**README**

1. Running instruction

* Installation: in vscode IDE install extension “C/C++”
* Run command in terminal: g++ \*.cpp -o p && ./p.exe (for compile and run the program)

1. Test case

* Level 1: the PostFix calculator use Stack as main method
* Level 2: implement (x key word)
* Function description: exchanges the contents of the two top elements in the stack used by the calculator during the run
* Test case

+ Case 1: Input: “4 5 – “ output expected: -1

+ Case 2: Input: “4 5 x – “ output expected: 1

+ Case 3: Input: “4 5 x - 5 + ” output expected: 6

* Test case explanation

+ Case 1: normal subtract equation “4-5” with output equal -1

+ Case 2: with “x” keyword exchanges the contents of two top elements so now we have our subtract equation “5-1” with output equal 1

+ Case 3: demonstrate “x” keyword can works with other operator and number

We have “4 5 x – “equal 1 and plus 5 we value 6 as expected

* Level 3: implement (s key word)
* Function description: which calculates the sum of all the numbers currently in the stack, then removes the numbers used for summing from the stack, and finally adds the total sum to the top of the stack.
* Test case

+ Case 1: Input: “4 5 6 s “ output expected: 15

+ Case 2: Input: “4 5 6 s 5 – “ output expected: 10

* Test case explanation

+ Case 1: a sum equation “4+5+6” with output equal 15

+ Case 2: demonstrate “s” keyword can works with other operator and number

We have “4 5 6 s 5 – “ equal 10 as expected

* Level 4: implement (a key word)
* Function description: calculates the average of all the numbers in the stack. The numbers involved in the averaging are removed from the stack (all) and the result (average) is pushed on the stack
* Test case

+ Case 1: Input: “4 5 6 a “ output expected: 5

+ Case 2: Input: “4.5 0.5 0.5 a “ output expected: 1.83333

+ Case 3: Input: “4 5 6 a 5 – “ output expected: 0

* Test case explanation

+ Case 1: average of sum “4 5 6” = 15/3 equal 5

+ Case 2: demonstrate “a” keyword can works with decimal number

+ Case 3: demonstrate “a” keyword can works with other operator and number

* Level 5:
* Level description: calculates the average of all the numbers in the stack. The numbers involved in the averaging are removed from the stack (all) and the result (average) is pushed on the stack
* Test case

+ Case 1: Input: “4 5 +“ output expected: 9

+ Case 2: Input: “4 5 -“ output expected: -1

+ Case 3: Input: “4 5 \*“ output expected: 20

+ Case 4: Input: “4 5 /“ output expected: 0.8

+ Case 5: Input: “6 5 %“ output expected: 1

+ Case 6: Input: “4 5 ^“ output expected: 1024

+ Case 7: Input: “4 v“ output expected: 2

* Test case explanation

+ Case 1: demonstrate “+” operator (4+5=9)

+ Case 2: demonstrate “-” operator (4-5=-1)

+ Case 3: demonstrate “\*” operator (4\*5=20)

+ Case 4: demonstrate “/” operator (4/5=0.8)

+ Case 5: demonstrate “%” operator (6/5=1 remainder 1)

+ Case 6: demonstrate “^” operator (4^5=4\*4\*4\*4\*4=1024)

+ Case 7: demonstrate “v” operator (square root of 4 is 2)

1. Score aiming

5 points as maximum points for the project

1. Contact information

* Email: [toan.tran@tuni.fi](mailto:toan.tran@tuni.fi)
* Phone number: +358 468885032
* Address : Ritakatu, 33530 Tampere

1. Word hours: 8 hours.
2. Briefly explain verbally the design solution

* Basic idea of my solution is that, I take input from user as character by character and then I use switch case with user input character as parameter to handle function and software behavior.
* To make the program works as it is expected, I need to create 2 other functions called “check\_input” and “check\_minu”

+Boolean function **check\_input** will check valid symbol by comparing user input, character by character with valid symbol lists that I created

+ Boolean function **check\_minu** will check whether the user input is minus operator or minus number for example “-5” mean negative 5 while “5 –“mean number 5 and minus operator.

+ void calculate function

1. Level 2 “s key word” :

Stack.top(number1)

Stack.pop()

Stack.top(number2)

Stack.pop()

Stack.push(number2)

Stack.push(number1)

1. Level 3 “s keyword”

While loop{

Stack.top(number1)

Result += number1

Stack.pop()

}

1. Level 4 “a keyword”

Int a=0;

While loop{

Stack.top(number1)

Result += number1

Stack.pop()

a++

}

Result=result/a

1. Level 5 : follow math rules for each operator