PostFix calculator

# Introduction

## 1.Program compilation

* Visual Studio 2022 Preview
* C++ environment

## 2. Linking

https://moodle.tuni.fi/course/view.php?id=21010

## 3. Installation

* Install g++
* Download file postfix\_calculator from Moodle.
* Find the code from Moodle
* Set up the code from all file

## 4. Running

* Compile every file from the directory
* Type command g++ \*cpp – o p to begin the program
* There are two modes to run the program

1. Step-by step:

Type p -p

1. Directing according to the point requirements

The details will shown in the next part of this report.

## 5. Work hours

|  |  |  |
| --- | --- | --- |
| Date | Work name | Implementation time |
| 15/10/2021 | Write the initial program. Think about the requirement and make some ideas. | 3 hours |
| 17/10/2021 | Complete the Stack and test program the first time | 1.5 hours |
| 23/10/2021 | Writing dc computer program | 3 hours |
| 24/10/2021 | Testing program | 1 hour |
| 25/10/2021 | Finish the program and write Readme.docx | 3 hours |

Total hour: 11.5 hours

## 6.Contact number

Email: [hung.truong@tuni.fi](mailto:hung.truong@tuni.fi)

Tel: 0403252908

7. Aiming score: 5/5

# Testing

## Step-by-step

* Type p.exe – p on CMD

The result:

Text

Description automatically generated

Choose “?” to push:

Text

Description automatically generated

Choose “=” to print top:

Text

Description automatically generated

Choose “+”:

Text

Description automatically generated

Choose “-”:

Graphical user interface, text

Description automatically generated

Choose “\*”:

Text

Description automatically generated

Choose “/”:

Graphical user interface, text

Description automatically generated

Choose “s”:

Text

Description automatically generated

Choose “a”:

Text

Description automatically generated

You can try more arthmetic method with “%” “v” “^”

## Directing with the point requirement

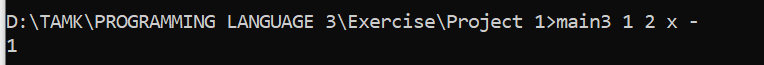
Chose each type of method and the next line is the result.

#### One-point job requirements

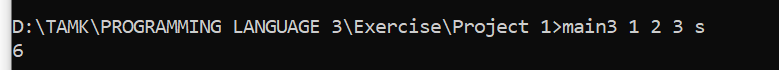
Text

Description automatically generated

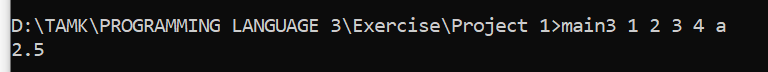
#### Two-point job requirements



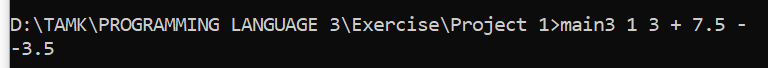
#### Three-point job requirements



#### Four-point job requirements



#### Five-point job requirements

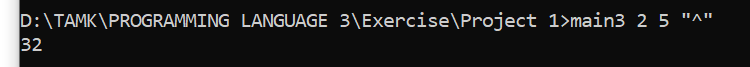


* Find remainder operation with %

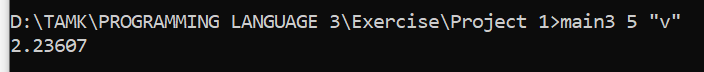
Text

Description automatically generated

* Find power increase with ^:



* Find square root with v:



Remember to type “v” and “^” in CMD because the CMD cann’t execute without double quote.

# Conclusion

1. There are 5 files in my project.

Text

Description automatically generated

1. The main idea:

A picture containing graphical user interface

Description automatically generated

* The Stack has attributes from Utility file.
* The Stack.h has kind of functions: empty( check Stack is empty or not), push(push an element to the Stack), pop( pop the last element of the Stack), top(show the last element of the Stack) and size(show the Stack size)
* The main3 has atrributes from Stack file.
* Functions in the main3:
  + get\_command: to receive command from the input of the user:

cA screenshot of a computer

Description automatically generated with medium confidence

* do\_command: receive command and do according to command as we decribe in the instruction. For example, in the case “?”(push an item to the Stack)

Text

Description automatically generated

* Is\_Number: determine a character is number or not.
* Introduction():

Graphical user interface, text, application

Description automatically generated

* Instructions():

Show a list of method that the user can use to calculate.

Text

Description automatically generated

In the content of main3, the program is written by two methods:

Firstly, check the second string. If the second string is equal to “-p”, the program will show all the content of instruction function as above and the user can use the program step by step. For example, the user input the command and the program will run it.

Text

Description automatically generated with low confidence

Secondly, if the second string is not equal to “-p”, the program will check whether the second string is number or not. If it is the number, the program will push this number to the Stack. If it is not the number, the program will implement according to the command that user inputs. After receiving enough data, the program will show the last result.

A screen shot of a computer

Description automatically generated with low confidence