
M* (**n1 n2 -- d1**) **"m times"**

Multiply two signed single precision numbers N1 and N2 together and leave the signed double product D1.

M/ (**d n1 -- rem n2**) **"m slash"**

M/ divides a signed divisor N1 into a signed double dividend D to yield a signed remainder REM and the signed quotient N2.

M/MOD (**d1 n1 -- rem d2**) **"m slash mod"**

Divides a signed value N1 into a signed double-value D1 and returns the signed remainder REM and a signed double quotient D2.

MAKEMODULE (**size -- , <module name>**)

See the chapter on MODULES.

MAKEUCASE (**-- addr**)

This is a user-variable which controls whether WORD will convert all alphabetic characters to upper-case.

Normally, all dictionary headers are created by WORD. FIND for reasons of speed is CASE-SENSITIVE. Therefore, all words, by convention, are converted to upper-case as they are moved to HERE, by WORD.

The programmer, should he/she wish to receive lower-case strings at HERE, may either set (and restore when done) the contents of MAKEUCASE himself, or use the supplied version LWORD, which does this automatically.

```
MAKEUCASE @ FALSE MAKEUCASE !
PROGRAMMERS-GET-WORD
MAKEUCASE !
```

Related Words: WORD LWORD HERE

MAP (**--**)

MAP displays general information concerning current memory allocation and dictionary utilization, number of vocabularies, relocation information, file usage, the setting of the current MAX-INLINE variable, the state of the modules, and whether the HASHed vocabulary search is in use..

MARKFCLOSE (**file-pointer --**) **"mark f close"**

MARKFCLOSE places the FILE-POINTER in a list of file-pointers that will be closed if the system executes QUIT.

This mechanism allows an application to automatically clean up its resources in both of the two possible paths most programs may take... normal completion or an irrecoverable error abort.

Note that if the application terminates normally, it should clear this AUTO-CLOSE status for the file by executing UNMARKFCLOSE for each file it had previously marked.

```
FOPEN ?DUP          \ get the filename, did it open?
IF DUP MYFILE ! \ yes, save pointer in my variable
MARKFCLOSE          \ set it to auto-close at quit
RUN-MY-PROGRAM \ do whatever I want to do with the file
MYFILE @ DUP        \ need to do TWO thing to the pointer...
UNMARKFCLOSE        \ remove it from the QUIT list...
FCLOSE              \ and close the file
```

```
ELSE ." Can't open the file!"
THEN
```

Related Words: UNMARKFCLOSE

MARKFREEBLOCK (memory-block --)

MARKFREEBLOCK places the memory-block on a list of memory-blocks that will be closed if the system executes QUIT .

This mechanism allows an application to automatically 'clean-up' its resources in both of the two possible paths most programs may take... normal completion or an irrecoverable error abort.

Note that if the application terminates normally, it should clear this AUTO-FREE status for the memory-block by executing UNMARKFREEBLOCK for each memory-block it had previously marked.

```
0 1024 ALLOCBLOCK ?DUP \ allocate 1K, did it allocate?
IF DUP MYMEM !      \ yes, save pointer in my variable
  MARKFREEBLOCK     \ set it to auto-free at quit
  RUN-MY-PROGRAM     \ do whatever I want to do with the mem
  MYMEM @ DUP        \ need to do TWO thing to the pointer...
  UNMARKFREEBLOCK    \ remove it from the QUIT list...
  FREEBLOCK          \ and free the memory
ELSE ." can't allocate memory "
THEN
```

Related Words: UNMARKFREEBLOCK

MATCH? (addr1 count1 addr2 count2 -- addr | FALSE)

Search for string2 within a larger string1, leaving the address within string1 that matches if found, or FALSE if not found.

By setting contents of the USER variable MCASE-SENSITIVE true, a programmer can make the search be case-sensitive.

```
: LOOKFOR ( $string -- addr )
  COUNT ( get address and count of big string )
  " RAIN" COUNT MATCH? ;
" THE RAIN IN SPAIN" LOOKFOR
```

Related Words: TEXT=? \$= MCASE-SENSITIVE COMPARE

MATHFFP?

```
MATHFFP_LIB
MATHFFP_NAME
```

Standards: JForth unique.

These are used to manage to MATHFFP library.

MATHIEEDOUBBAS?

```
MATHIEEDOUBBAS_LIB
MATHIEEDOUBBAS_NAME
```

Standards: JForth unique.

These are used to manage the MATHIEEDOUBBAS library. See :LIBRARY.

MATHIEESINGBAS?

MATHIEESINGBAS_LIB
MATHIEESINGBAS_NAME

Standards: JForth unique

These are used to manage the MATHIEESINGBAS library. See :LIBRARY.

MATHTRANS?

MATHTRANS_LIB
MATHTRANS_NAME

Standards: JForth unique.

These are used to manage the MATHTRANS library. See :LIBRARY.

MAX (n1 n2 -- n1 | n2)

Compare the top two items on the stack, leave the largest signed value. You can use MIN and MAX to clip a value to a boundary.

```
23 7 MAX . ( print 23 )  
( make sure x not negative )  
CALC.X ( -- x ) 0 MAX SQRT
```

Related Words: MIN >

MAX-INLINE max inline

This is a USER-variable examined by the compiler when it is about to compile a reference to a word that may be either INLINE or CALLED. See (CFA,) . If a referenced word is defined as BOTH, and its size is over MAX-INLINE then a subroutine call to that word will be compiled. Otherwise it will be compiled as an inline macro. You can use this word to make trade offs between memory usage and speed. The higher MAX-INLINE is, the faster the code that is produced. Generally varying MAX-INLINE from 6-256 results in a size difference of 5-20% .

```
16 MAX-INLINE !
```

Related Words: (CFA,) BOTH INLINE

MAXVOCs max vocs

This is a system CONSTANT, equal to the maximum number of VOCABULARIES that may be defined.

Standards: JForth unique

Related Words: VOCABULARY

MCASE-SENSITIVE (-- addr)

This variable is used to control the case sensitivity of TEXT=?, a primitive for MATCH?.

See MATCH? TEXT=?.

MEASURE (<forth> --)

Measure the time it takes to execute the following code and print the time. This is handy for benchmarking code. Use BENCH if you are benchmarking things in a loop and need to subtract the overhead time.

```
\ Time how long it takes to compile it.  
MEASURE INCLUDE JU:DEBUGGER
```

MEMF_CHIP	
MEMF_CLEAR	
MEMF_FAST	
MEMF_PUBLIC	
AMIGA Constants. See ALLOCBLOCK.	
MENU.SETUP	(0name menu# menu-addr --)
Initialize an Intuition Menu structure to reasonable defaults. See the chapter on Amiga Menus.	
Related Words:	EZMENU
MIN	(n1 n2 -- n1 n2)
Compares the top two stack values n1 and n2, leaving the smallest value.	
(limit your deduction)	
TAX-DEDUCTION @ MAX-ALLOWED @ MIN	
Related Words:	MAX
MOD	(n1 n2 -- rem) "mod" or "modulus"
Leaves signed remainder REM on top of the stack after dividing N1 by N2.	
13 5 MOD . (print 3)	
Related Words:	
/MOD */MOD M/MOD	
MODE_NEWFILE	MODE_OLDFILE
See FOPEN.	
MOVE	(addr1 addr2 u --)
Copies U bytes in memory starting at address ADDR1 to memory starting at address ADDR2.	
MOVE will decide whether to copy from top to bottom, or vice versa, if the destination overlaps the source area. This will avoid the problem of data overwriting itself which can happen with CMOVE.	
MOVE is also an optimized data transfer; under the right conditions, it will move data in excess of 1 megabyte/second on a standard Amiga 500.	
Related Words:	CMOVE CMOVE> WMOVE
MSEC	(#milliseconds --)
Delay for approximately that many milliseconds. Delays over 20 milliseconds use the Amiga Delay() function. Delays less than 20 milliseconds have to use software timing and can err in accuracy. Do not use this where extreme accuracy is required. From file JU:MSEC.	
1000 MSEC (wait one second)	
MYDSP	my d s p
This USER-variable is not implemented in JForth V2.0, but is reserved for future use.	
MYRP	my r p
This USER-variable is not implemented in JForth V2.0, but is reserved for future use.	
MYTOS	(-- addr) "my tos"
This USER-variable is not implemented in JForth V2.0, but is reserved for future use.	

N>LINK (nfa -- lfa) "n to link"

Used to derive the link-field-address of a dictionary header from the name-field-address.

Related Words: LINK> >LINK ' >NAME

N>TEXT (n -- addr count) "n to text"

Convert a number to it's text equivalent in the current base. Like . except it doesn't type out the characters. This is useful when you want to write formatted files or draw numbers graphically.

```
GR.INIT GR.OPENTEST 20 30 GR.MOVE
1234 N>TEXT GR.TYPE ( draw number )
```

Related Words: # . #DIGITS BASE EMIT LOGTO

NAME> (nfa -- cfa) "name to"

NAME> converts the name-field-address of a dictionary header to the code-field-address. The code-field-address is that place in a standard dictionary entry that contains the executable code for that definition.

```
. " The most recently compiled code is "
LATEST DUP ID. NAME> DISM
```

Related Words: >NAME '

NEGATE (n -- -n)

NEGATE reverses the sign of N.

```
2 NEGATE . ( prints -2 )
```

Related Words: DNEGATE ABS XOR

NEW

(--)

This word places MODE_NEWFILE in the user-variable FILEMODE. When the next file is opened using FOPEN, it will now create a NEW file instead of opening an existing one. See the chapter on File I/O.

CAUTION: do not set the NEW mode, unless a call to FOPEN, \$FOPEN, or 0FOPEN immediately follows. Forgetting that NEW has been executed can result in the next file to be opened being inadvertently lost. The word OLD reverts filemode to MODE_OLDFILE, as do both FOPEN operations.

```
NEW FOPEN EMPTY-FILE
```

Related Words: FOPEN \$FOPEN OLD

NEXTTASK

This USER-variable is not implemented in JForth V2.0, but is reserved for future use.

NFA (cfa -- nfa)

NFA has been renamed to >NAME to comply with the '83 standard. If you want NFA load JU:MULTISTANDARD.

NIP (a b -- b)

Remove second item from stack. Equivalent to, but faster than, SWAP DROP.

Related Words: SWAP DROP

NO@ (--)

Set the mode for local variable to not automatically fetch their values. They will leave their address instead. See the section on Local Variables.

Related Words: { } YES@

NO-COMMAS (--)

Turn off the use of commas when formatting numbers for output, by storing a FALSE in the user-variable (COMMAS).

Related Words: COMMAS (COMMAS)

NOCASE (-- n)

NOCASE is a CONSTANT used to determine the desired characteristics of output ASCII case . See OUTPUT-CASE.

NOCONSOLE (-- var-addr)

See the chapter on CLONE.

NOOP (--) "no op"

This is an empty compiled definition, generally useful for disabling deferred words that have no effect on the stack.

```
DEFER MY-VECTOR ' NOOP IS MY-VECTOR
```

NOT (flag -- logically-opposite-flag)

Reverse the logic of the flag on the stack. This word acts to change a non-zero value to zero, and a zero to -1. This is a synonym for 0= .

NOTE: this is a departure from the '83 standard, which is to perform a 1's-complement operation on the flag. Load JU:MULTISTANDARD if you need the standard NOT .

```
BEGIN MONEY-GONE? NOT
WHILE SPEND-MORE
REPEAT
```

Related Words: 0= COMP

NUMBER (\$addr -- d)

Converts a character string starting at address \$ADDR into a double cell signed number using the value in BASE. The ADDR parameter points to a length byte. NUMBER is deferred and is changed when the floating point system is active.

If a decimal point is present in the string, the value in DPL will reflect the number of digits following it; otherwise DPL contains -1.

If the string cannot be converted, it is printed on the standard output device, and the ERROR exit is taken.

```
: ADDONE ( -- , convert add & print )
  BL WORD NUMBER ( -- d )
  DROP 1+ . ;
ADDONE 1234 ( prints 1235 )
```

Related Words: ERROR NUMBER? CONVERT DPL

NUMBER? (*\$addr* -- *d* *true* | *false*)

Attempt to convert the string at address *\$ADDR* to a number, using the current value of *BASE*. If successful, return the double number and *true*; otherwise return *false*.

Standards: JForth unique.

Related Words: *NUMBER* *CONVERT*

OB.ARRAY (*<name>* --)

Create an instance, or object, of the class *OB.ARRAY*. This is a powerful type of array data structure. See the chapter on ODE.

OB.ELMNTS (*<name>* --)

Create an instance, or object, of the class *OB.ELMNTS*. This is a powerful type of multidimensional array data structure. See the chapter on ODE.

OB.STATS? (*<member>* -- *offset* *#bytes*)

Returns information about a structure member. This information is used by the compiler to compile proper references to Amiga 'C' structures.

GETMODULE INCLUDES

OB.STATS? NW_WIDTH .S

OBJECT (*<name>* --)

Create a basic *OBJECT*. *OBJECT* is the root class from which all other ODE objects are derived. See the chapter on ODE.

ODE

ODE is not an executable Forth word. It stands for the Object Oriented Development System. See the chapter on ODE.

ODD! (*n* *addr* --) "odd store"

Store a cell *N* at an odd or even address *ADDR*. Uses sequential *BYTE* accesses.

Normally not needed, since all JForth cell addresses are word aligned, but must be used if your code has a possibility of doing *LONG* references to odd addresses. The 68000 will generate an address error if you access word or long data at an odd address.

Standards: JForth unique. 68000 unique. 68020 will be able to use *@*. Existence and definition depend on the host CPU type.

Related Words: *@* *!* *D!* *D@* *EVEN-UP* *ALIGN*

ODD@ (*addr* -- *n*) "odd fetch"

Fetch a cell *N* from an odd or even address *ADDR*. See comments under *ODD!* above.

71 ODD@ . (won't crash)

Related Words: *ODDW@* *ODDW!* *ODDD@* *@* *!* *D!* *D@* *EVEN-UP* *ALIGN*

ODDD! (*d* *addr* --) "odd d store"

Store a double number at an odd or even address. See comments under *ODD!* above.

ODDD@ (*addr* -- *d*) "odd d fetch"

Odd addressable *D@*. See comments under *ODD!* above.

ODDW! (**n addr --**)

Odd addressable W! See comments under ODD! above.

ODDW@ (**addr -- n**)

Odd addressable W@. See comments under ODD! above.

OF (**value n --**)

Used in conditional statement with CASE . See CASE .

OFF (**var-address --**)

Set the contents of the given address off.

FILEHEADERS OFF

Related Words: ON !

OFFSET_BEGINNING

OFFSET_CURRENT

OFFSET_END

(-- n1)

These are AMIGA constants, used to specify the type of FSEEK operation to perform. See FSEEK.

OLD (**--**)

This word places MODE_OLDFILE in the user-variable FILEMODE. This will cause FOPEN to attempt to open an existing file. See the chapter on File I/O.

OLD FOPEN AN-EXISTING-FILE

Related Words: FOPEN 0FOPEN NEW

ON (**var-address --**)

Set the contents of the given address to TRUE (-1).

VARIABLE DO-WINDOWS

DO-WINDOWS ON (set TRUE)

Related Words: OFF !

ONLY (**--**)

Remove all vocabularies from the context stack except ROOT. This is used to reset the vocabulary search path. See the section on Forth Vocabularies.

ONLY FORTH (set back to default)

Related Words: ALSO PREVIOUS FORTH ORDER

OPENFV (**var-addr -- memory-block**)

OPENFV is used to allocate a 1024-byte memory area that can later be used for a sequential virtual-buffer for fast file I/O. See the chapter on File I/O.

OR (**n1 n2 -- n3**)

Logical bit by bit or'ing of n1 and n2 . Result n3 is on top of stack . If a bit is on in either N1 OR N2 then the same bit will be on in N3.

BINARY 0011 1001 OR . (print 1011)


```
( do something if key hit or input out of range )
100 > ?TERMINAL OR IF
```

Related Words: NOT AND XOR IF

ORDER (--)

Display the order of vocabularies that will be searched by the Forth compiler. It also shows the CONTEXT vocabulary which is where newly defined words will go. See the section on Forth Vocabularies.

OUT (-- addr)

User variable that contains a value incremented by EMIT, OUT indicates the column number that the cursor is currently in. OUT is reset by CR .

```
: GOTAB ( -- , move to next tab )
  OUT @ 8 MOD
  8 SWAP - SPACES ;
```

Related Words: EMIT CR CR? >NEWLINE +OUT

OUTBUF (-- addr)

Returns the address of the output buffer being used by the FAST I/O mode. See FAST.

OUTPUT-CASE (-- var-addr)

This is a USER-variable that may be used to force characters that are EMITed to one of the desired output forms:

- 1) NOCASE - output in mixed case, no conversion (default)
- 2) LOCASE - output appears in lower case only.
- 3) HICASE - output appears in upper case only.

```
HICASE OUTPUT-CASE !
." Hello Fred" CR
```

OVER (a b -- a b a)

Copy second item on stack to top. Leapfrog.

```
11 22 OVER .S ( print 11 22 11 )
```

Related Words: DUP ROT

PAD (-- addr)

PAD returns the address of a temporary work area. PAD is defined as HERE + 128. Be careful HERE doesn't move when you are using PAD because it will move too. Immediate strings are stored at PAD. Numeric strings are built just below PAD. PAD extends up until it runs into the data stack area.

```
" Hello" . PAD .
```

Related Words: #K HERE SP@

PARSE (char -- addr count)

PARSE uses the value of >IN as an index into the address and count returned by SOURCE, parsing from that location until either CHAR is found or the string is exhausted.

PARSE returns the address ADDR of where it started looking and the COUNT of the characters until CHAR was found.

PARSE does not skip over leading occurrences of CHAR. It returns even if the first character checked is CHAR .

```
: PS" ( -- , like ." )
  ASCII " PARSE TYPE ;
20 5 - PS" Answer = " . CR
( print Answer = 15 )
```

Related Words: /STRING SOURCE SKIP SCAN WORD QUERY

PARSE-WORD (char -- addr count)

PARSE-WORD is just like PARSE except it will skip over skip over leading delimiters. IF the first character it finds equals char , it will continue checking until a non-delimiter is found. It will then again look for a delimiter, after which it will return.

Related Words: /STRING SOURCE

PICK (... v2 v1 v0 M -- ... v2 v1 v0 vM)

Create a copy of value number M, replacing M on the stack. Value number 0 is the value immediately below the PICK index.

!!! WARNING !!! - This '83 version of PICK differs from old standard versions of PICK. In old versions, the top value was value number 1 not 0 . Please see the MULTISTANDARDS facility for access to different versions of PICK . The following code shows the equivalent of doing a DUP.

```
OLD WAY      vs      NEW WAY
1 PICK              0 PICK
```

Standards: '79 '83

```
33 45 66 87 2 PICK . ( print 45 )
```

Related Words: RPICK XPICK ROT

PLACE (source-address count \$dest-addr --)

PLACE is used to move a string from one address to another. If the user-variable MAKEUCASE is non-zero, it will convert the string to upper-case after the move.

Instead of manipulating MAKEUCASE, the programmer may call LPLACE, which will not perform the conversion, allowing lower-case output.

Standards: JForth unique

```
CREATE SPAD 80 ALLOT \ make string holder
" Hello World" COUNT SPAD PLACE
SPAD COUNT TYPE \ print copied string
```

Related Words: \$MOVE MOVE COUNT

PREVIOUS (--)

Drops the vocabulary added to the context stack by ALSO. This provides a way to unnest changes to the compiler search order. See the section on Forth Vocabularies.

```
ALSO MUSIC \ search music vocab first
INCLUDE COMPOSITION
PREVIOUS \ now back the way it was
```

POP (memblock -- n)

POP is used on a memory block that had been allocated via ALLOCBLOCK, and returns the cell just under that being pointed to by FREEBYTE, adjusting the FREEBYTE counter.

POP, in conjunction with PUSH, allows memory of this type to be conveniently used as stacks. Using the contents of the FREEBYTE field of a memblock, these words keep track of the next free location.

Note that, for efficiency, no error checking is performed by POP. It is the responsibility of the calling program to insure data is present before trying to POP it.

Standards: JForth unique.

```
\ Close a stack of files.
MY-FILES @   DUP FREECELL 0  ( -- memblock #cells 0 )
DO   DUP POP ( -- memblock file-pointer ) FCLOSE
LOOP DROP
```

Related Words: PUSH ALLOCBLOCK FREEBYTE +STACK -STACK

POTGO?

```
POTGO_LIB
POTGO_NAME
```

These words are used to manage the POTGO Library. See :LIBRARY.

PRINTER.ON

Echo JForth output to both the screen and printer. Uses \$LOGTO to vector EMIT.

```
INCLUDE JU:LOGTO
PRINTER.ON  MAP  PRINTER.OFF  ( print MAP on printer )
```

If you want print a file on the printer, you can use the AmigaDOS copy command.

```
COPY filename PRT:
```

If you want to print a file with line numbers and proper page breaks, you can use the JForth Print Application. See the appendix on the Clonable Applications.

PULLTIB (--) "pull t i b"

PULLTIB restores the previous TIB environment that had been saved via PUSHTIB.

PULLTIB causes the TIB, #TIB, >IN and FBLK to be restored.

Standards: JForth unique

```
PUSHTIB  QUERY INTERPRET  PULLTIB
```

Related Words: PUSHTIB #TIB TIB

PUSH (n memblock --)

PUSH is used on a memory block that had been allocated via ALLOCBLOCK, and places N in the location being pointed to by the base address+FREEBYTE, adjusting FREEBYTE to the next available location.

PUSH, in conjunction with POP, allows memory of this type to be conveniently used as stacks.

Note that, for efficiency sake, no error checking is performed by PUSH. It is the responsibility of the calling program to insure that room exists in the allocated memory space before PUSHing to that area.

Standards: JForth unique.

```
( -- file-pointer ) MY-FILES @   DUP FREEBYTE OVER SIZEMEM <
IF   PUSH
ELSE ." FILE-STACK FULL!  NO ROOM FOR ANOTHER PUSH!" QUIT
THEN
```

Related Words: FREEBYTE FREECELL SIZEMEM POP ALLOCBLOCK

PUSHRELOC (rel-addr --) "push reloc"

This creates a relocation entry for an address that requires relocation by the Amiga Loader as JForth is being loaded in from mass storage.

Occasionally, in a large image, it is necessary to compile an absolute 32-bit address, referencing another part of JForth, within the body of a definition.

This potentially could pose a problem, as programs cannot specify their destination when being loaded, and any absolute references will be invalid.

JForth solves the problem by keeping a table of all addresses needing relocation. To add to this table, just pass to PUSHRELOC the JForth-relative address of the location that contains the 32-bit reference. Note that the location should be properly initialized with the correct 32-bit absolute address when PUSHRELOC is called.

Standards: Amiga unique

```
VARIABLE IMAGE-BASE 0 >ABS IMAGE-BASE ! IMAGE-BASE PUSHRELOC
```

PUSHTIB (--)

PUSHTIB saves information about the TIB and INTERPRET environment. Input can then be redirected to a new level, then each environment restored by a corresponding call to PULLTIB.

This technique is used in such nestable system words as INCLUDE and ?PAUSE.

PUSHTIB is nestable to a level of 16.

Standards: JForth unique

```
PUSHTIB QUERY INTERPRET PULLTIB
```

Related Words: PULLTIB #TIB

QUERY (--)

QUERY will repeatedly receive characters from the keyboard, placing them in the TIB until a carriage return is entered. Upon exiting, #TIB will contain the number of characters received, and >IN will be 0.

Note that calling QUERY while a valid INTERPRET stream exists in the TIB will result in the loss of the remainder of the stream, unless PUSHTIB and PULLTIB are utilized before and after the QUERY operation and subsequent user processing.

```
: GET-NUMBER ( -- d ) ." Input a number: "  
  QUERY BL WORD NUMBER ;
```

Related Words: EXPECT TIB >IN PUSHTIB PULLTIB #TIB

QUIT (-- , quit whatever and return to interpreter)

QUIT is a DEFERred word, whose default content in JForth is (QUIT). When a program has sensed an irrecoverable error, QUIT is the function used to terminate program flow, and restart the system from a known point. In a cloned application, QUIT will cause an exit from the program. For more information specific to QUIT in the development environment, see (QUIT).

```
0< IF ." Error in this word!" QUIT THEN
```

Related Words: INTERPRET ABORT CLONE