	340	00:00:02.21	00:00:13.29

The working set limit was 750 pages.
4700 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 10 non-local and 0 local symbols.
479 source lines were read in Pass 1, producing 37 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\$:MTHBITOPS/OBJ=OBJ\$:MTHBITOPS MSRC\$:MTHBITOPS/UPDATE=(ENH\$:MTHBITOPS)

MTH4OVP LIS	MTHABS LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHERR SOL	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHASIN LIS	MTHABSL LIS	MTHABSL LIS	MTHCDABS LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHATAN LIS	MTHABSL LIS	MTHATANH LIS	MTHCDLOG LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHJACKET MAR	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHBITOPS LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHALOG LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHDEF FOR	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHANT LIS	MTHABSL LIS	MTHABSL LIS	MTHCABS LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS
MTHABSL LIS	MTHABSL LIS	MTHACOS LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHABSL LIS	MTHCDEXP LIS	MTHABSL LIS	MTHABSL LIS