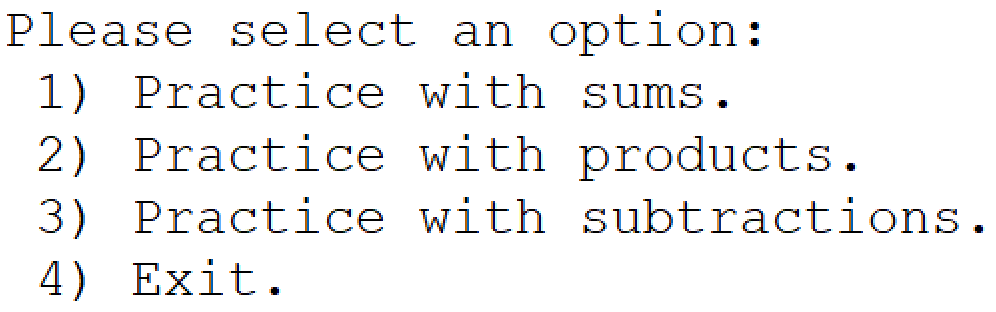
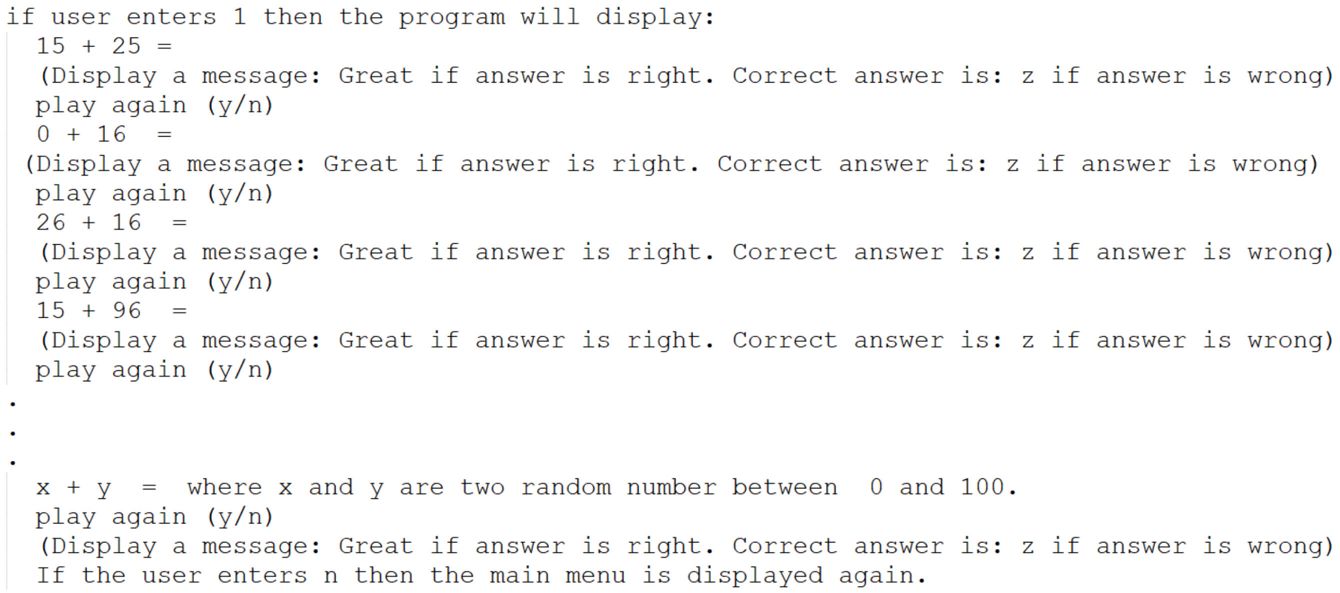
PROGRAMMING LOGIC AND DESIGN.
MIDTERM EXAM GRADE = MIDTERM PROJET GRADE.

The grade obtained in this activity will be assigned to your midterm exam and midterm project. Create a lab report for the following programming challenge.

Create a program that will display, initially, the following menu:



```
Please select an option:
 1) Practice with sums.
 2) Practice with products.
 3) Practice with subtractions.
 4) Exit.
```



```
if user enters 1 then the program will display:
  15 + 25 =
  (Display a message: Great if answer is right. Correct answer is: z if answer is wrong)
  play again (y/n)
  0 + 16  =
 (Display a message: Great if answer is right. Correct answer is: z if answer is wrong)
  play again (y/n)
  26 + 16  =
  (Display a message: Great if answer is right. Correct answer is: z if answer is wrong)
  play again (y/n)
  15 + 96  =
  (Display a message: Great if answer is right. Correct answer is: z if answer is wrong)
  play again (y/n)
.
.
.
  x + y  =  where x and y are two random number between  0 and 100.
  play again (y/n)
  (Display a message: Great if answer is right. Correct answer is: z if answer is wrong)
  If the user enters n then the main menu is displayed again.
```

Repeat, the above steps, for options 2 and 3.

Requirements:

- For multiplications one of the factors must be between 0 and 10. The other between 0 and 100.
- For subtractions, be sure to display the numbers in a way that the correct answer is always positive. Do not display something like 5-10. You should display 10-5.
- Run your program at least 10 times for each mathematical option and report the output from your program and input from the user.

- When the user enters and answer, display total questions X, Total correct answers Y, Total wrong answers Z.
- Implement modularity as much as possible.
- Create a txt file with the C++ code and submit it. Name it math.txt

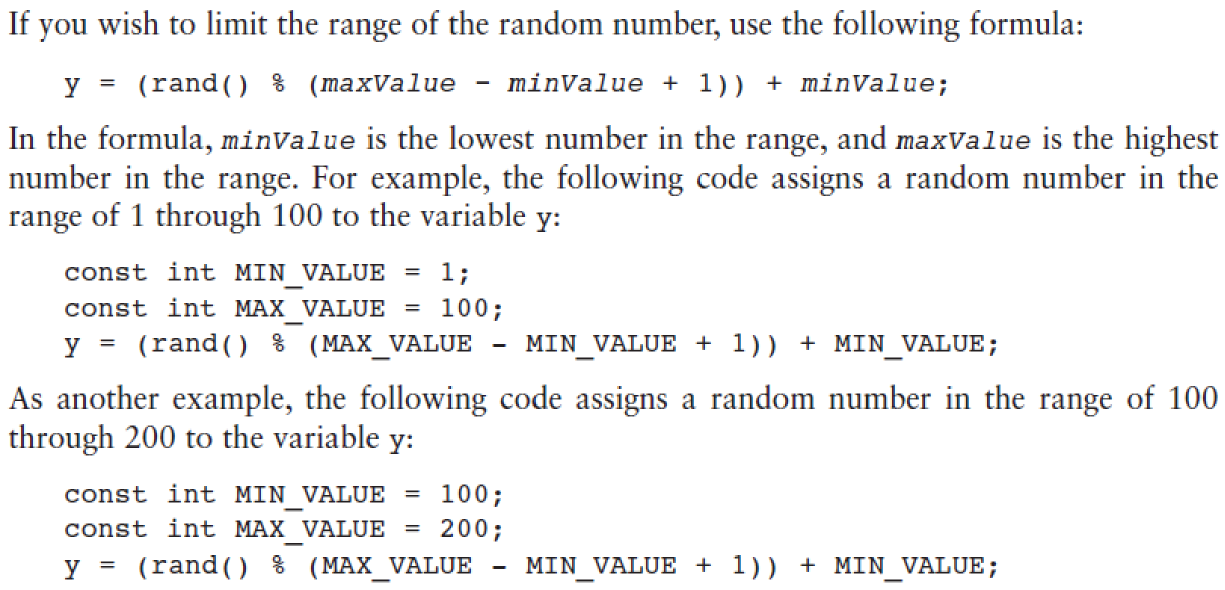
Your lab report must contain the following sections.

- Program requirements. (5 points)
- Program Analysis. (5 Points)
- Program Design. (10 Points)
- Implementation. (20 Points)
- Testing. (60 points).

Testing Requirements:

- Run your program and demonstrate with pictures and comments that every option of your menu is working.
- Run your program and show at least 10 operations for each option. Document output, inputs and total results for each try.
- Document, with pictures and comments, that your program returns to the main menu when the user terminates with a particular operation

References:



If you wish to limit the range of the random number, use the following formula:

```
y = (rand() % (maxValue - minValue + 1)) + minValue;
```

In the formula, `minValue` is the lowest number in the range, and `maxValue` is the highest number in the range. For example, the following code assigns a random number in the range of 1 through 100 to the variable y:

```
const int MIN_VALUE = 1;
const int MAX_VALUE = 100;
y = (rand() % (MAX_VALUE - MIN_VALUE + 1)) + MIN_VALUE;
```

As another example, the following code assigns a random number in the range of 100 through 200 to the variable y:

```
const int MIN_VALUE = 100;
const int MAX_VALUE = 200;
y = (rand() % (MAX_VALUE - MIN_VALUE + 1)) + MIN_VALUE;
```

At the end of each studied chapter there is a section on C++ basic commands used for our implementations.

Any question? Just email me.

Enjoy this project!