Programming Fundamentals – Spring 2013

(BS-SE-F12 Morning & Afternoon)

Lab # 10

Instructions:

- Indent your code properly.
- Use meaningful variable and function names. Follow the naming conventions.
- Use meaningful prompt lines/labels for all input/output that is done by your programs.

Task # 1

Write a C++ program for solving **Programming Challenge # 13** (Chapter 7) on Page 446-447 of your textbook.

Task # 2

Implement a function **`specialSearch`** which takes a one-dimensional array of integers, its size, and an integer *key* as arguments. This function will determine the number of elements less than, and number of elements greater than *key* in the given array. The prototype of your function should be:

```
    void specialSearch (int arr[], int n, int key,
                        int& numLess, int& numGreater)
```

In the above function prototype: **`arr`** is an array which contains **`n`** integers in *unsorted* order, **`key`** is the value based upon which the searching will be performed, **`numLess`** and **`numGreater`** are reference parameters which will be used to return the counts of the number of elements less/greater than the **`key`**.

Task # 3

Implement a function **`findMedian`** which takes an *unsorted* one-dimensional array of integers and its size as parameters, and determines (and returns) the **median** element. The prototype of your function will be:

```
                int findMedian (int arr[], int n)
```

Remember that the **median** of a given collection of n numbers is a number k such that the number of elements less than k is equal to the number of elements greater than k.

You can assume that:

- The no. of elements (n) in the array is odd
- All the elements of the array are distinct

Hint: *Use the function* **`specialSearch`** *that you implemented in Task # 2 above.*

Task # 4

Write a C++ program for solving **Programming Challenge # 2** (Chapter 8) on Page 487 of your textbook.

Task # 5

Modify the function **`findMedian`** that you implemented in **Task # 3** so that it works for even values of *n* as well. Then, modify the function so that it works correctly even if the elements are not distinct.

Task # 6

Write a C++ program for solving **Programming Challenge # 8** (Chapter 7) on Page 444 of your textbook.

Task # 7

Write a C++ program for solving **Programming Challenge # 12** (Chapter 7) on Page 446 of your textbook.

Task # 8

Write a C++ program for solving **Programming Challenge # 17** (Chapter 7) on Page 447-448 of your textbook.

Task # 9

Write a C++ program for solving **Programming Challenge # 18** (Chapter 7) on Page 448 of your textbook.

Task # 10

Write a C++ program for solving **Programming Challenge # 3** (Chapter 8) on Page 487 of your textbook.

Task # 11

Write a C++ program for solving **Programming Challenge # 7** (Chapter 8) on Page 488 of your textbook.