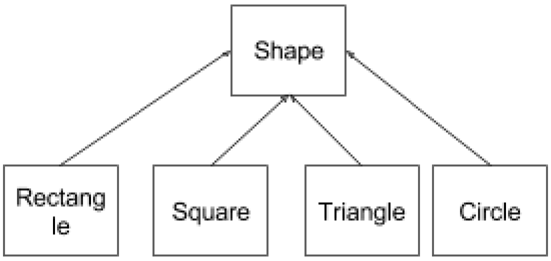
Programing Language C++

TP8 (2017) for Group B and C

Question 1. Make a program with five class: 1. Shape, 2. Rectangle, 3. Triangle, 4.Square and 5.Circle. Using the concept of heritage and polymorphism make a program that calculates the area. Requirements:

1. The class Shape has the proprieties: width, height, and radius;
2. The method area must be declared in the class shape, and it haven't any implementation;
3. All descendent class must implement the method areas correctly, for that the type of shape must be regarded.
4. Create the function main to illustrate the functionality. In the main declare four objects: rectangle, square, triangle and circle. All class have the class Shape as ancestral class, But the each initialization must be done using the respective class: Rectangle, Square, Triangle and Circle.



Question: In order of the get the correct result you must use the instruction virtual member! Why? Where the virtual member must be declared? Execute the program with the virtual instruction ative and not active (with and without virtual), what the difference of results? Make a new implementation that virtual has a interference in the result.

Question 02: Create a program to generate a text file with text informed by the user, and the user type what who wants and the program saves it in a file. After that, the program must sum up the quantity of each character found in the text and shows on screen in alphabetical order list. Ex:

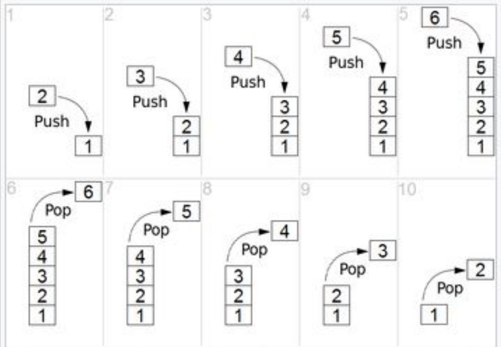
File_Question_02.txt
This is just a simple Text.

Output:

```
a = 1
e = 2
h = 1
i = 3
j = 1
l = 1
m = 1
p = 1
s = 4
t = 4
u = 1
x = 1
```

OBS: Considering just letters and number, ignore the special characters. The UPPERCASE and LOWERCASE are not differentiated.

Question 03: The Stack is a type data structure very simple, it is used to collect and store data in a specific way, the idea of the stack is that the data are put in follow FILO discipline, FILO means first in, last out! A Stack has the operator's: pull and push. The Push add data, and the Popl get the data. When the operator Pull is called the last data include by the Pull operator is got. The figure below illustrate these operations.



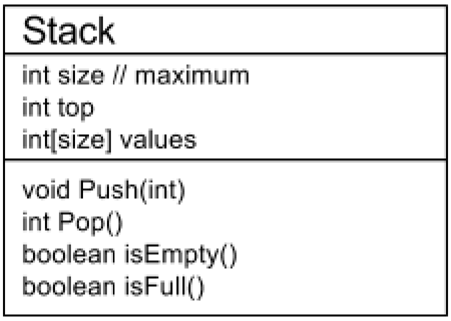
Simple representation of a stack runtime with *push* and *pop* operations.

source: https://en.wikipedia.org/wiki/File:Lifo_stack.png

A Stack is a dynamic structure (Class), and you can use the operator Push and Pop many times it is necessary, but some requirements must be respected:

- It is impossible to get data when the Stack is empty.)
- It is impossible to add data when the Stack is full;

Therefore, a Stack has two critical states: Empty and Full. If the Stack is not full, it is possible to increment a new data, and if the Stack is not empty it possible recover data from it. (Use the propriety “top” to know if the a stack is empty or full). A way of design a Stack is shown below:

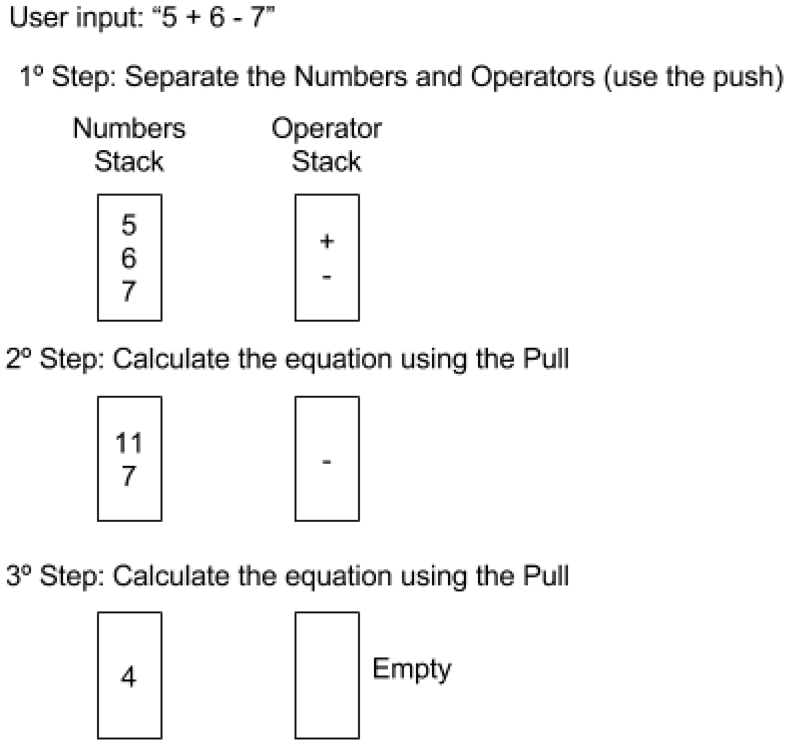


Stack

int size // maximum
int top
int[size] values

void Push(int)
int Pop()
boolean isEmpty()
boolean isFull()

Activity: Develop a program using two Stack, the first is used to store numbers and the second to store operators (use char to store an operator). The program must receive a basic arithmetic operation like: “5+5+6-7” from the user. After the program must decompose the string into two stacks, and the numbers are stored in the Stack_Number and operators are stored in the Stack_Operator. The calculation of the equation must use the two stacks. Then, develop a program in C++ to resolve a basic arithmetic equation (use just integer number and operations +, -, * and /) using the two stack. The sequence is shown below:


User input: “5 + 6 - 7”
1º Step: Separate the Numbers and Operators (use the push)
Numbers Stack
Operator Stack
5
6
7
+
-
2º Step: Calculate the equation using the Pull
11
7
-
3º Step: Calculate the equation using the Pull
4
Empty

Observation:

1. The user can enter a string with many operators and operations.
2. The program must inform if the user informed an incorrect string, for example: “5 + ” or “ 6 + 4 +” or “ + +”. Note, since the calculation, if the Stack_Number is empty and the Stack_Operator is not empty, then the User informed a wrong input.
3. The class Stack must be implemented, then don’t use any ready Class from C++ libraries to do it. You must store data in the stack using a simple array int[] values.

Question 04. Implement a class Vehicle and Car. The class Vehicle is an abstract class, and the Class Car is a Vehicle specialization, so it must implement the methods. Each Car has the propriety consume and fuel tank capacity. For example, a car's fuel tank stores a maximum of 50 liters of gasoline and car consumes 15 km/liter. It should be possible to:

1. Refuel the car with a certain amount of gasoline;
    a. Test if the quantity is possible, that is if the is space in the tank.
2. Move the car at a certain distance (measured in km);
    a. If the distance is more than the capacity the method must prompt in the screen the distance traveled and warning that the car is stopped.
3. Return the amount of fuel and the total distance traveled.

The table below is an example of use:

Method	Example	Warning
Create	Car = car1(50,10) // tank 50L, 10 km/l	
Refull	car1.refuel(10) // fill up 10L	
Move	car1.move(110)	Car traveled 100Km and Stopped. Remain distance 10km
Refull	car1.refuel(60)	Error: 60L is not possible, capacity maxim 50L
Info	car1.getInfo()	Traveled: 100Km Fuel: 0L

In the main program, create 2 cars. On creation the 2 objects car the value of fuel tank capacity and consume must be passed through the class constructor.

After, Fill 20 liters in the first and 30 liters in the second. Move the first in 200 km and the second in 400 km. Display the distance traveled and the total fuel remaining for each.

GOOD JOB!