Indian Institute of Information Technology Sri City
(An Institute of National Importance under An Act of Parliament)
Computer Programming
First Semester 2021-2022
Lab Assignment – 06

Name: CP Lab - 06	Duration: 3 Hrs
Date: 24 Jan, 2022	Maximum Marks: 20

INSTRUCTIONS:
1. Please carefully read all assignment problems and write the required programs in the C language.
2. All the PROBLEMS are COMPULSORY.
3. You should submit only a single C file containing all your answers. Make sure that
4. during submission, no part of your code is commented.
5. Name the file as follows: S2021xxxxx_A6.c
6. DO NOT zip. Upload a single .c file directly to your submission in the common Google classroom.
7. Don't share or copy the codes. If malpractice found, you will be awarded Zero.

*If you do not follow the above-mentioned instructions, a strict penalty would be imposed.

1. You are given two strings *t* and *p*, where the string *t* contains all the letters of string *p* in the same order (it doesn't have to be contiguous). For example, *t* could be 'abdefcb' and *p* could be 'abb' but *p* could not be 'bab' (order is not preserved) or 'bxy' ('xy' is not in *t*).

   You are also given a set of operations on an array *a*. Each element *a*[i] denotes the index of the letter to be removed from string *t* (a[i]>=1). For example, if the array, say **a**, is '3 5 4 7 2 1 6' applied on the string *t* which is 'abdefcb', then the operation is executed as follows:
   a. first remove the 3rd element 'd', so *t* becomes 'ab~~d~~efcb' (character 'd' has been striked through).
   b. Then, we remove the 5th element 'f', so *t* becomes 'ab~~def~~cb' (character 'f' has been striked through) and so on.

   Note that after removing one letter, the indices of other letters don't change. So, if a letter was at 5th position and you remove the 3rd letter, then still the letter at the 5th position stays at 5th position.

   QUESTION: Determine and print the maximum number of letters that can be removed from string *t* following the operations in the array so that the string *p* is obtainable from *t*. Note that array *a* has the same size as the string *t* and the elements in array *a* are distinct.

Input:
First line contains string *t*
Second line contains string *p*.
Third line contains an integer *n* which denotes the number of elements in array *a*.
Fourth line contains array *a*.

Output:
Print the maximum number of letters that you can remove following the order in array *a*.

Example :
abdefcb
abb
7
3 5 4 7 2 1 6

Output :
3

Explanation:
Following the order of removals in array *a*, you first remove 'd', then 'f', then 'e'. Now if you remove the 7th letter which is 'b', then abb is in no way obtainable. So, you stop after 3 operations and output is 3.

2. Write a C program to read a string **s** and **m** queries, where each query contains three integers *i, j, k*.
- The query is defined as a cyclically shifting the values from **s[i]** to **s[j]** for *k* times, where *i<j*.
- One operation of cyclic shift (rotation) is equivalent to moving the last character to the position of the first character and shifting all other characters one position to the right.
- Suppose index starts from 1. For example, a string **s** is give as 'abcdef' and the query is 2, 4, 1, then the values from **s[2]** to **s[4]** are 'bcd' and shifting it *k = 1* times results in 'dbc'. The final string is 'adbcde'.

Input format:
First line string **s**
Second line integer **n** - the number of queries.
Each of the next **n** lines contains *i, j, k* ($1 \leq i \leq j \leq |s|$ and $1 \leq k \leq |s|$).

Ouput:
A single string which is the result of all the operations.

Example:
abbcde
2
2, 4, 1
1, 3, 2
Output:
cbabde