```
false 0 ; true -1 ;
off ( a -- ) true swap ! ; on ( a -- ) false swap ! ;
pad ( -- a ) temporary space Vhere @ 256 + ;
The number conversion routines A and B address registers.
hex 16 Base ! ; decimal 10 Base ! ; octal 8 Base ! ;
8 Infd 8 Outfd
type ( a n -- ) Outfd @ -rot ( fd a n ) syswrite drop ;
negate ( n -- -1*n ) not 1 + ;
abs ( n -- |n| ) test -if negate then ;
asciise ( u1 -- u2 ) converts a digit u1 to an ascii value
    10 - -if 1 + 57 ascii code of 9 + ; then
    97 ascii code of a + ;
hold ( c -- ) put c in the ascii string c!- ;
sign ( n -- ) test -if 45 ascii code of - hold ; then drop ;
<# start number conversion 48 ascii code of 0 pad dup A! dup B! c! ;
#> ( -- a n ) A 1 + B A - ;
# ( u1 -- u2 ) write a single digit in ascii at the address in A
    Base @ /mod asciise c!- ;
#s ( u1 -- ) convert u1 to an ascii string at the adress in A
    begin # until ;
. ( n -- ) A push B push dup abs <# #s sign #> type pop B! pop A! ;
OREAD 0 ; OWRITE 1 ; ORDWR 2 ; OTRUNC 16 ; DMREAD 4 ; DMWRITE 2 ;
close ( fd -- ) sysclose drop ;
exit Infd @ close Outfd @ close bye ;
8 Type 8 Len 264 Str
read ( a n -- ) Infd @ -rot ( fd a n ) sysread test if drop ; then
    eof exit ;
swrite ( counted_string -- ) Outfd @ swap count syswrite drop ;
space writes a space to outfd   swrite ;
newline writes a newline to outfd   swrite ;
tab writes a tab to outfd   swrite ;
none do nothing exit ;
>name read to Len and Str from Infd Len 1 read Str Len c@ read ;
name> ( counted_string -- ) writes the counted string, Len and Str to Outfd
    Len c@ . space Str Len c@ type space ;
name ( counted_string -- ) read the name and write the counted_string and name
    >name swrite name> ;
>base Base 1 read ; base> Base @ . space ;
num ( counted_string -- ) read the number and write the counted_string and number
    >base >name swrite base> name> decimal ;
number Number  num ; literalcompile Literalcompile  num ;
textcomment Textcomment  name ; execute Execute  name ;
compile Compile  name ; define Define  name ;
variable Variable  name ; address Address  name ;
macrocompile Macrocompile  name ; string String  name ;
perm DMREAD DMWRITE or ; perms perm dup 3 shl over 6 shl or or ;
opens 1 arg OREAD sysopen Infd ! 2 arg OWRITE perms syscreate Outfd ! ;
type Type 1 read Type c@ ;
interpret jump
    none ; textcomment ; execute ; number ; compile ; literalcompile ;
```

```
    macrocompile ; address ; string ; tab ; newline ;
    define ; variable ; none ;
loop type interpret loop ;
main opens loop ;
```