

CSCD 240

Lab 11

In Java `array = fillArray(kb, length);` is possible; yet in C we can't write that code using a statically sized array. In class we talked about dynamically allocating arrays. Let's write a small program to reinforce those concepts.

I have provided `cscd240Lab11.c`. In this main are eight (8) functions that you have to write. The specification for those functions is below.

- `readInitialLength` – reads the initial length of the array from the user; guarantees the length is one (1) or greater.
- `createAndFill` – takes the value from `readInitialLength` and creates the array. It then fills the array with integers entered by the user. The newly created array is returned.
- `menu` – valid choices are
 - `printSortedArray` – takes the array and the number of elements in the array – this function prints the elements in sorted ascending order.
 - `addItem` – takes the original array and the length of the original array (by reference) – makes a new array, copies over the old elements, and adds a new number (entered by the user) into the array. The new array is returned and the length is updated to represent the new length. You must `cleanUp` the old array in this function.
 - `readValue` – reads a value from the user to find in this array – this value is returned from the function
 - `printIfFound` – takes the value to find, the array, and the length of the array. Prints if the element was found in the array (first occurrence), and what location it was found at. If the item is not found – the appropriate error message is displayed.
 - `Quit`
- `cleanup` – returns any dynamically allocated memory back to the system.

Specifics

- You must place your prototypes above main and your functions below main
- You may NOT change my main in any fashion
- There is no reason for static arrays in the lab don't use them
- `cscd240Lab11.c` will contains my `main()` you may not change anything in this file
- You MAY NOT use static arrays
- You must use `stdin` - if you use a `FILE` * you will receive 0 points
- All input scores will be integer values

To Turn In

A zip file containing:

- cscd240Lab11.c which is necessary to compile and grade your code
- An output capture named cscd240Lab11out.txt
- A valgrind capture named cscd240Lab11val.txt to show you are leak free
- A Makefile with a target of lab11 - provided

Your zip will be named your last name first letter of your first name lab11.zip (Example: steinerslab11.zip)

My valgrind output

```
==4396==
==4396== HEAP SUMMARY:
==4396==    in use at exit: 0 bytes in 0 blocks
==4396==   total heap usage: 3 allocs, 3 frees, 48 bytes allocated
==4396==
==4396== All heap blocks were freed -- no leaks are possible
==4396==
==4396== For counts of detected and suppressed errors, rerun with: -v
==4396== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)
```