SFWRENG 2XC3- Major lab 3 - Lab section L02

Rules for Major Labs:

1. For major labs, students must be taking part in the registered lab section and should not move between other lab sections.

2. All major labs are an open book; use of all resources printed, written, or from anywhere on the Internet is permitted, but using ChatGPT and any AI-based language models is not allowed.

3. You are not allowed to solicit help from other persons or use work of other persons for even a part of the solution or discuss the solution with other persons. Especially, you are not allowed to discuss the assignment problems with the fellow students. In simple terms, the entire solution must be your own work.

4. You can submit your work (even partial work) as many times as you wish at any time from the posting of the assignment to the assignment's deadline; all submissions are saved in the repository. For marking, the latest submission is used.

5. An integral part of the submission process of your work is for you to check the submission.

6. Files saved in your Avenue account (Dropbox) are considered submitted.

7. After the assignment's deadline, during lab session time, no submission via the email or via Avenue is possible.

8. If submission is not possible for whatever reason, you could submit it to the TA and CC me, my email address: yazdinea@mcmaster.ca

In this lab, there are two tasks and two deliverables: text files (C programs) **mprog.c** for the first task, and **tprog.c** for the second task. Creating a screenshot of the output of your codes is recommended in a Word or PDF file.

Task 1: C Program named mprog.c

Description of the C program mprog.c [9.5 marks (of which 4.5 marks are for isDecreasingSeq function)]

1. The program starts by showing the message: "Please input a sequence:".
1. The program captures the user's input into a character array. Assuming a maximum length of array 100.
2. This array is sent to the isDecreasingSeq() function. The function examines if the digits in the array form a decreasing sequence. A decreasing sequence is defined as one where each digit is less than its preceding digit. For instance, "987431" is decreasing.

• Ensure you know what a decreasing sequence means. For a sequence of digits, each digit should be less than the digit before it. For example, in "54321", 4 is less than 5, 3 is less than 4, and so on.

3. The isDecreasingSeq() function returns 1 for decreasing sequences and 0 otherwise.
4. The main program announces either "The sequence 'XXX' is a decreasing sequence." or "The sequence 'XXX' is not a decreasing sequence." based on the function's return, where 'XXX' is the user's input.
5. Compilation instruction: gcc -o mprog mprog.c.

Description of the Function isDecreasingSeq:

1. The function's signature: int **isDecreasingSeq(char x[])**.
2. As the function processes the character array x:

• It checks if **x[i]** is less than or equal to **x[i+1]**, and if true, returns 0.

3. If the function processes the entire array without returning 0, it verifies the sequence as decreasing and returns 1.

Sample Runs:

```
[yazdinea@moore ~/M3] ./mprog
Please input a sequence:876543
The sequence "876543" is a decreasing sequence.
[yazdinea@moore ~/M3] ./mprog
Please input a sequence:2468
The sequence "2468" is not a decreasing sequence.
[yazdinea@moore ~/M3] ./mprog
Please input a sequence:998765532
The sequence "998765532" is not a decreasing sequence.
[yazdinea@moore ~/M3]
```

Task 2. C program named **tprog.c**

Description of the C program tprog1.c [9.5 marks (of which 4.5 marks are for flipChars)]

1. The program displays the message "Please enter a text string."
2. Using the **scanf()** function, it reads the user's response into a char array. Assuming a maximum length of array 100.
2. It then displays a message saying "You entered the string: 'XXX'", where XXX is the text string entered by the user and stored in the array.
3. Afterward, the program passes the array to a function called **flipChars()**, which will modify the characters of the array (details below).
4. Finally, the program displays the modified word in a message: "String after flip operation: 'XXX'", where XXX is the modified text string stored in the array.
5. The program is compiled in the ordinary way using **gcc -o tprog1 tprog1.c**.

Description of the function flipChars

The function's prototype is **void flipChars(char x[])**. In a loop, the function traverses the array **x** and alters each character according to the following schema:

1. **Inverting Vowels:**
    - Lowercase vowels ('a', 'e', 'i', 'o', 'u') are converted to uppercase.
    - Uppercase vowels ('A', 'E', 'I', 'O', 'U') are converted to lowercase.
2. **Converting Consonants:**

- Lowercase consonants are converted to the next consonant in the alphabet with the following nuances:
    - 'h' is converted to 'j' (skipping the vowel 'i').
    - 's' is converted to 't'.
    - 'z' rolls over to 'b'.
- Uppercase consonants undergo similar transformations:
    - 'H' is converted to 'J'.
    - 'S' is converted to 'T'.
    - 'Z' rolls over to 'B'.

3. **Inverting Digits:**
    - Each digit is inverted such that 0 becomes 9, 1 becomes 8, ..., and 9 becomes 0.
4. **Special Characters:**
    - Any other characters are replaced by an exclamation mark ('!').

The function then terminates.

A Sample Run tprog1

For the input **hello123**:

1. h -> j (since 'h' is a consonant, it gets changed to the next consonant after skipping 'i', which is 'j')
2. e -> E (the vowel 'e' gets inverted to 'E')
3. l -> m (since 'l' is a consonant, it gets changed to the next consonant, which is 'm')
4. l -> m (again, 'l' becomes 'm')
5. o -> O (the vowel 'o' gets inverted to 'O')
6. 1 -> 8 (digit inversion: 9 - 1 = 8)
7. 2 -> 7 (digit inversion: 9 - 2 = 7)
8. 3 -> 6 (digit inversion: 9 - 3 = 6)

So, the transformed string becomes: **jEmmO876**

Thus, the output of the program is correct for the input **hello123**.

```
Please enter a text string.
hello123
You entered the string: 'hello123'
String after flip operation: 'jEmmO876'
[yazdinea@moore ~]
```