CSCI3240 Lab 3: Arrays & Strings

Objective: To understand the use of arrays and strings for handling multiple data items of the same type.

Check the following links before you start:

- https://www.tutorialspoint.com/cprogramming/c_arrays.htm
- https://www.tutorialspoint.com/cprogramming/c_multi_dimensional_arrays.htm
- https://www.tutorialspoint.com/cprogramming/c_strings.htm

1. Write a program "***lab3Problem1.c***" to read the numbers in an array and reverse the order of the elements. Example: If array, **`arr = [1,2,3,4,5],`** after reversing it, the array should be, **`arr = [5,4,3,2,1].`**

   *The base code for **`lab3Problem1.c`** is provided alongside the lab3 document in D2L. Please add your logic to reverse the array in the TODO section.*

   Input Format:
   - The first line contains an integer, n, denoting the size of the array.
   - The next line contains n space-separated integers denoting the elements of the array.
   - The code to input the value of n and the element of the array is provided in "lab3Problem1.c" file.

   Constraints:

   - $\mathbf{1 \leq n \leq 500}$
   - $\mathbf{1 \leq arr_i \leq 1000}$, where arr_i is the i^{th} element of the array.

   Output Format

   - The output is handled by the code provided in "lab3Problem1.c", which would print the array.

   Sample Run

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ gcc lab3Problem1.c -o lab3Problem1
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem1
5
9 8 2 14 6
6 14 2 8 9
```

2. Write a program "***lab3Problem2.c***" to read a line of text from the user. Your program should count the number of words. For each word, if the word starts with a vowel, capitalize the word and display it in ascending order.

Constraints:

- There can be at max 50 words in the sentence.
- Each word can be of at a max of 20 letters.
- Hint: you can use #define macro to define the limits.
    - Check: https://www.tutorialspoint.com/cprogramming/c_preprocessors.htm

Output Format

- Output format should be same as shown in the sample run below. Each word should be printed in separate lines.

Sample Run

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ gcc lab3Problem2.c -o lab3Problem2
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem2
Enter a line of text: A quick ivory unicorn jumps over the lazy elephant
Words starting with vowels:
A
ELEPHANT
IVORY
OVER
UNICORN
```

Hint:

1. *Use fgets function to read the text. The scanf function stops at whitespaces (space, tab, newline) by default.*
    a. *C library function - fgets() (tutorialspoint.com)*
2. *Use multidimensional array to hold each word of the input text. You can use* ***strtok*** *function to divide the sentence into multiple words : https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm*
3. *Use qsort function to sort the words: https://www.tutorialspoint.com/c_standard_library/c_function_qsort.htm*

Rubrics (Total 100 points):

Problem1 (50 points):

Criteria	Points
If the array is not reversed	-50
The source code is not named correctly (it should be lab3Problem1.c)	-10
No submission or source code is missing	-50

Problem2 (50 points):

Criteria	Points
Some words starting with vowels are missing	-15
Words are printed in the wrong order (not ascending)	-15
Words are not capitalized	-5
Words are not printed on different lines	-5
Miscellaneous print errors	-5
The program is not able to read any valid input of maximum length	-5
The source code is not named correctly (it should be lab3Problem2.c)	-10
No submission or source code is missing	-50

Steps to Create the Log File:

1. **Open your terminal and start the scripting process by typing:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ script Lab3_Log.txt
```

2. **List all the files in the current directory:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ ls
```

3. **Compile your problem 1:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ gcc lab3Problem1.c -o lab3Problem1
```

4. **Run your problem1:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem1
```

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem1
```

Note: You can test with multiple other input values here.

5. **Compile problem 2:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ gcc lab3Problem2.c -o lab3Problem2
```

6. **Run your problem 2:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem2
```

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ ./lab3Problem2
```

Note: You can test with multiple other input values here.

7. **Exit the scripting process to finish and save the log file:**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ exit
```

8. **Convert the log file from txt to pdf using the txt created from using script**

```
(base) jovyan@jupyter-asainju:~/3240/lab3$ wkhtmltopdf Lab3_Log.txt Lab3_Log.pdf
```

The '**`Lab3_Log.pdf`** ' file will be generated in your current directory. Make sure this file is included in your submission.

Submission Instructions:

Please upload the following files:

- lab3Problem1.c
- lab3Problem2.c
- Lab3_Log.pdf
- AI_Disclaimer.pdf

- **Submission Due:** Check Lab 3 Dropbox