Forth-79	Forth-83	dpANS	FIG	eForth	RETRO
!	!	!	0	!	DUP
#	#	#	1	!IO	DROP
#>	#>	#>	2	0<	SWAP
#S	#S	#S	3	>R	CALL
'	#TIB	'	!	?BRANCH	EQ?
(	'	(	!CSP	?RX	-EQ?
*	(	*	#	@	LT?
*/	*	*/	#>	AND	GT?
*/MOD	*/	*/MOD	#S	BRANCH	FETCH
+	*/MOD	+	'	BYE	STORE
+!	+	+!	(	C!	+
+LOOP	+!	+LOOP	(+LOOP)	C@	-
+LOOP	+LOOP	,	(.")	doLIST	*
,	,	-	(;CODE)	doLIT	/MOD
-	-	.	(ABORT)	DROP	AND
-TRAILING	-TRAILING	."	(DO)	DUP	OR
.	.	/	(FIND)	EXECUTE	XOR
."	."	/MOD	(LINE)	EXIT	SHIFT
/	.(	0<	(LOOP)	NEXT	FETCH-NEXT
/MOD	/	0=	(NUMBER)	OR	STORE-NEXT
0<	/MOD	1+	*	OVER	PUSH
0=	0<	1-	*/	R>	POP
0>	0=	2!	*/MOD	R@	0;
1+	0>	2*	+	RP!	S:TO-NUMBER
1-	1+	2/	+!	RP@	S:EQ?
2+	1-	2@	+-	SP!	S:LENGTH
2-	2+	2DROP	+BUF	SP@	CHOOSE
79-STANDARD	2-	2DUP	+LOOP	SWAP	IF
:	2/	2OVER	+ORIGIN	TX!	-IF
:	:	2SWAP	,	UM+	COMPILER
;	;	:	-	XOR	HEAP
;	<	;	-->		,
<	<#	<	-DUP		S,
<#	>	<#	-FIND		;
>	>BODY	>	-LINE		[
>IN	>IN	>BODY	-TRAILING		]
>R	>R	>IN	.		DICTIONARY
?	?DUP	>NUMBER	."		D:LINK
?DUP	@	>R	.R		D:XT
@	[	?DUP	/		D:CLASS
[	[']	@	/MOD		D:NAME
[COMPILE]	[COMPILE]	[	0<		CLASS:WORD
]	MOD	[']	0=		CLASS:PRIMITIVE
ABORT	ABORT	[CHAR]	0BRANCH		CLASS:MACRO
ABS	ABORT"	]	1+		CLASS:DATA
ALLOT	ABS	ABORT	2+		D:ADD-HEADER
AND	ALLOT	ABORT"	;		PREFIX:#
BASE	BASE	ABS	;CODE		PREFIX::
BEGIN	BEGIN	ACCEPT	;S		PREFIX:&
BEGIN	BLK	ALIGN	<		PREFIX:$
BLK	BLOCK	ALIGNED	<#		REPEAT
BLOCK	BUFFER	ALLOT	<BUILDS		AGAIN
BUFFER	C!	AND	>		INTERPRET
C!	C@	BASE	>R		D:LOOKUP
C@	CMOVE	BEGIN	?		ERR:NOTFOUND
CMOVE	CMOVE>	BL	?COMP		
COLON	COMPILE	C!	?CSP		
COMPILE	CONSTANT	C,	?EXEC		
CONSTANT	CONVERT	C@	?LOADING		
CONSTANT	COUNT	CELL+	?PAIRS		
CONTEXT	CR	CELLS	?STACK		
CONVERT	CREATE	CHAR	?TERMINAL		
COUNT	D+	CHAR+	@		
CR	D<	CHARS	[		
CREATE	DECIMAL	CONSTANT	[COMPILE]		
CREATE	DEFINITIONS	COUNT	]		
CURRENT	DEPTH	CR	ABORT		
D+	DNEGATE	CREATE	ABS		

D<	DO	DECIMAL	AGAIN		
DECIMAL	DOES>	DEPTH	ALLOT		
DEFINITIONS	DROP	DO	AND		
DEPTH	DUP	DOES>	B/BUF		
DNEGATE	ELSE	DROP	B/SCR		
DO	EMIT	DUP	BACK		
DO	EXECUTE	ELSE	BASE		
DOES>	EXIT	EMIT	BEGIN		
DOES>	EXPECT	ENVIRONMENT?	BL		
DROP	FILL	EVALUATE	BLANKS		
DUP	FIND	EXECUTE	BLK		
ELSE	FLUSH	EXIT	BLOCK		
ELSE	FORGET	FILL	BLOCK-READ		
EMIT	FORTH	FIND	BLOCK-WRITE		
EMPTY-BUFFERS	FORTH-83	FM/MOD	BRANCH		
EXECUTE	HERE	HERE	BUFFER		
EXIT	HOLD	HOLD	C!		
EXPECT	I	I	C,		
FILL	IF	IF	C@		
FIND	IMMEDIATE	IMMEDIATE	CFA		
FORGET	J	INVERT	CMOVE		
FORTH	KEY	J	COLD		
HERE	LEAVE	KEY	COMPILE		
HOLD	LITERAL	LEAVE	CONSTANT		
I	LOAD	LITERAL	CONTEXT		
IF	LOOP	LOOP	COUNT		
IF	MAX	LSHIFT	CR		
IMMEDIATE	MIN	M*	CREATE		
J	MOD	MAX	CSP		
KEY	NEGATE	MIN	D+		
LEAVE	NOT	MOD	D+-		
LIST	OR	MOVE	D.		
LITERAL	OVER	NEGATE	D.R		
LITERAL	PAD	OR	DABS		
LOAD	PICK	OVER	DECIMAL		
LOOP	QUIT	POSTPONE	DEFINITIONS		
LOOP	R>	QUIT	DIGIT		
MAX	R@	R>	DLIST		
MIN	REPEAT	R@	DLITERAL		
MOD	ROLL	RECURSE	DMINUS		
MOVE	ROT	REPEAT	DO		
NEGATE	SAVE-BUFFERS	ROT	DO		
NOT	SIGN	RSHIFT	DOES>		
OR	SPACE	S"	DP		
OVER	SPACES	S>D	DPL		
PAD	SPAN	SIGN	DR0		
PICK	STATE	SM/REM	DR1		
QUERY	SWAP	SOURCE	DROP		
QUIT	THEN	SPACE	DUMP		
R>	TIB	SPACES	DUP		
R@	TYPE	STATE	ELSE		
REPEAT	U.	SWAP	EMIT		
REPEAT	U<	THEN	EMPTY-BUFFERS		
ROLL	UM*	TYPE	ENCLOSE		
ROT	UM/MOD	U.	END		
SAVE-BUFFERS	UNTIL	U<	ENDIF		
SCR	UPDATE	UM*	ERASE		
SEMICOLON	VARIABLE	UM/MOD	ERROR		
SIGN	VOCABULARY	UNLOOP	ERROR		
SPACE	WHILE	UNTIL	EXECUTE		
SPACES	WORD	VARIABLE	EXPECT		
STATE	XOR	WHILE	FENCE		
SWAP	=	WORD	FILL		
THEN		XOR	FIRST		
THEN		=	FLD		
TYPE			FORGET		
U*			FORTH		
U.			HERE		
U/			HEX		

UNTIL
UNTIL
UPDATE
VARIABLE
VARIABLE
VOCABULARY
WHILE
WHILE
WORD
XOR
 =

HLD
HOLD
I
ID.
IF
IMMEDIATE
IN
INDEX
INTERPRET
KEY
LATEST
LEAVE
LFA
LIMIT
LIST
LIT
LITERAL
LOAD
LOOP
M*
M/
M/MOD
MAX
MESSAGE
MIN
MINUS
MOD
MON
MOVE
NEXT
NFA
NUMBER
OFFSET
OR
OUT
OVER
PAD
PFA
POP
POP
PREV
PUSH
QUERY
QUIT
R
R#
R/W
R>
R@
REPEAT
ROT
RP!
S0
SCR
SIGN
SMUDGE
SP!
SP@
SPACE
SPACES
STATE
SWAP
S→D
TASK
THEN
TIB
TOGGLE
TRAVERSE
TRIAD

			TYPE U* U/ UNTIL UPDATE USE USER VARIABLE VLIST VOC-LINK VOCABULARY WARNING WHILE WIDTH WORD X XOR =		