INFSCI 0020 Program Design and Software Tools

Homework 1

Due: Friday, 5PM Jan 23

1. *Selection Sort* (Read Exercise 4.31) [**40 Points**]

   Write two functions IterSort and RecSort that implement the selection sort algorithm described in 4.31. Function IterSort implements it using iterative approach whereas function RecSort implements it using recursion. In the main program, you should create an array of pointers to the two functions.

   For user interface, your program should print the menus of available functions as follows:

   Function Menu
   [1] Iterative Sort (Generate 10 numbers randomly between 1 and 100)
   [2] Recursive Sort (Generate 10 numbers randomly between 1 and 100)
   [3] Iterative Sort (User inputs 10 numbers between 1 and 100)
   [4] Recursive Sort (User inputs 10 numbers between 1 and 100)

   Enter Choice: <*user will enter a number between* 1 *and* 4>

   If the user chooses a 1 or 2, your program should generate 10 random numbers between 1 and 100. Use srand() function to generate the random numbers - refer to example given in pages 186-189.

   If the user inputs choices 3 or 4, the program should ask the user to input 10 numbers that are less than 100 - make sure you catch wrong inputs.

2. *Palindromes* (Read Exercise 4.32):

   For this problem also, create two functions with appropriate names. One uses the iterative approach and the other uses the recursive approach. Again, use an array of pointers to the two functions.

   Your program should have a *sentinel* controlled do/while loop that prompts a user to enter a string. Allow the user to choose one of the two functions to use after he has entered the string to be checked.

3. Write a program (or two separate programs) that implements the two functions that correspond to the following exercises in the text book:
   a. Exercise 5.41 (two separate functions for **strlen**)
   b. Exercise 5.46

   You may make it a one program and provide proper user interface to allow users to choose the function he wants to use. Or you may make them two separate programs.

Submission:

1. Submit all files through e-mail, possibly zipped, to the GSA.
2. Provide a concise document (2 pages at the most) that provides any information regarding the steps needed to run the program. Anything that will help ease the running of your program, or identify any peculiarities can be included.

Grading Notes:

About 10% of points will be given for following "good programming practices," such as proper documentation, proper naming of variables, modular organization of the program, proper indentations, etc. Please pay special attention to the "*Good Programming Practices*" annotations in the book.