value	operator
stack	stack

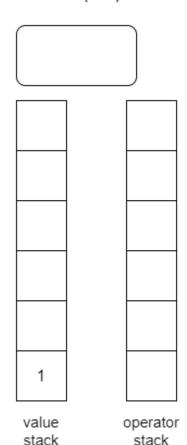
 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a 'f'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 1 Copyright Arman Ashkari

1	
value stack	operator stack

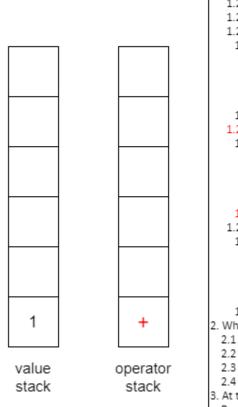
 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 2 Copyright Arman Ashkari



 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 3 Copyright Arman Ashkari



 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 4 Copyright Arman Ashkari

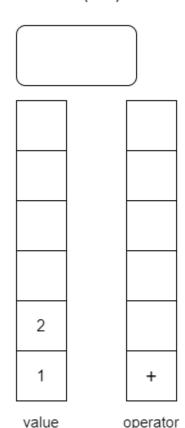
2	
1	+
value	perator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 5 Copyright Arman Ashkari



stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 6 Copyright Arman Ashkari

2	*
1	+
value	perator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Copyright Arman Ashkari Page 7

	1	
6		
2		*
1		+

operator

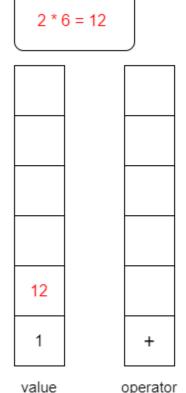
stack

value

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 8 Copyright Arman Ashkari

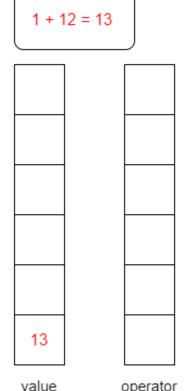


stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 9 Copyright Arman Ashkari



stack

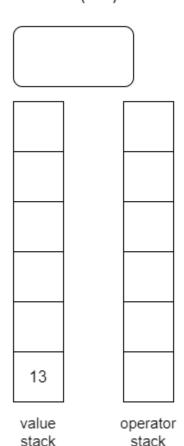
operator

stack

```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
   1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
     1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
     1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
     1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
     1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
  2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
```

Page 10 Copyright Arman Ashkari



 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 11 Copyright Arman Ashkari

13		•
value	. (perator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 12 Copyright Arman Ashkari

2)	
1	3	-

operator

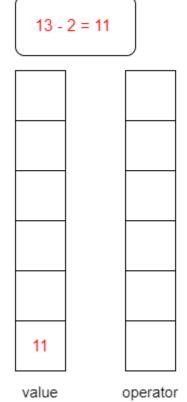
stack

value

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 13 Copyright Arman Ashkari

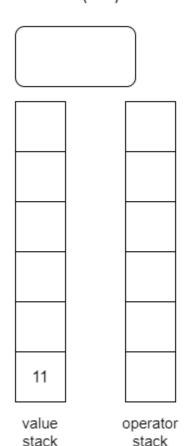


stack

stack

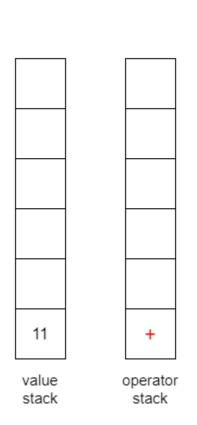
 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 14 Copyright Arman Ashkari



 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

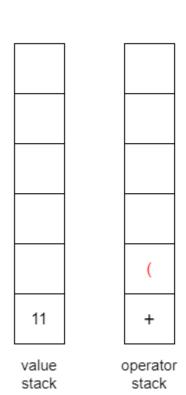
Page 15 Copyright Arman Ashkari



```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
   1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
     1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
     1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
     1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
     1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
     1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
  2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
  Pop it and print the value.
```

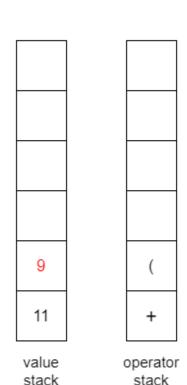
Page 16 Copyright Arman Ashkari



```
    While there are still characters left in the string,

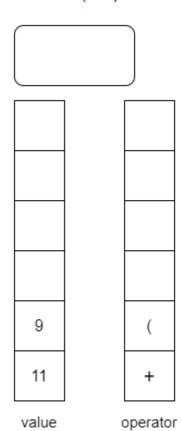
 1.1 Get the next character.
 1.2 If the character is:
   1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
     1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
     1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
     1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
     1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
     1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
  2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
  Pop it and print the value.
```

Page 17 Copyright Arman Ashkari



 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 18 Copyright Arman Ashkari



stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 19 Copyright Arman Ashkari

	-
9	(
11	+
value	perator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 20 Copyright Arman Ashkari

6	-
9	(
11	+
value	perator

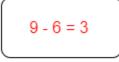
stack

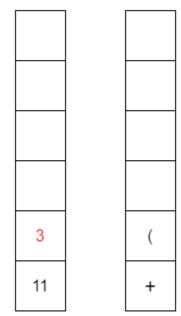
stack

```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
  1.2.1 A digit '0' to '9': push it onto the value stack.
  1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
    1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
    1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
    1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
    1.2.5.2 Push the '*' or '/' onto the operator stack.
While the operator stack is not empty,
 2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
  Pop it and print the value.
```

Page 21 Copyright Arman Ashkari





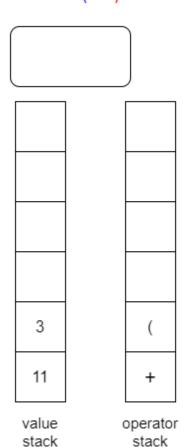
value operator stack stack

```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
   1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis "(",
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
     1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
     1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
     1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
     1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
  2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
```

3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 22 Copyright Arman Ashkari



```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
   1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis "(",
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
     1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
     1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
     1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
     1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
  2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
```

Page 23 Copyright Arman Ashkari

3	
11	+

operator

stack

value

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result. Pop it and print the value.

Page 24 Copyright Arman Ashkari

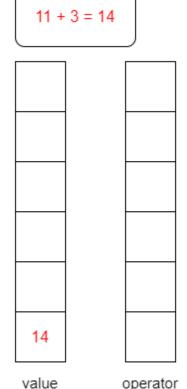
3		
11		+
value	(perator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 25 Copyright Arman Ashkari



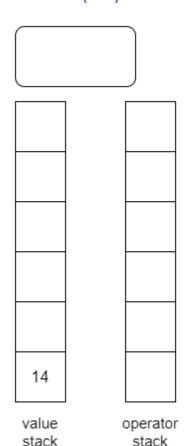
stack

stack

```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
  1.2.1 A digit '0' to '9': push it onto the value stack.
   1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
    1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
    1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
    1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
    1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
 2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
```

Page 26 Copyright Arman Ashkari



```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
  1.2.1 A digit '0' to '9': push it onto the value stack.
  1.2.2 A left parenthesis '(': push it onto the operator stack.
   1.2.3 A right parenthesis ')':
    1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',
      1.2.3.1.1 Pop the operator from the operator stack.
      1.2.3.1.2 Pop the value stack twice, getting two operands.
      1.2.3.1.3 Apply the operator to the operands, in the correct order.
      1.2.3.1.4 Push the result onto the value stack.
    1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
   1.2.4 A '+' operator or a '-' operator:
    1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
      1.2.4.1.1 Pop the operator from the operator stack.
      1.2.4.1.2 Pop the value stack twice, getting two operands.
      1.2.4.1.3 Apply the operator to the operands, in the correct order.
      1.2.4.1.4 Push the result onto the value stack.
    1.2.4.2 Push the '+' or '-' onto the operator stack.
   1.2.5 A '*' operator or a '/' operator:
    1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"),
      1.2.5.1.1 Pop the operator from the operator stack.
      1.2.5.1.2 Pop the value stack twice, getting two operands.
      1.2.5.1.3 Apply the operator to the operands, in the correct order.
      1.2.5.1.4 Push the result onto the value stack.
    1.2.5.2 Push the '*' or '/' onto the operator stack.
2. While the operator stack is not empty,
 2.1 Pop the operator from the operator stack.
  2.2 Pop the value stack twice, getting two operands.
  2.3 Apply the operator to the operands, in the correct order.
  2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
```

Page 27 Copyright Arman Ashkari

14	
value	operator

stack

stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order. 2.4 Push the result onto the value stack. 3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 28 Copyright Arman Ashkari

stack

					1.2.2
				1.2.3	
					1.2
					1
					1
					1
					1
					1.2
					1.2.4
			\vdash		1.2
					1
					1
					1
					1
					1.2
					1.2.5
					1.2
					1
					1
					1
					1
	14				1.2
	1 -				2. While 2.1 Po
		l			2.1 Po
	value operator			2.2 PC	
	value	value operator			2.5 AL

stack

```
    While there are still characters left in the string,

 1.1 Get the next character.
 1.2 If the character is:
  1.2.1 A digit '0' to '9': push it onto the value stack.
       2 A left parenthesis '(': push it onto the operator stack.
       .3 A right parenthesis ')':

 2.3.1 While the thing on top of the operator stack is not a left parenthesis '(',

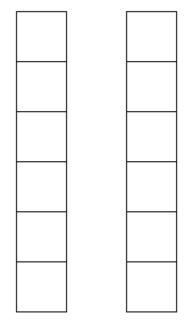
       1.2.3.1.1 Pop the operator from the operator stack.
       1.2.3.1.2 Pop the value stack twice, getting two operands.
       1.2.3.1.3 Apply the operator to the operands, in the correct order.
       1.2.3.1.4 Push the result onto the value stack.
       2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it.
       4 A '+' operator or a '-' operator:
       2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(',
       1.2.4.1.1 Pop the operator from the operator stack.
       1.2.4.1.2 Pop the value stack twice, getting two operands.
       1.2.4.1.3 Apply the operator to the operands, in the correct order.
       1.2.4.1.4 Push the result onto the value stack.
       2.4.2 Push the '+' or '-' onto the operator stack.
       .5 A '*' operator or a '/' operator:

    While the operator stack is not empty, and (the top thing on the operator stack is a '*' or a '/'),

       1.2.5.1.1 Pop the operator from the operator stack.
       1.2.5.1.2 Pop the value stack twice, getting two operands.
       1.2.5.1.3 Apply the operator to the operands, in the correct order.
       1.2.5.1.4 Push the result onto the value stack.
       2.5.2 Push the '*' or '/' onto the operator stack.
       e the operator stack is not empty,
       op the operator from the operator stack.
       op the value stack twice, getting two operands.
       pply the operator to the operands, in the correct order.
 2.4 Push the result onto the value stack.
3. At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.
 Pop it and print the value.
```

Page 29 Copyright Arman Ashkari

Ans = 14



value operator stack stack

 While there are still characters left in the string, 1.1 Get the next character. 1.2 If the character is: 1.2.1 A digit '0' to '9': push it onto the value stack. 1.2.2 A left parenthesis '(': push it onto the operator stack. 1.2.3 A right parenthesis ')': 1.2.3.1 While the thing on top of the operator stack is not a left parenthesis '(', 1.2.3.1.1 Pop the operator from the operator stack. 1.2.3.1.2 Pop the value stack twice, getting two operands. 1.2.3.1.3 Apply the operator to the operands, in the correct order. 1.2.3.1.4 Push the result onto the value stack. 1.2.3.2 Pop the left parenthesis '(' from the operator stack, and discard it. 1.2.4 A '+' operator or a '-' operator: 1.2.4.1 While the operator stack is not empty, and the top thing on the operator stack is not a left parenthesis '(', 1.2.4.1.1 Pop the operator from the operator stack. 1.2.4.1.2 Pop the value stack twice, getting two operands. 1.2.4.1.3 Apply the operator to the operands, in the correct order. 1.2.4.1.4 Push the result onto the value stack. 1.2.4.2 Push the '+' or '-' onto the operator stack. 1.2.5 A '*' operator or a '/' operator: 1.2.5.1 While the operator stack is not empty, and (the top thing on the operator stack is a "+" or a 'f"), 1.2.5.1.1 Pop the operator from the operator stack. 1.2.5.1.2 Pop the value stack twice, getting two operands. 1.2.5.1.3 Apply the operator to the operands, in the correct order. 1.2.5.1.4 Push the result onto the value stack. 1.2.5.2 Push the '*' or '/' onto the operator stack. 2. While the operator stack is not empty, 2.1 Pop the operator from the operator stack. 2.2 Pop the value stack twice, getting two operands. 2.3 Apply the operator to the operands, in the correct order.

At this point the operator stack should be empty, and the value stack should have only one value in it, which is the final result.

Page 30 Copyright Arman Ashkari

2.4 Push the result onto the value stack.