EHB 208E - Data Structures & Programming
Homework-2

Assignment Date :	16.04.2021
Due Date :	30.04.2021 at 18:00

Write a C program (not C++) to calculate and display the numeric result of a **Postfix Arithmetic Expression.**

In the Postfix Notation, an arithmetic operator is placed **after** two operands, and parentheses are not used.

EXAMPLE :
Infix notation : (10 + 5) / 2 + 6 - 10 / 2
Postfix notation : 10 5 + 2 / 6 + 10 2 / -
Numeric result : 7.5 + 6 - 5 = 8.5

You should use the following method for calculating the numeric result of a postfix arithmetic expression:

1) In a loop, program asks user to enter the **tokens** in the **POSTFIX** arithmetic expression one-by-one.

- A **token** is a string defined as a char array.
- A token can be either a float number **operand** (such as "**152.43**", "**2.7**", etc.), or it can be an arithmetic **operator** (such as "**+**", " -", "*****", "**/**").
- In order to finish the looping, user will enter the word "**end**" as the last token.
- There is no need to check the validity of the postfix arithmetic expression entered by user.

2) Whenever program reads an **operand** from keyboard, program **pushes** it onto a **Stack** as a float number**.** Program should convert the token string (operand) to a float number and then should push it onto the Stack. You should use the array-implemented **Stack** that can be found in the lecture slides, but you should not use the built in C++ STL stack class.

3) Whenever program reads an arithmetic **operator** from keyboard, program **pops** two operands from the Stack. Then program applies the arithmetic operation on those two operands, and **pushes** the intermediate result onto the Stack.

4) Looping finishes when all tokens are processed. Then program displays the final calculated numeric result on the screen.

EXAMPLE SCREEN OUTPUT

```
To finish, enter the word "end".

Enter a token :  10
Enter a token :  5
Enter a token :  +
Enter a token :  2
Enter a token :  /
Enter a token :  6
Enter a token :  +
Enter a token :  10
Enter a token :  2
Enter a token :  /
Enter a token :  -
Enter a token :  end

Result is :  8.50
```

EXAMPLE STACK OPERATIONS

EXAMPLE :
Infix notation : (3 + 2) * 10
Postfix notation : 3 2 + 10 *
Numeric result : 5 * 10 = 50

STEP	OPERATIONS
1	Postfix Expression : **3 2 + 10 *** Empty Stack 5 4 3 2 1 0
2	Postfix Expression : **3 2 + 10 *** Push 3 Stack 5 4 3 2 1 0 : 3
3	Postfix Expression : **3 2 + 10 *** Push 2 Stack 5 4 3 2 1 : 2 0 : 3
4	Postfix Expression : **3 2 + 10 *** Pop two times and add them Stack 5 4 3 2 1 0 2 Popped 3 Popped
5	Postfix Expression : **3 2 + 10 *** Push the result of + operator Stack 5 4 3 2 1 0 : 5 2+3 = 5 5 Pushed

STEP	OPERATIONS
6	Postfix Expression : **3 2 + 10 *** Push 10 Stack 5 4 3 2 1 : 10 0 : 5
7	Postfix Expression : **3 2 + 10 *** Pop two times and multiply them Stack 5 4 3 2 1 0 10 Popped 5 Popped
8	Postfix Expression : **3 2 + 10 *** Push the result of * operator Stack 5 4 3 2 1 0 : 50 10*5= 50 50 pushed
9	Postfix Expression : **3 2 + 10 *** No more tokens. Pop and return the result. Stack 5 4 3 2 1 0 50 Popped

IMPORTANT NOTES

- Make sure you write your full name and ID number in the C file, in the following format:

  /* @Author
  Student Name: <student name>
  Student ID : <student id>
  Date: <date> */

- You must use the template code given below.
- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- Use comments wherever necessary in your code to explain what you did.
- Note that **YOUR CODES WILL BE CHECKED WITH THE PLAGIARISM TOOLS.**

TEMPLATE CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

......

int main()
{

  ......

 // You must use a float type variable named as 'result' to print your output to screen.
 float result;

  ......

 printf("Result is : %.2f \n", result);
 return 0;
}
```