LAB 05: STRINGS

1 IN-CLASS

Complete the following requirements, each of which needs to be implemented by a function using C++ strings (i.e. `<string>`) or a function using C strings (i.e. include `<string.h>`).

1. Input a string and output the reverse of the input string. For example,
   - Input: Captain Marvel
   - Output: levraM niatpaC
2. Input a string that contains numbers only and output the spelling of the input string. For example,
   - Input: 1234
   - Output: Mot ngan hai tram ba muoi bon.
3. Input a string and output the following pieces of information
   - How many words are there?
   - Which word has the highest number of occurrences? How many times?
   - What are the number of occurrences of every characters in the string?
4. Input three strings, `str1`, `str2` and `str3`, find `str2` in `str1` and replace all occurrences by `str3`. For example,
   - `str1`: There are 5 flowers: red flower, yellow flower, blue flower, black flower and white flower.
   - `str2`: flower
   - `str3`: ranger
   - Output: There are 5 **ranger**s: red **ranger**, yellow **ranger**, blue **ranger**, black **ranger** and white **ranger**.
5. Input two strings, `str1` and `str2` (`str1` is longer than `str2`), and output the number of occurrences of `str2` in `str1`.

2 HOMEWORK

1. **Phone directory**:

   A contact in the phone directory contains: *{First name and Last name} and Phone number*. For instance:

   - *Tony Stark, 555-1234*
   - *Bruce Banner, 555-2222*
   - ...

   Write the functions for the following requirements

   (a) Inform the user to enter one contact ore more, then save them in a dynamic array, and display the contacts in the array to the screen

   (b) Sort the contacts in ascending order by **Name**

   (c) Inform the user to enter a *Last name* or *First name* and show the corresponding contact

2. **Password and Password Encoding**:

   (a) When a user signs up for an account, he needs to enter a password, which is basically a string, `str`. Write a function to check whether the password is strong enough. A strong password must meet the following criteria

   - Contain from 9 - 20 characters.
   - Have both lowercase and uppercase letters
   - Have at least one number
   - Have at least one special character (i.e., not lowercase/uppercase/number).

   (b) If the password is strong enough, it will be encrypted. Here, we are going to use a simple symmetric encryption by increasing the ÂSCII value of each character by `k`. Write a function that takes the integer `k` as the input and encrypts the password. Note: the ASCII value of the input characters is in range [21, 126].

3 PREPARING YOUR SUBMISSION

Create a new folder and name it with your **Student ID**, e.g. 19127001. This folder includes

- **Code**: a sub-folder that contains your source code (*.cpp, *.h, etc.). Do not forget to delete all intermediate files.
- **Report** (if required): a sub-folder that contains your written report (*.pdf).

For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.