i) A + ((B − C * D) / E) + F − G / H

Postfix:

INPUT	STACK	POSTFIX
A	Empty	A
+	+	A
(	+(	A
(	+((	A
B	+((	A B
-	+((-	A B
C	+((-	A B C
*	+((-*	A B C
D	+((-*	A B C D
)	+(	A B C D * -
/	+(/	A B C D * -
E	+(/	A B C D * - E
)	+	A B C D * - E /
+	+	A B C D * - E / +
F	+	A B C D * - E / + F
-	-	A B C D * - E / + F+
G	-	A B C D * - E / + F+G
/	-/	A B C D * - E / + F+G
H	-/	A B C D * - E / + F + G H
NONE	EMPTY	A B C D * - E / + F + G H/-

Evaluation: A B C D * - E / + F + G H / -

For; A=1 B=2 C=3 D=4 E=5 F=6 G=7 H=1

INPUT	OPERATION	STACK	RESULT
1	Push	1	
2	Push	1 2	
3	Push	1 2 3	
4	Push	1 2 3 4	
*	Evaluate	1 2	12
->	Push	1 2 12	
-	Evaluate	1	-10
->	Push	1 -10	
5	Push	1 -10 5	
/	Evaluate	1	-2
->	Push	1 -2	
+	Evaluate		-1
	Push	-1	
6	Push	-1 6	
+	Evaluate		5
->	Push	5	
7	Push	5 7	
1	Push	5 7 1	
/	Evaluate	5	7
->	Push	5 7	
-	Evaluate		-2

H / G − F +) E /) D * C − B ((+ A

Prefix:

INPUT	STACK	POSTFIX
H	Empty	H
/	/	H
G	/	H G
-	-	H G /
F	-	H G / F
+	+	H G / F -
)	+)	H G / F -
E	+)	H G / F - E
/	+)/	H G / F − E
)	+)/)	H G / F − E
D	+)/)	H G / F − E D
*	+)/)*	H G / F − E D
C	+)/)*	H G / F − E D C
-	+)/)-	H G / F − E D C *
B	+)/)-	H G / F − E D C * B
(	+)/	H G / F − E D C * B -
(	+	H G / F − E D C * B -/
+	+	H G / F − E D C * B -/+
A	+	H G / F − E D C * B - / + A
NONE	EMPTY	H G / F − E D C * B - / + A+

Evaluation: H G / F – E D C * B - / + A+

For; A=1 B=2 C=3 D=4 E=5 F=6 G=7 H=1

INPUT	OPERATION	STACK	RESULT
1	Push	1	
7	Push	1 7	
/	Evaluate		1/7
	Push	1/7	
6	Push	1/7 6	
-	Evaluate		-41/7
	Push	-41/7	
5	Push	-41/7 5	
4	Push	-41/7 5 4	
3	Push	-41/7 5 4 3	
*	Evaluate	-41/7 5	12
	Push	-41/7 5 12	
2	Push	-41/7 5 12 2	
-	Evaluate	-41/7 5	10
	Push	-41/7 5 10	
/	Evaluate	-41/7	½
	Push	-41/7 1/2	
+	Evaluate		-65/14
	Push	-65/14	
1	Push	-65/14 1	
+	Evaluate		-51/14?

ii) ! (A && ! ((B < C) || (C > D))) || (C < E)

Postfix:

INPUT	STACK	POSTFIX
!	!	
(	!(	
A	!(	A
&&	!(&&	A
!	!(&&!	A
(	!(&&!(	A
(	!(&&!((	A
B	!(&&!((	A B
<	!(&&!((<	A B
C	!(&&!((<	A B C
)	!(&&!(	A B C <
||	!(&&!(||	A B C <
(	!(&&!(||(	A B C <
C	!(&&!(||(	A B C < C
>	!(&&!(||(>	A B C < C
D	!(&&!(||(>	A B C < C D
)	!(&&!(||	A B C < C D >
)	!(&&!	A B C < C D >||
)	!	A B C < C D > || ! &&
||	||	A B C < C D > || ! && !
(	|| (	A B C < C D > || ! && !
C	|| (	A B C < C D > || ! && ! C
<	|| (<	A B C < C D > || ! && ! C
E	|| (<	A B C < C D > || ! && ! C E
)	||	A B C < C D > || ! && ! C E <
NONE	EMPTY	A B C < C D > || ! && ! C E < ||

Evaluation: A B C < C D > || ! && ! C E < ||

For: A=1 B=2 C=3 D=4 E=5

INPUT	OPERATION	STACK	RESULT
A	Push	1	
B	Push	1 2	
C	Push	1 2 3	
<	Evaluate	1	2<3=true
	Push	1 true	
C	Push	1 true 3	
D	Push	1 true 3 4	
>	Evaluate	1 true	3>4 =false
	Push	1 true false	
||	Evaluate	1	true||false = true
	Push	1 true	12
!	Evaluate	1	!true=false
	Push	1 false	
&&	Evaluate		1&&false=false
	Push	false	
!	Evaluate		!false=true
	Push	true	
C	Push	True 3	
E	Push	True 3 5	
<	Evaluate	True	3<5=true
	Push	True true	
||	Evaluate		True||true = true

) E < C (| |))) D > C (| |) C < B ((! && A (!

Prefix:

INPUT	STACK	POSTFIX
)	)	
E	)	E
<	)<	E
C	)<	E C
(		E C <
||	||	E C <
)	||)	E C <
)	||))	E C <
)	||)))	E C <
D	||)))	E C < D
>	||))) >	E C < D
C	||))) >	E C < D C
(	||))	E C < D C >
||	||)) ||	E C < D C >
)	||)) ||)	E C < D C >
C	||)) ||)	E C < D C > C
<	||)) ||) <	E C < D C > C
B	||)) ||) <	E C < D C > C B
(	||)) ||	E C < D C > C B <
(	||)	E C < D C > C B < ||
!	||) !	E C < D C > C B < ||
&&	||) &&	E C < D C > C B < ||!
A	||) &&	E C < D C > C B < ||! A
(	||	E C < D C > C B < ||! A &&
!	|| !	E C < D C > C B < ||! A &&
NULL	EMPTY	E C < D C > C B < || !A && ! ||

Evaluation: E C < D C > C B < || !A && ! ||

For: A=1 B=2 C=3 D=4 E=5

INPUT	OPERATION	STACK	RESULT
E	Push	5	
C	Push	5 3	
<	Evaluate		5<3 =false
	Push	false	
D	Push	False 4	
C	Push	False 4 3	
>	Evaluate	False	4>3=true
	Push	False true	
C	Push	False true 3	
B	Push	False true 3 2	
<	Evaluate	False true	3<2=False
	Push	False true false	
||	Evaluate	False	True||false =true
	Evaluate	False true	
!	Evaluate	False	!true=false
	Push	False false	
A	Push	False false 1	
&&	Evaluate	False	False&& 1 ==false
	Push	False false	
!	Evaluate	False	!false=true
	Push	False true	
||	Evaluate		false||true = true