

CS 150 Lab 12

Arrays and Functions

The purpose of today's lab is for you to get more experience working with arrays and functions.

- Be sure your output looks exactly like the specified output
- Be sure to submit your solution to **CS150-01 Drop** when you are done by Friday at 5pm

In the folder CS150Public, you will find a file called **12_1_ArraysPlusFunctionsClean.cpp** which contains a C++ program that has several function prototypes and a main function that contains code to test the function definitions that you must write.

Create an empty file **main.cpp** in a new project **12_1_ArraysPlusFunctions**, then copy all of the code from **12_1_ArraysPlusFunctionsClean.cpp** into **main.cpp**. When you first compile and run the program, you will get two warnings about two unreferenced local variables that you should ignore for now.

Be sure to write the function definition for each function prototype one at a time. For example, for the first function `printTitle`, you would:

- Write the `printTitle` function definition after `main`.
- Uncomment the first function call to the `printTitle` function: `printTitle ("Printing Array A");`
- If this first functional call works as expected, uncomment **all code** in `main` that calls the `printTitle` function.

The function prototypes and descriptions are listed on the following page, and the correct output is listed on the third page. **The function prototypes are arranged in order of increasing difficulty, so we suggest that you develop them in this order.**

Some important notes:

- 1) In those functions that increase or decrease the number of elements in the array, both the array (by default) and the number of elements are passed by reference. Make sure that the appropriate parameter in your function is updated to accommodate for any change in number of elements in the array.
- 2) Also, note that the function `copyArray`:

```
void copyArray (int arryDest[], int &sizeDest,  
               int arrySrc[], int sizeSrc, int max)
```

MUST call the function `insertLastArray`:

```
void insertLastArray (int arryDest[], int &size, int max, int value);
```

for EACH number that is inserted into the array `arryDest[]`. Also, the function `copyArray` should overwrite any existing values in the array `arryDest[]`. Furthermore if `sizeSrc < sizeDest`, you do not need to worry about resetting any extraneous values in `arryDest[]` that may have already been in this array.

Function Prototypes and Descriptions

```
// printTitle prints 5 asterisks, a space, a title, a space, and 5 more
// asterisks
void printTitle (string title);

// printArray prints each element of an array right justified in a
// field of 3 followed by a newline
void printArray (int arry[], int size);

// sumOfArray returns the sum of all elements in the array
int sumOfArray (int arry[], int size);

// sumTwoArrays sums the values at the same index in firstArray and secondArray
// and places it in the same index in thirdArray; firstArray and secondArray
// must be the same size
void sumTwoArrays (int firstArray[], int sizeFirst,
                  int secondArray[], int sizeSecond,
                  int thirdArray[], int &sizeThird);

// isSorted returns true if the array is in order from smallest to largest;
// otherwise, false is returned
bool isSorted (int arry[], int size);

// insertLastArray adds value at the end of the array if there is
// space available
void insertLastArray (int arryDest[], int &size, int max, int value);

// insertFirstArray adds a value at the beginning of the array if there is
// space available. Each element of the array is shifted down by 1 to make
// room for the new value
void insertFirstArray (int arryDest[], int &size, int max, int value);

// copyArray copies each element from arrySrc to arryDest, overwriting any
// existing values. This function MUST use the function insertLastArray in your
// solution
void copyArray (int arryDest[], int &sizeDest,
               int arrySrc[], int sizeSrc, int max);

// reverseArray returns the array with elements reversed ... there is a cool
// way to do this and a not so cool way!!!
void reverseArray (int arry[], int size);
```

Expected Output

***** Printing Array A *****

1 2 3 4 5

***** Printing Array B *****

10 9 8 7 6

***** Sum Of Array A *****

15

***** Sum Of Array B *****

40

***** Print arrayAB which is the sum of arrayA and ArrayB *****

11 11 11 11 11

***** Check Array A is sorted *****

Array A is sorted

***** Check Array B is sorted *****

Array B is NOT sorted

***** Print arrayA after adding 6 at the end *****

1 2 3 4 5 6

***** Print arrayA after adding 6 at the beginning *****

6 1 2 3 4 5 6

***** Print arrayC after copying arrayA to arrayC *****

6 1 2 3 4 5 6

***** Printing Array A after reversing it *****

6 5 4 3 2 1 6

***** Printing Array B after reversing it *****

6 7 8 9 10

***** Print arrayB after adding 11 at the end *****

6 7 8 9 10 11

***** Print arrayB after adding 6 at the end *****

6 7 8 9 10 11 6

***** Printing Array A *****

6 5 4 3 2 1 6

***** Printing Array B *****

6 7 8 9 10 11 6

Press any key to continue . . .

Show the instructor or TA your solution

- 1) Your programs are to compile without any errors or warnings.
- 2) Do not use any magic constants in your program. Define your constants before defining the rest of your program's variables.
- 3) All functions should include comments, including main.

Once your projects are complete, place your solution into the CS150-01 Drop folder on grace. Your solution is to have ALL previous projects completely working and correct.