PROGRAMMING 1:

FIRST PRACTICAL EXAM

November 17th , 2016

Full name _____ Exam length: 1 hour and 30 minutes.

Implement a basic calculator (integer numbers) using the C programming language. The program will ask the user to enter two input numbers and an operator. If the entered numbers or operator are not valid, the program must show an error message and ask again the user to enter input data. Otherwise, it shows the result of the operation. Numbers must be values between 0 and 1000, if the user enters the number '9999' a random number between 1 and 1000 will be generated. Operators must be one of the following symbols: '+', '-', '*','/'. Once the calculator has performed the first calculation, the program will ask the user to enter one of the following options: continue doing calculations '**c**', new calculation '**n**' or quit '**q**' the program. If the user chooses **'c',** the program will ask for an operator and an input number, and it will perform that operation using the previous result as the first input number. Otherwise, if the user chooses new calculation **'n'**, the program will ask again the user to enter two input numbers and an operator, previous result will not be used. Finally, if the user enters the quit option, **'q'**, the program will exit.

In addition to the *main()* function, the student must implement, at least, the following modules:

a) A module for validating input numbers: 'checkNumber'. It takes as an input an integer value, and returns whether this input is valid (true) or not (false).

b) A module for validating input operators: 'checkOperator'. It takes as an input a character, and returns whether this input is valid (true) or not (false).

c) Modules for performing addition, subtraction, multiplication and division of the input numbers

- `int sum_(int n1, int n2);`
- `int sub_(int n1, int n2);`
- `int mul_(int n1, int n2);`
- `int div_(int n1, int n2);`

The multiplication and division operators must be implemented without using the C programming language instructions '/' and '*', only the '+' and '-' operators are allowed. Remember that the division and multiplication operators can be implemented as a sequence of additions and subtractions.

d) A module that ask the user to enter an input number and checks that the input number is valid. It will keep asking the user to enter a number until it is valid: 'read_input_number'.

e) A module that ask the user to enter an input operator and checks that the input operator is valid. It will keep asking the user to enter an operator until it is valid: 'read_input_operator'.

f) A module that generates and returns a random number within a certain range: 'generatesRandomNumber'. To generate random numbers include the headers *ctime* and *cstdlib*. Remember to initialize the random number generator using the instruction *srand(time(0));* The *rand()%500* instruction generates random numbers between 0 and 499.

Division by zero must be handled by our software displaying an error to the user.

IMPORTANT:

- Source file must me **named** using your **initials**, example: Pedro Garcia Roca, ***pgr.c*** and uploaded to the **Moodle platform.**
- You should choose the most convenient type of module for each case, and a proper type for its parameters
- **YOU MUST NOT** use global variables
- Code must be properly **indented**
- We will consider the use of comments describing the program to evaluate the exam
- A non---compiling program will penalise a 20% of the mark

EXECUTION EXAMPLE

```
Programming I Calculator
========================
Input number 1: 1
Operator: *
Input number 2: 3
Result:3
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: c

Operator: +
Input number 2: 50
Result:53
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: c

Operator: !
Wrong input operator
Operator: 3
Wrong input operator
Operator: /
Input number 2: 10
Result:5
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: n

Input number 1: 3
Operator: -
Input number 2: -3
Wrong input number: -3
Input number: 1001
Wrong input number: 1001
Input number: 2
Result:1
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: n

Input number 1: 1
Operator: +
Input number 2: 9999
Random number 57
Result:58
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: n

Input number 1: 3
Operator: /
Input number 2: 0
Ops.. Something went wrong
Input number 1: 3
Operator: /
Input number 2: 1
Result:3
Do you want to [c]ontinue, [n]ew operation, [q]uit the app: q
```