

Left parenthesis – do nothing



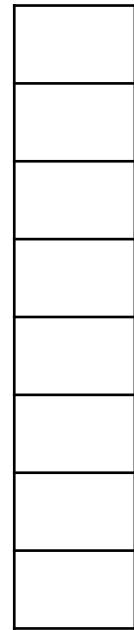
```
( ( A + B ) * ( C - D ) ) ;
```



Left parenthesis – do nothing



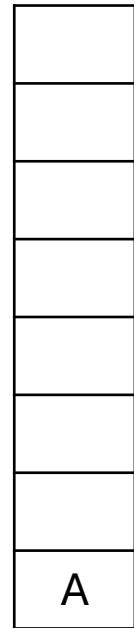
( ( A + B ) \* ( C - D ) ) ;



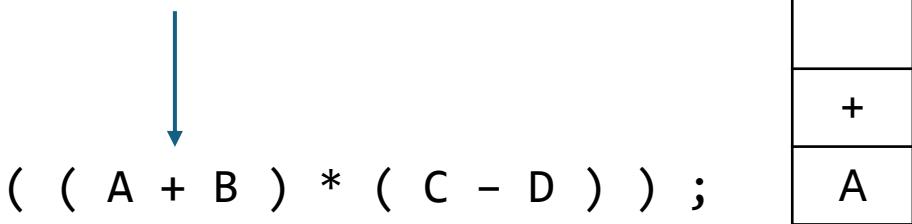
Add A to stack



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



## Add + to stack



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 ) ) ;



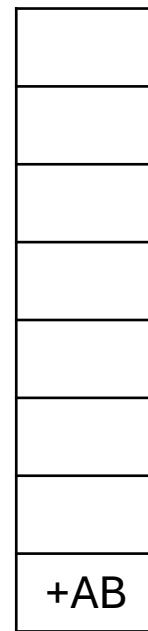
RHS = B  
OP = +  
LHS = A

EXPR = OP + LHS + RHS

Right parenthesis – Combine and push  
onto stack



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



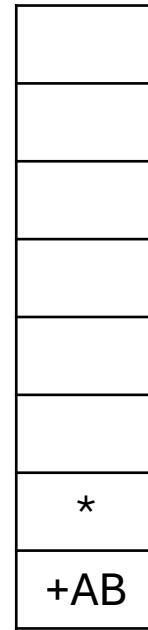
RHS = B  
OP = +  
LHS = A

EXPR = OP + LHS + RHS  
EXPR = +AB

Add \* to stack

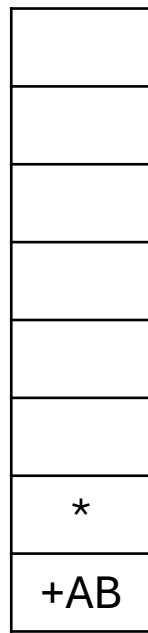


( ( A + B ) \* ( C - D ) ) ;



## Left parenthesis – do nothing

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



Add C to Stack

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



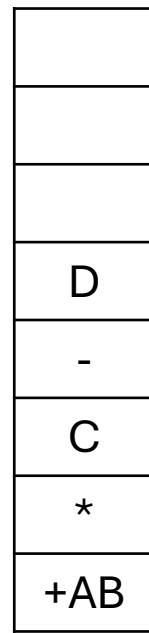
Add - to Stack

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



Add D to Stack

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



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 ) ) ;

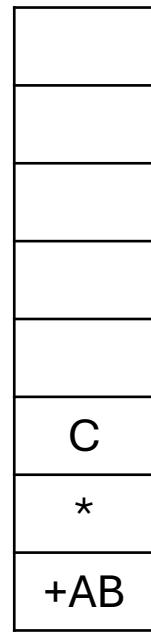


-
C
*
+AB

RHS = D

Right parenthesis – pop 3 things off stack

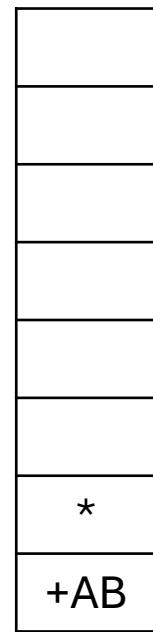
( ( A + B ) \* ( C - D ) ) ;



RHS = D  
OP = -

Right parenthesis – pop 3 things off stack

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



RHS = D  
OP = -  
LHS = C

Right parenthesis – Combine and push  
onto stack

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



-CD
*
+AB

RHS = D  
OP = -  
LHS = C

EXPR = OP + LHS + RHS  
EXPR = -CD

Right parenthesis – pop 3 things off stack

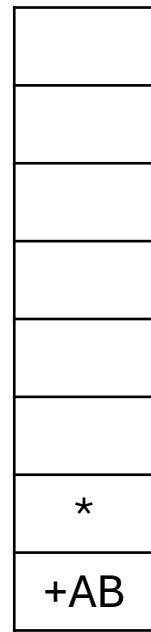
( ( A + B ) \* ( C - D ) ) ;



-CD
*
+AB

Right parenthesis – pop 3 things off stack

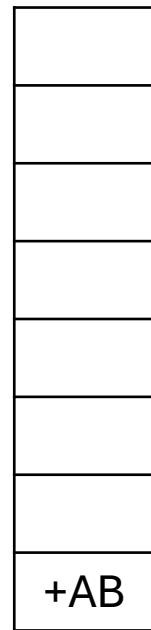
( ( A + B ) \* ( C - D ) ) ;



RHS = -CD

Right parenthesis – pop 3 things off stack

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



RHS = -CD  
OP = \*

Right parenthesis – pop 3 things off stack

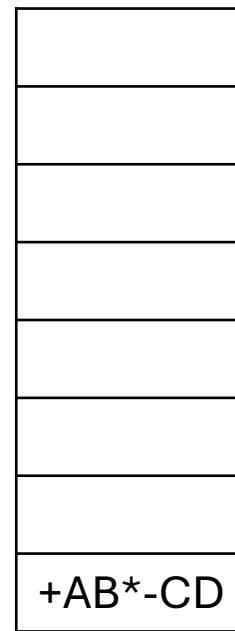
( ( A + B ) \* ( C - D ) ) ;



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

End of expression – Stack has prefix

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

