EECE 231 — Introduction to Programming Using C++ and MATLAB, Sections 3, 4, and 12
Quiz I

Oct 13, 2015

- The duration of this exam is **1 hour and 45 minutes**.

  Keep in mind that you need around **10 minutes** at the end of the duration of the exam to **upload** your answers to the Moodle website. It is your responsibility to make sure your files are correctly uploaded.
- The exam consists of 3 problems for 105 points.
- You can use all the material available on Moodle.
- You are **NOT** allowed to use the **web**. You are not allowed to use **USB's** or files previously stored on your machine.
- If you get caught violating the above rules or if you communicate with a person other than the exam proctors during the exam, you will immediately get zero and you will be referred to the appropriate disciplinary committee.
- Active cell phones and any other unauthorized electronic devices are absolutely not allowed in the exam rooms. They should be turned off and put away.
- The questions are in an increasing order of difficulty.
  Plan your time wisely. Do not spend too much time on any one problem.
  Read through all of them first and attack them in the order that allows you to make the most progress.
- Submit your solutions each part in a separate file as indicated in the booklet. Include your name and ID number in each file.
- Good luck!

Problem 1 (35 points). Flipping Arrays

Write a C++ function that takes as parameters an array of integers and its size. The function is supposed to flip the array about its center. This means that, for an array size N, the elements in position 0 is taken to position $N-1$, and the one originally in position $N-1$ goes to position 0. The element in positions 1 and $N-2$ are also swapped, and so on. For example, $\{2, 3, -7, 5, 0, -1, 6\}$ should becomes $\{6, -1, 0, 5, -7, 3, 2\}$. Name this function *flipArray*. Note that this function is not about printing the array in reverse order. It actually re-orders the array. **(20 points)**

In your *main* program:

a) **(3 points)** Define and initialize and array of integer having a known size.

b) **(5 points)** Display the elements of the array.

c) **(5 points)** Call the *flipArray* function passing the initialized array as a parameter.

d) **(2 points)** Now display the modified array.

Submit your solution in a file called Prob1.cpp including your name and ID number.

Problem 2 (35 points). Circles

Write a C++ program that prompts the user to enter the coordinates x_0 and y_0 of the center of a circle and its radius R. x_0, y_0 and R are all of the *double* data type. **(5 points)**

After reading x_0, y_0 and R, you are asked to do the following:

a) **(10 points)** Ask the user to enter the coordinates x_p, y_p of a point, and then check whether or not it is inside the circle. A point on the periphery of the circle is considered inside the circle.

b) **(20 points)** Use nested loops to find and print all the points with *integer* coordinates inside the circle. To get full credit, your loops should do the minimum number of iterations. For this, you have to pay attention to the initial and stop values of your loops counters.

In a) and b), to compute the distance of a point to the center of the circle, you can use the function *distance* you implemented in Problem 1 of Lab Assignment 4.

Your program's input and output should be formatted as in the following sample:

```
Enter coordinates x0 and y0: 7.9 4.2
Enter radius R: 3

Enter coordinates xp and yp: 5.0 1.3
No. The point (5,1.3) is not inside circle.

The points with integer coordinates inside the circles are:
(5,4) (6,2) (6,3) (6,4) (6,5) (6,6) (7,2) (7,3) (7,4) (7,5) (7,6) (7,7) (8,2) (8,3) (8,4)
(8,5) (8,6) (8,7) (9,2) (9,3) (9,4) (9,5) (9,6) (10,3) (10,4) (10,5) (10,6)
```

Submit your solution in a file called Prob2.cpp including your name and ID number.

Problem 3 (35 points). Series Expansion of cos(x)

The series expansion of $\cos(x)$ is given by:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \cdots = 1 + \sum_{k=1}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!} \tag{1}$$

A very good approximation of $\cos(x)$ is obtained by taking the first few terms of the series. Let's take the terms up to $k = 10$. In this case, we have

$$\cos(x) \simeq 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \cdots + \frac{x^{20}}{20!} = 1 + \sum_{k=1}^{10} \frac{(-1)^k x^{2k}}{(2k)!} \tag{2}$$

In Equations 1 and 2, the (!) denotes the factorial of a number. The factorial of an integer n, denoted $n!$, is given by: $n! = 1 \times 2 \times 3 \times \cdots \times n$. Note that $0! = 1$ by definition, and $1! = 1$.

In this problem, you are asked to:

a) **(15 points)** Write a C++ function, called *factorial*, that takes as parameter an integer and returns it factorial. You can use a *for* loop to implement this function.

b) **(20 points)** In your *main* program, prompt the user to enter a variable x of the *double* data type. After reading the value of x, implement Equation 2 using a *for* loop, to compute the cosine of x. Note that you must not use the *cos()* function from the *cmath* library. Test your program for $x = 0.5236$ $(= \pi/6)$ and for $x = 1.047197$ $(= \pi/3)$.

- *Hint:* Inside the *for* loop that implements Equation 2, you need to call the function *factorial* in each iteration to compute the denominator of each term.

- Notes: In Equation 2, x is in radians not degrees.
  To earn full credit, try to implement Equation 2 without using the *pow()* function from the *cmath* library.

Submit your solution in a file called Prob3.cpp including your name and ID number.