CS240 Programming In C

Homework 4

Pointers

100 pts

Due Apr. 16 11:59 pm

(50 pts.) Please write a program where you will count the number of vowels and consonants in a string. Note: you must use dynamic memory allocation. An example is shown below:

```
Pointer : Count the number of vowels and consonants :
-------------------------------------------------------
Input the length of your string: 12
Input a string: Hello world!
Number of vowels : 3
Number of consonants : 7
```

Please note that the white space and exclamation point are not counted as either vowels or consonants.

(50 pts.) Please write a program where you will sort an array of numbers. You will prompt the user to make a choice, 1. Ascending sort or 2. Descending sort. Based on the user choice, your program will sort the values accordingly. A screenshot is shown on the next page. **Please note we are implementing bubble sort, other sorting algorithms will not be accepted.** Your code must have the following specifications:

- Your code must ask the user for the number of elements ('n') they want to input.
- Your code must prompt the user to input 'n' number of integers for the array
    - For example, if the user wants to have an array with 6 elements, your code should ask the user to input 6 values for those elements.
    - If the user wants to have an array with 10 elements, your code should ask the user to input 10 values for those elements.
- Your code must then prompt the user to input an integer value indicating whether to sort in ascending or desending
- You must use a pointer (int *arr) to point to int which will be used to point to the array elements.
- You must use dynamically allocate memory using malloc and free.
    - To use malloc, you must include the library <stdlib.h> as a preprocessor.
    - You must include error handling for malloc.
        - For example (if NULL) printf("Memory allocation failed") exit(1)
- Finally, your code will sort these numbers and return a sorted array at the end. An example output is shown on the next page.

General Tips:

- Create a for loop to input the elements into the array.
- Create a nested loop. Your outer loop is to control how many iterations you need

  inner loop (j) should start at the next element and go to the end.

  - For every pair (i,j) if i is greater than j, swap them using pointer arithmetic.

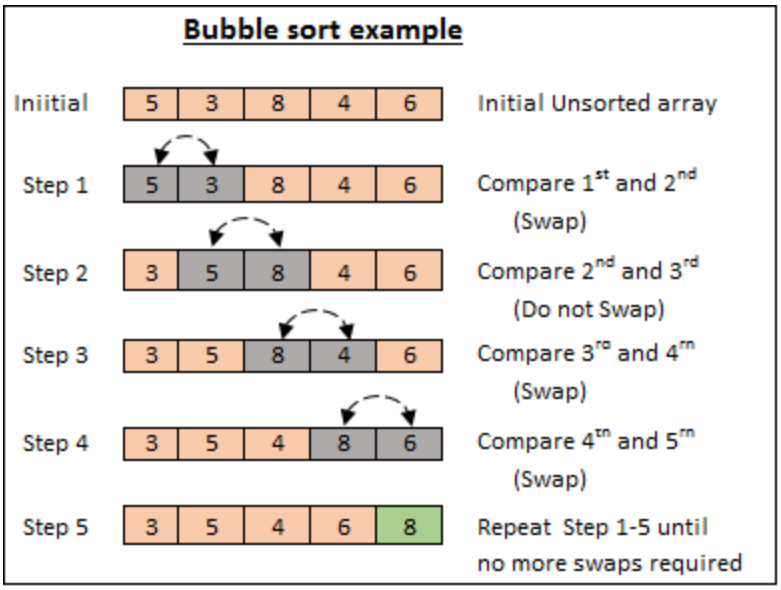
- To dynamically allocate memory we need to use malloc
  - C Reference function malloc() » CodingUnit Programming Tutorials
  - Lecture_9.pdf
  - When we use malloc we must also use free at the end to “free” the memory
- Use another for loop to print the sorted elements.

```
 Please input the number of values in your array : 5
Input your array values, you should input 5 values :
 element - 1 : 3
 element - 2 : 2
 element - 3 : 4
 element - 4 : 6
 element - 5 : 1
Choose sorting order:
1. Ascending
2. Descending
1

 The elements in the array after sorting :
 element - 1 : 1
 element - 2 : 2
 element - 3 : 3
 element - 4 : 4
 element - 5 : 6
```

```
 Please input the number of values in your array : 5
Input your array values, you should input 5 values :
 element - 1 : 3
 element - 2 : 2
 element - 3 : 4
 element - 4 : 6
 element - 5 : 1
Choose sorting order:
1. Ascending
2. Descending
2

 The elements in the array after sorting :
 element - 1 : 6
 element - 2 : 4
 element - 3 : 3
 element - 4 : 2
 element - 5 : 1
```



Please thoroughly test your code for errors and bugs before submission. You must include a makefile with your submission. If your code does not compile you will incur a **50-point penalty**.

Please submit your code to the CS Unix Servers. You should not include any executable files. Your submission should only include your C file, makefile, and readme. If you do not follow these guidelines, you will automatically incur a 50-point penalty.

- Q1.c, Q2.c
- Readme
- Makefile