

## Project 5: Evaluate Expression

Define `int calculate(string s)`, which takes an expression, and return its evaluation.

- (1) Operands are integers.
- (2) Operators are `+`, `-`, `*`, and `/`.
- (3) (Optional) Add `(` and `)` in expression.
- (4) Use only integer operations, for example, `8 / 3` returns 2.
- (5) There might be spaces before or after an operator or operand. Some examples of expressions are as follows.

"2 \* (3 + 5 \* 6) ",  
"1 - (2 - (3 + 5 \* 6) + 6) ",  
"1 + (2 \* (3 + 5 \* 6) - 2) ",  
"1 + (2 \* (3 + 5 \* 6) - 2) ",  
"(1+(4+5+2)-3)+(6+8)",

Test examples:

3+2-6 is -1  
3+2-6+3+3+62 is 67  
3 + 2 - 6 + 3 + 3 + 2 is 7  
8 \* 2 / 3 is 5  
8 / 3 \* 3 is 6  
2 + 3 \* 5 is 17  
2 + 3 \* 5 + 8 / 3 - 6 + 2 \* 5 is 23  
2 + 3 \* 5 + 8 / 3 - 6 + 2 \* 5 / 6 - 5 is 9  
-2 \* -2 is 4  
-2 - -2 is 0  
-2 + -2 is -4  
-12 + 11 is -1  
1\*2-3/4+5\*6 is 32  
1\*2-3/4+5\*6-7\*8+9/10 is -24  
(3 + 5 ) is 8  
2 \* (3 + 5 ) is 16  
(3 + 5 ) \* -2 is -16  
2 \* (3 + 5 \* 6) is 66  
1 - (2 - (3 + 5 \* 6) + 6) is 26  
1 + (2 \* (3 + 5 \* 6) - 2) is 65  
1 + (2 \* (3 + 5 \* 6) - 2) is 65  
(1+(4+5+2)- 3)+(6+8) is 23  
(1+(4+5+2)-3)+(6+8) is 23  
((2\*(6-1))/2)\*4 is 20  
5 \* 4 + 3 / 2 is 21  
5 \* (4 + 3) / 2 is 17

$5 * (4 + 3) / 2$  is 15  
 $5 * (4 + 3 / 2)$  is 25  
 $(1 + (4 + 5 + 2) - 3 * -2) + (6 + 8)$  is 32  
 $-(3 + (4 + 5))$  is -12

Hints:

- (1) Process spaces to separate operands and operators.

Warning: the following approach do NOT work, here are some reasons:

```

string word = "";
for (auto x: str) //use C++ 11 or later
{
    if (x != ' ')
        word = word + x;
    else continue; //read the next token
    ...
}
  
```

- (a) There might not be spaces in an expression like "3+5-2".  
 (b) Furthermore, even there are spaces around each operator and operand, while the operand has more than one int, the code will not work due to if-statement in
- ```

if (x != ' ')
    ...
  
```

Change the above if-statement to while loop will not help since there is no way to update x in an expanded for-loop like for (auto x: str).

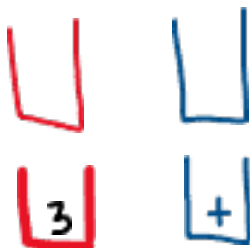
You can use

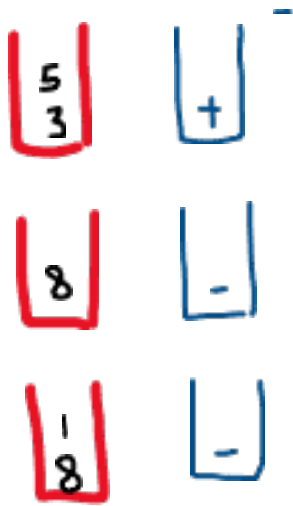
```

int i = 0;
int size = s.size(); //s is expression
while ( i < size )
    ...
  
```

- (2) We can work on this project one-step a time. What if we only work with + and -, that is, work on expression like  $3 + 5 - 1$  or  $9 / 2 * 3$ .  
 You can use `stoi("3")` to convert string "3" to int 3.

$3 + 5 - 1$





Pop up 1 (second operand) and 8 (first operand), do – operation. Get 7.

- (3) Then we expand our code to work on expression to incorporate \* and /. That is, work on expression like  $3 + 4 * 5$ .





Pop 5 (second operand) and 4 (first operand), do \* operation. Push to stack of operand.



Pop 20 (second operand) and 3 (first operand), do + operation. Get 23.

- (4) (optional) Finally, we will work on expressions with parentheses ( and ).
- (5) You may use stack, one for operands, one for operator. Depending on precedence and order of operators, learn when to push and pop.

### Online tester

There is online tester for part of our code.

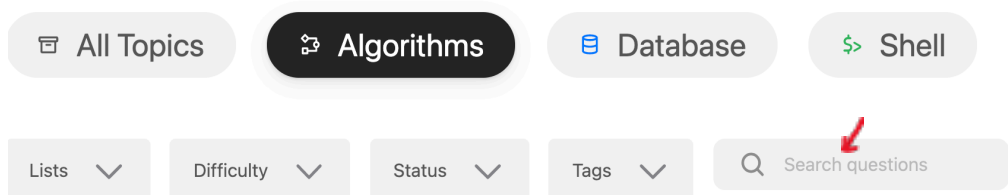
- (1) Calculator will test expression with +, -, (, and ).
- (2) Caculator II will test expression with +, -, \*, and /.
- (3) If our code works, it must pass both tests.

1. Register an account for leetcode.com if you have not done so.  
Choose <https://leetcode.com/problemset/algorithms/>, or go to Steps 2-3.

2. Click Problems.



3. Choose Algorithms.



4. In textbox Search questions, enter calculator.
5. Choose the problem.
6. In the pop-up window, choose language.

```
i C++ Autocomplete
1 class Solution {
2 public:
3     int calculate(string s)
4     {
5
6     }
7 };
```

7. Paste your code to replace the place-holder of function. Note that your function must be named as calculate, take a string, then return an int. Then click submit button in the bottom right to test your code.
8. If success, you will see in the left pane some data as follows.

Success Details >

Runtime: 32 ms, faster than 21.28% of C++ online submissions for Basic Calculator.

Memory Usage: 9.2 MB, less than 27.54% of C++ online submissions for Basic Calculator.

Next challenges:

Different Ways to Add Parentheses

Expression Add Operators

Basic Calculator III

Show off your acceptance:



| Time Submitted   | Status   | Runtime | Memory | Language |
|------------------|----------|---------|--------|----------|
| 07/06/2021 08:27 | Accepted | 32 ms   | 9.2 MB | cpp      |

Basic calculator:

Implement a basic calculator to evaluate a simple expression string.

The expression string may contain open ( and closing parentheses ), the plus + or minus sign -, non-negative integers and empty spaces.

You may assume that the given expression is always valid.

Some examples:

"1 + 1" = 2

" 2-1 + 2 " = 3

"(1+(4+5+2)-3)+(6+8)" = 23

Basic calculator II:

Implement a basic calculator to evaluate a simple expression string.

The expression string contains only non-negative integers, +, -, \*, / operators and empty spaces. The integer division should truncate toward zero.

You may assume that the given expression is always valid.

Some examples:

"3+2\*2" = 7

" 3/2 " = 1

" 3+5 / 2 " = 5

<https://leetcode.com/problems/basic-calculator-ii/submissions/>