Institute for Advancing Intelligence, TCG CREST
(TCG Centres for Research and Education in Science and Technology)

Introduction to Programming and Data Structures, 2023-24, Semester-II
Assignment 02

Maximum Marks: 150
Topic: Strings and I/O from files

Submission Deadline: **2023-Aug-31**
Clarification Deadline: **2023-Aug-28**

Here are five problems. You are to write C programs for the following problems. In each solution, you should take/give input/output only from/to a file only.

AP0201: Twin Palindromes: Tanmoy[1] bets Pritam. So, Tanmoy gives two positive integers a and b. Pritam has to find two distinct palindrome strings S_1 and S_2 consisting of only the characters '0' and '1' such that,

both S_1 and S_2 contain a occurrences of '0' and b occurrences of '1'.
If Pritam can not find any such combination, he will lose and should output the *Lost* string.

It should have the following functions.

- `char ** find_special_strings(int a, int b)`: returns array of two string pointers.
- `int show_special_strings(char ** arrayOfStrings)`: returns success indicators.
- `int write_special_strings(char ** arrayOfStrings, FIlE * dstFilePtr)`: returns success indicators.

Sample Input: from a file with filename "`input_0201.txt`"
3 //Number of test cases
4 5 //case 1
7 3 //case 2
2 6 //case 3
Output: to a file with filename "`output_0201.txt`"

AP0202: Romantic Reversals: Bishakha feels disturbed as Umme often asks many questions that ruins her coffee-break. To get ride of these, Bishakha plans something. She takes a string S of length N performs K steps on the string, numbered from 1 to K. In the i-th step, she reverses the first i-characters of the string. For example, if S = "abferty" and K = 3, then the string after the 3 steps will be "faberty". Bishakha gives the final string S' together with K and challenges Umme to find the original string S. Write a C program to help Umme win the challenge.

It should have the following functions.

- `int find_reversed_string(char * inpStr, int K)`: change *inpStr* and returns success indicator.
- `int find_original_string(char *reversedStr, int K)`: change *reversedStr* to its original returns success indicators.
- `int write_original_strings(char *InpStr)`: Write *inpStr* in output file and return success indicators. Display win/loss in the terminal.

Sample Input: from a file with filename "`input_0202.txt`"
3
vjbaadksl 5
ugyadkb 3
webwkela 6
Output: to a file with filename "`output_0202.txt`"

[1] Disclaimer: All Characters Are Fictitious

AP0203: Power Naps: Alamgir and Nikita occasionally have to work on the same project from their respective home on some holiday. They have to work on some project for the next N hours. At the beginning of each such day, their supervisor gives a work plan to do this, which is a binary string S of length N. $S[i] =$ '1', if Alam has to work on the project during the i-th hour, and $S[i] = 0$ if Alamgir is free during the i-th hour, during Alamgir's free time Nikita has to work and vise-verse. Alamgir would like to use some of his free time to take naps. He needs a minimum of K consecutive hours of free time to take a nap. What is the maximum number of naps that Alamgir can take during the next N hours for a project?

Write a C program to find answer.

Input: from a file with filename "`input_0203.txt`"

3 //Number of test cases
15 2 //case 1
001001000001001
17 3 //case 2
10001001000001001
21 4 //case 3
111000100100000100010

Output: In terminal, the number of naps Alamgir can take.

AP0104: Subham is planning to set up a secure password for his bank account. For a password to be secure, the following conditions should be satisfied:

- Password must contain at least one lower-case letter [a-z];
- Password must contain at least one upper-case letter [A-Z] strictly inside, i.e. not as the first or the last character;
- Password must contain at least one digit [0-9] strictly inside;
- Password must contain at least one special character from the set '@', '#', '%', '&', '?'
- The length of the password must be between 10 to 30.

Input: from a file with filename "`input_0204.txt`"

- First line contains T, the number of test cases.
- Each test case contains a single string S.

Output: In the file with filename "`output_0204.txt`"

For each test case, print "YES" if the password is strong, else print "NO", each in a new line.

[40+40+40+40]