

Left parenthesis – do nothing

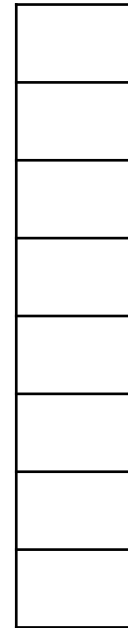


((A + B) * (C - D)) ;



Left parenthesis – do nothing

((A + B) * (C - D)) ;



Add A to stack

((A + B) * (C - D)) ;



Add B to stack

((A + B) * (C - D)) ;



Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = B

Right parenthesis – pop 3 things off
stack

((A + B) * (C - D)) ;



RHS = B
OP = +

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = B
OP = +
LHS = A

Right parenthesis – Combine and push
onto stack

((A + B) * (C - D)) ;



RHS = B

OP = +

LHS = A

EXPR = OP + LHS + RHS

Right parenthesis – Combine and push onto stack

((A + B) * (C - D)) ;



RHS = B

OP = +

LHS = A

EXPR = OP + LHS + RHS

EXPR = +AB

Add C to Stack

((A + B) * (C - D)) ;



Add - to Stack

((A + B) * (C - D)) ;



-
C
*
+AB

Add D to Stack

((A + B) * (C - D)) ;



D
-
C
*
+AB

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



D
-
C
*
+AB

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = D

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = D
OP = -

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = D
OP = -
LHS = C

Right parenthesis – pop 3 things off stack

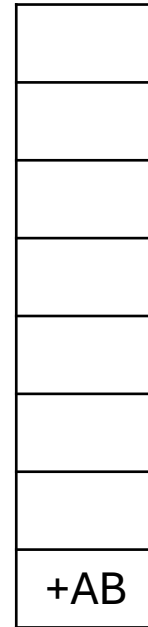
((A + B) * (C - D)) ;



RHS = -CD

Right parenthesis – pop 3 things off stack

((A + B) * (C - D)) ;



RHS = -CD
OP = *

Right parenthesis – pop 3 things off stack

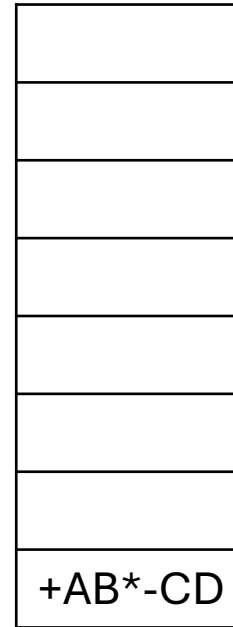
((A + B) * (C - D)) ;



RHS = -CD
OP = *
LHS = +AB

Right parenthesis – Combine and push onto stack

((A + B) * (C - D)) ;



RHS = -CD

OP = *

LHS = +AB

EXPR = OP + LHS + RHS

EXPR = +AB*-CD

