CS19101 Programming and Data Structures
Strings

General instruction to be followed strictly

1. Do not use any global variable unless you are explicitly instructed so.
2. Use proper indentation in your code and comment.
3. Name your file as `<roll_no>_<assignment_no>`. For example, if your roll number is 14CS10001 and you are submitting assignment 3, then name your file as `14CS10001_3.c` or `14CS10001_3.cpp` as applicable.
4. Write your name, roll number, and assignment number at the beginning of your program.
5. Make your program as efficient as possible.

Part-I

Submit one (single) C program.

Write a C program to perform the following tasks.

Definition of Hamming distance of two strings: Let there be 2 strings s_1, s_2 with respective lengths ℓ_1, ℓ_2 such that $\ell_1 \leqslant \ell_2$ (wlog). Assuming the strings start with index 0, let the number of indices $0 \leqslant i \leqslant \ell_2 - 1$ where the two strings differ be t. All indices between $\ell_1 - 1$ and $\ell_2 - 1$ will be considered to be diferring. Then the Hamming distance of the two strings is t. Eg: The two strings `abcd` and `bbdefgh` are of respective lengths 4 and 7 and among the indices between 0 and 6, they differ in indices $0, 2, 3, 4, 5, 6$. Therefore, the Hamming distance of the two strings is 6.

1. Define a structure *user* which stores 2 strings corresponding to the username and password of a user. Assume that a username or password can be of at most 100 characters.
2. In the main program, take as input a positive integer n.
3. Then take as input the details of n structures of type *user*. When taking the input for the i^{th} structure, make sure that the i^{th} username hs Hamming distance at least 1 from the previous $i-1$ usernames.
4. Ask a user to input their username and password.
   - ▷ If the username does not match with any of the n usernames, output a message asking the user to register this username, and ask for a username and password again. If the username does not match after 3 attempts, write an error message "You have used up all your attempts."

- ▷ If the username matches with one of the n usernames but the corresponding password s does not, then output an error message, and request for a new password.
  Let s' be the string entered as a new password. Now you have to make two checks with s and s':
  (a) If the s' has at least 1 vowel,
  (b) If the Hamming distance of s and s' is at most $\text{length}(s \cdot s')/3$.
  Until both the conditions are satisfied, you have to keep asking for a suitable new password. Finally, replace s in the structure with s' and output a message saying that the password has been reset.
- ▷ If both username and password match, then output a message that the login was successful.

You can write your own functions wherever necessary, but proper commenting is required to explain the purpose of the function. Functions from string.h can be used.

Part-II

Sample Output

```
Enter positive integer: 2
Enter the username of user 1: userone
Enter the password of user 1: pass
Enter the username of user 2: userone
This username is already taken. Enter the username of user 2: usertwo
Enter the password of user 2: word
Enter your username to login: user
Enter your password to login: word
This username is not registered!
Enter your username to login: userone
Enter your password to login: word
Your password does not match, please enter a new password: word
Thank you, your password has been reset.
```

Policy on Plagiarism

Academic integrity is expected of all the students. Ideally, you should work on the assignment/exam consulting only the material we share with you. You are required to properly mention/cite anything else you look at. Any student submitting plagiarised code will be penalised heavily. Repeated violators of our policy will be deregistered from the course. Read this to know what is plagiarism.