Lab 10 – One-D Arrays

Topics to be covered:

- One-D array
- Passing array to function
- (string comparison, length, substring…)

Example code:

- OneDArray.cpp

Pieces of code:

```
************** array declaration ********************
int array[25];
char ch[5];
************** access to elements in array ***********
//initialization
for (int i = 0; i < 5; i++)
{
     cout << "Please input one character\n";
     cin >> ch[i];
}
//printing out
for (int i = 0; i < 25; i++)
     cout << array[i] << " ";
cout << endl;

************** pass array to a function **************
//function prototype
void initialize (int [], int );
int max (int []);

//function call
initialize (array, 25);
int maxValue = max (array);

//function definition
void initialize (int arr [], int size)
{
     for (int i = 0; i < size; i++)
          array[i] = rand () % 100 + 1;
}
int max (int array[25])
{
     int max = array[0];
     for (int i = 1; i < 25; i++)
          ...
     return max;
}
```

Lab 10

<u>Exercise 1:</u>

Type up a reflection of the process you went through in designing and writing labs 8 and 9. It should contain what you learned, what you would redo if you did it again, how you would improve it and how it could be extended. Include a discussion of how difficult you found this project, what was the most difficult hurdle for you and your thoughts on your design and programming skills at this point in the course. *The minimum length of your write-up: 1 page, double spaced.*

<u>Assignment 1</u>:

Write a complete C++ program that presents the user with a menu and processes the below options until the user indicates that he/she wishes to stop.

Option 1: Call a function to play with an array
Option 2: Call a function to modify a phrase

Option 1 described:

This function should create an array of size 50 containing randomly generated letters within the range [A, Z]. The program should write the randomly generated array to an output file named by the user. Determine the maximum and minimum values and the indices of their first occurrence and also the number of times each occurs. Write the values, number of occurrences and indices to the output file. Sort the array in ***increasing*** order using a "Bubble Sort" and write the sorted array to the output file. Bubble Sort is described in your text book on pages 1152 – 1154.

Option 1 ***must contain***, minimally, ***functions*** to:

Initialize the array - fill the array with random letters
Determine the max, the index of its first occurrence and the number of times it occurs
Determine the min, the index of its first occurrence and the number of times it occurs
Perform the bubble sort on the array
Write the contents of the array with 10 numbers per line (5 lines total)

Option 2 described:

Continually prompt the user for a phrase (sentence) and letter (terminating character) until the user enters the phrase "stop". Search each phrase for the first occurrence of the letter in the phrase and modify the phrase so it terminates with the character preceding the user's chosen letter. Write the modified phrase out to the screen. In the event that the letter does not appear in the phrase write an appropriate informational message.

Example: (italics indicated user entered information)

The cow jumped over the moon
u
The cow j

Hello there
X
The letter X does not appear in the entered phrase

stop
You have chosen to stop, goodbye

Option 2 **must contain** at least two **functions**: