CSCI – 297 Systems Programming

Assignment 1

Due 11:59 PM, Oct 3, 2023

In this assignment, you will be working on implementing a couple of functions that will work with arrays in C. You will be writing these functions as a library (Task 1) and will include them in your test file (Task 2). You will also write a makefile to compile your test and library files (Task 3). You should complete all your work for this assignment and any other future assignments in the AWS EC2 virtual machine.

Task 1:

In this task, you will be implementing a couple of functions in a C file named **`myarray.c`** After implementing your functions, you should also create any additional files required to convert the set of functions into a library. The functions you are required to implement are listed below. You have also been provided with a prototype that you should use for each of these functions, which specifies the input and output parameters and corresponding types.

- Write a function that will copy the values from one array to another array.

```
void copyArray (int fromArray[], int toArray[], int size);
```

- Write a function that will sort an array in ascending order. You may use whatever sorting algorithm you wish.

```
void sortArray (int arr[], int size);
```

- Write a function that will perform a linear search on the unsorted array. The function is to "return" two values. The first should be the position in the array where the value was found or -1 if the value was not found. The second is the number of comparisons needed to determine if/where the value is located in the array.

```
int linearSearch (int arr[], int size, int target, int* numComparisons);
```

- Write a function that will perform a binary search on the sorted array. Similar to linear search this function will also return two values. The first should be the position in the array where the value was found or -1 if the value was not found. The second is the number of comparisons needed to determine if/where the value is located in the array.

```
int binarySearch (int arr[], int size, int target, int* numComparisons);
```

Task 2:

In this task you will implement your tester file for your library. You should name your file **`main.c`.** For this file, you will be writing all your code inside one main function. The requirements of the main function are as follows:

- Read an integer from standard input and store the values into a dynamic array. The array should be able to grow in size if it becomes full. You should use the scan function to continue reading data inside a loop. You should terminate the loop when it sees a value of -1. You can test this functionality with the sample data file provided by piping it as input to your compiled executable in the command line.

- Next, make a copy of the integer array using the array copy function that you have implemented inside your library from task 1.

- Sort one of the arrays using the sorting function you implemented. Leave one of the arrays unsorted.

- Next, within your main function, prompt the user to input a number to search in the arrays. Using both searching functions, you implemented in your library, search for the number in the array, and print out the following information, for both, linear and binary search.

    - The value being searched, and the type of search.
    - Whether the item was found, or not found.
    - The number of comparisons need to search through the array.
    - The index of the item if it was found.

Task 3:

After testing your library and your main function, write a **`makefile`** that automatically compiles and links your program. In this file, you should include the following four entries:

- An entry for program, that links all object files
- An entry compiling the **`main.c`** file
- An entry for compiling a library archive (.a) for **`myarray.c`**
- An **`all`** and **`clean`** entry.

Submission Instructions:

Compress all three files from the tasks in a zip archive named **`assignment._1.zip`** and submit on Canvas.