

Lab Exercise 4 – Dynamic allocation and sorting

June 1, 2010

Sorting an array of integers

Write an ANSI-C program that

- reads the value of integer n from the standard input,
- reads n integers from the standard input into an integer array (each in new line),
- sorts the integers in the array in the value-ascending order,
- prints the sorted integers in the value-ascending order to the standard output.

Your program should prompt the user to enter the value of n at the beginning of the program. The user prompt should be “**Please enter the number of values:** ”. After reading a value for n from the standard input, your program should prompt the user again to enter n integers. This user prompt should be “**Please enter v integers:** ”, where v is the value the user has inputted for n .

Your program should output “**Sorted integers:** ” before printing the sorted values. The sorted values are separated by a **single space** and end with a newline character.

You can assume the following:

- The user input for n is a valid integer greater than 1 and no larger than 10 and ends with a newline character. You can use `scanf` to read it without checking its validity.
- The user inputs for the n integers (to be sorted) are valid integers separated by newlines. You can use `scanf` to read them without checking their validity.

You can use any sorting algorithm to sort the integers. Below is how the *selection sort* algorithm works:

- Find the minimum value in the array
- Swap it with the value in the first position
- Repeat the steps above for the remainder of the array (starting at the second position and advancing each time)

Sample run of the program is:

```

Please enter the number of values: 7
Please enter 7 integers: 97
-83
835
0
-983
97
16
Sorted integers: -983 -83 0 16 97 97 835

```