Northeastern University
College of Engineering
Department of Electrical & Computer Engineering

EECE7205: Fundamentals of Computer Engineering

Fall 2019 - Homework 1

Instructions

- For programming problems:
  - Your code must compile and run on the COE Linux server before submitting it on Blackboard.
  - Your code must be well commented by explaining what the lines of your program do. Have at least one comment for every 4 lines of code.
  - At the beginning of your source code files write your full name, students ID , and any special compiling/running instruction (if any).

- Submit the following to the homework assignment page on Blackboard:
  - Your homework report submitted as one PDF file. The report includes the answers to the non-programming problems and the screen shots of your program's sample runs for the programming problems. Your report must be developed by a word processor (no hand written or drawn contents are acceptable).
  - Your well-commented source code file(s) for the programming problems.
  - Do NOT submit your files (the PDF and source code) as a compressed (zipped) package. Rather, upload each file individually.

Note: Yon can submit multiple attempts for this homework, however, only your last submitted attempt will be graded.

Problem 1 (20 Points)

Write a C++ program to implement and test two functions: SwapP and SwapR. SwapP swaps the values of two integer variables using pass-by-pointer. SwapR swaps the values of two integer variables using pass-by-reference.

Problem 2 (20 Points)

Replace the question marks in the following C++ program with C++ code to swap the p1 and p2 pointers to point to each other's values.

```
# include <iostream>
using namespace std;
int main ()
{
  int x = 1, y = 9;
  int *p1, *p2;
  p1 = &x;
  p2 = &y;

  ??????????

  cout << *p1 << " and " << *p2 << endl; // Prints "9 and 1"
 return 0;
}
```

Problem 3 (60 Points)

Write a C++ program that starts by asking a teacher for the size of his class (the number of students). Then the program asks the teacher to enter for each student his/her name along with his/her grade in an exam. All grades are integers that have to be in the range from 0 to 100 (inclusive). Store the names in an array of strings and the grades in another array of integers. Use dynamic memory to create both arrays.

Define a function that implements the sorting algorithm `INSERTION-SORT(A,n)` as explained in class. Call this function to sort the above arrays in a descending order of the students' grades in the exam.

Your program will display the sorted list of students' names along with their grades.