Basic Calculator

Try to solve the Basic Calculator problem.

We'll cover the following

Statement

Given a string containing an arithmetic expression, implement a basic calculator that evaluates the expression string. The expression string can contain integer numeric values and should be able to handle the "+" and "-" operators, as well as "()" parentheses.

Constraints:

Let `s` be the expression string. We can assume the following constraints:

- $1 \leq$ `s.length` $\leq 3 \times 10^3$
- `s` consists of digits, "+", "-", "(", and ")".
- `s` represents a valid expression.
- "+" is not used as a unary operation ($+1$ and $+(2 + 3)$ are invalid).
- "-" could be used as a unary operation (-1 and $-(2 + 3)$ are valid).
- There will be no two consecutive operators in the input.
- Every number and running calculation will fit in a signed 32-bit integer.

Examples

Sample example 1

Input

```
(8 + 100) + (13 - 8 - (2 + 1))
```

Output

```
110
```

1 of 2

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Basic Calculator

1 What is the output if the following string is given as input?

(13 + 50) + (56 - 29 - (7 + 2))

A) -81

B) 81

C) 91

D) 90

Submit Answer

Question 1 of 4
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

> **Note:** We'll evaluate the expression in the following sequence:
>
> 1) Convert consecutive digits into a single operand.
>
> 2) Handle "+", "-" operators.
>
> 3) Handle the "(" bracket.
>
> 4) Handle the ")" bracket.

Drag and drop the cards to rearrange them in the correct sequence.

Initialize three variables: number to store the integer form of the current number in the string (initially 0), sign value to store a multiplication value to change the sign (initially 1), and result to

store the evaluated result of different operations (initially 0).

Upon encountering a digit character, update the number variable by multiplying its existing value by 10 and adding the integer value of the digit character to it: number = number × 10 + digit

Upon encountering a '(' character, push the value of the result variable and then the sign value onto the stack. In addition, reset the value of the sign value, and result variable to 1, 0 respectively.

Upon encountering a '+' or '-' character, change the sign value variable to 1 or -1 respectively. Then evaluate the expression on the left by multiplying the existing value of the result variable by the sign value variable and adding the number to this: result = number + (result × sign value)
In addition, reset the value of the number variable to 0.

Upon encountering a ')' character, update the result variable to evaluate the expression within the parenthesis:
result = number + (result × sign value)
Then pop the sign value and stored digit from the stack and update the result variable again:
result = (result × sign value) + digit
In addition, reset the value of the number variable to 0.

Reset | Show Solution

Implement your solution in the following coding playground:

Java

usercode > main.java

```java
import java.util.*;

public class Main{
    public static int calculator(String expression) {

        // Write your code here

        return 0;
    }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

"12 - (6 + 2) + 5"

Basic Calculator

Back

Stacks: Introduction

Next

Solution: Basic Calcul...

Mark as Completed