

CS1XC3 - Computer Science Practice and Experience: Development Basics

Assignment #4 – Dynamic memory allocation

Weight

7% of total course grade

Due date

Sunday February 28th at 11:59pm

Primary Learning Objectives

- Write a C program that uses dynamic memory allocation.

Requirements

Create a program that will dynamically allocate space for an array of doubles according to the user's requested size, fill that array with values from user input, and then sort the array with insertion sort.

The program should ask the user to enter the number of elements. You can assume that the user will enter a positive integer value and do not need to perform input validation.

The program should dynamically allocate space for an array of doubles that can store this many elements. The program should then ask the user to input a double value for each element one at a time, "Enter element 0", "Enter element 1", etc. Each double value that the user inputs should be stored in the array in the order they are entered. Again you can assume the user will enter valid double values and you do not need to perform input validation.

After the user has finished entering the required number of values, print out the unsorted array (with 2 decimal digits of precision for each element).

The array should then be sorted from the highest to lowest using the **insertion sort** algorithm. Implement the insertion sort algorithm in C. You can find a pseudocode version of it here: https://en.wikipedia.org/wiki/Insertion_sort. Translating it to C can be done in about 10 lines of code.

After the array has been sorted, print out the sorted array (again with 2 decimal digits of precision). Remember to free the dynamically allocated memory.

Your program should function identically or near identically to this sample program.

```
Enter the numbers of elements: 8
Enter element 0: 9.4
Enter element 1: 4.5
Enter element 2: 1.5
Enter element 3: 9.5
Enter element 4: 2.3
Enter element 5: 4.5
Enter element 6: 6.7
Enter element 7: 5.7
```

Unsorted array:

```
9.40 4.50 1.50 9.50 2.30 4.50 6.70 5.70
```

Sorted array:

```
9.50 9.40 6.70 5.70 4.50 4.50 2.30 1.50
```

Submission

Save your solution in a file named **sort.c** and put it in a zip file named **a4.zip** and submit it to the Assignment #4 dropbox on Avenue to Learn.

Marking scheme

Component	Description	Marks
Dynamic memory allocation	Is the array dynamically allocated? Is the memory freed?	40
Insertion sort	Insertion sort implementation, highest to lowest sorting	20
User input	User input for the array size, element values	10
Array output	Is the unsorted and sorted version of the array output?	20
Comments	Are there sensible comments documenting the code?	10
Total:		100