Lab 5: Arrays

Recommended Reading: Chapter 6

Mandatory Additional Reading: Array Notes, see Chapter 6 Module

Due Thursday, Feb 19 before 2:00 pm. Canvas submissions for this lab close at 2:10 pm.

Lab Time Ends at 2:00 pm. You should have submitted something on Canvas by that time.

File # 1 : arrayfunctions.h

1. Preprocessor wrapper
2. Prototypes for the 5 functions listed below.

File # 2 : arrayfunctions.c

1. void bubbleSort (int array[ ], int size) – complete problem 6.11 on page 268, both a) and b).
2. void fillArray (int array[ ], int size, int max) – fill the array with random values between 1 and max (inclusive). Use time(NULL) to seed the random number generator.
3. int linearSearch(int array[ ] , int size, int key) – if key is found in the array, return the index of the first element where it was found. If key is not found, return -1.
4. void printNperline (int array[ ] , int size, int n) – print the elements of the array, n elements per line with two spaces between values.
5. void shiftElements (int array[ ] , int size, int count) - shift all of the elements by count positions (wrapping elements around to the beginning as needed). If count is positive, shift elements to the right. If count is negative, shift elements to the left.

File # 3 : lab5.c This is a minimal list of tests that you should make.

1. Declare an array of 50 elements. Use the fillArray function to fill the array with random values between 1 and 1000.
2. Print the unsorted array using the printNperline function. (Try several different values for n when you test the printNperline function.)
3. Sort the array with the bubbleSort function, then print the sorted array.
4. Search the array using the linearSearch function for the value 82 and print the result.

5. Search the array for the value in the last array element and print the result.
6. Shift the array to the right 6 elements, using the shiftElements function, then print the array.
7. Shift the array to the left 4 elements, then print the array.
8. Include additional code to test the functions thoroughly.

File # 4 : makefile

Targets and appropriate compiler commands to produce an executable named **lab5** from the 3 files listed above.

*** There is no partial credit on the makefile. If it doesn't correctly produce the executable named lab5, your grade will be deducted 3 points. ***

Submit arrayfunctions.c, arrayfunctions.h, lab5.c, and makefile on Canvas.