CNIT 31500 – Lab 3

Purdue University

Fall 2023

September 25, 2023 version 2. Subsequent changes will be marked in blue.

1 Goals

The primary goal of this project is to write, test and complete a c program which uses arrays, strings, and pointers. It is critical that you know how to program and use pointers. Some of the solutions you may be able to find on the web. If you decide to use those, remember to cite your sources, and comment each line of code, including the meaning of each variable. The secondary goal is to understand how fast the algorithms are, especially when they are applied to the same tasks.

2 Specifications

2.1 Arrays

In this part of the lab you will ask a user for input for an array of integers.

- Input:
  - You may ask for a size of an array first, and then for the numbers. In this case you will need to verify that the numbers specified match the size of the array.
  - You may also ask for a list of numbers separated by a space or a comma, compute how many numbers are in the list, and insert them into the list.
- The size of your declared array may be larger than your number of elements that you will insert.
- You will then use **Bubble sort**, **Selection sort**, and **Quick sort** to sort the array in **decreasing** order.
- Several of these Sort functions swap elements. Write a function that swaps the elements. Feel free to use the functions as described in Chapter 7. Swap does not need to work by checkoff.

- Keep track of the number of times the elements are switched in the array for each of the sorts and display this information after all sorts are completed.
- Analyze the performance.

2.2 Strings

You will also be asking a user for a string of words.

- Your task is to reverse the string using both pass by value and pass by reference functions.
- Do not use built-in functions for it

!!! Your functions should be recursive.

2.3 Repeat if necessary

As usual, the user should be able to repeat the experiments as many times as they choose, feel free to use logic similar to the previous labs.

3 Submission Guidelines

Submit your code on Gradescope, NOT Brightspace. Do not indicate your name or username in the comments to allow for anonymous grading.

Submit your code AND the Sort analysis. Sort analysis can be submitted as a separate file or included in the comments.

3.1 Checkoff requirements

Bubble Sort should be checked off based on the specifications for array input and the number of the moves (swaps) in the sort function. Checkoff of this component must be completed **in lab** by (week of September 25th).

4 Sorting the Arrays in Decreasing Order

Ask a user to input an array. The array should then be inputted into three functions: Bubble Sort, Selection Sort, Quick Sort. Make sure that the results of these functions do not overwrite each other.

Bubble, Selection, and Quick Sort are standard techniques to sort a list of data. You will be able to find pseudo code and implementation on the web and in your book. If you use pseudo code, you do not need to comment every line, just reference where pseudo code comes from. You do need to comment every line even if the code is in your book.

Remember that you are asked to sort the array in decreasing order – make necessary changes to the code you find.

Each time a use inputs an array, call all Sort functions. For each of the Sort functions, keep track of the numbers of moves it takes to sort the array.

Analyze your results, in terms of the number of moves, and based on the data that you received for:

- a completely sorted array
- an almost sorted array
- about half-sorted array
- completely unsorted array (you can try reverse order)

You do not need to program the analysis, writing it as text is sufficient.

5 String Reversal

Use recursion to reverse a string, by changing the order of its characters. You must use direct recursion for this and not any canned functions! Write one function where you use pass by value to make the swap, and one function where you use pass by reference. You may keep a copy of your original string, if it helps.

Check if the string is a palindrome. You should write a separate function for it, which returns true or false. Do not duplicate the code from the string reversal, call the result of one of the functions instead.

Print the original string, the reversed one, and whether the string is a palindrome.