IT251 Lab Assignment 8 – KMP, Rabin-Karp

Note:
1. For all the following problems, read in the input using a text file, and NOT by typing it in the console. The input file should be given as an argument while running your code. For e.g. for a file solution.cpp, compile it by 'g++ test.cpp' and run it by './a.out input.txt' , where 'input.txt' contains the input to the problem.
2. Submit a **single file** with your code for the two problems in this assignment. The problem to be run will be specified as an argument during runtime. For e.g. './a.out 2 inputfile.txt' should run the code to solve problem 2 with the input given in the file 'inputfile.txt'

The pattern matching problem we are trying to solve here is the following:

Input: A string **Pattern** and a collection of text **Texts** containing longer strings.
Ouput: All the starting positions in **Texts** where the string **Pattern** appears as a substring.

We will solve this problem using two algorithm: KMP algorithm and Rabin-Karp algorithm.

Problem 1: Knuth-Morris-Pratt (KMP) Algorithm

In this part we will implement the KMP algorithm. The string **Pattern** is pre-processed, and a prefix array π of the same size of **Pattern** is computed. The entries of array π will tell us how to shift the pattern in case of a mismatch with the text.

Input: The first line of the input contains the string **Pattern.** The second line contains an integer n and the following n lines contain the collection of strings in **Texts.** Each of the text strings does not exceed a single line, so these n lines contain the n text strings on which the occurrence of **Pattern** will be searched.

Output: All starting positions (in increasing order) in **Texts** where the string **Pattern** appears as a substring.

Constraints: 1 ≤ |**T**| ≤ 10000 for all strings T in **Texts**; 1 ≤ n ≤ 5000; 1 ≤ |**Pattern**| ≤ 100**;** all strings contain only symbols A, C, G, T;

Sample Input/Output:
Input:
ATA
3
ATAGATACA
ATC
GATA

Output:
0 4
Pattern Not Found
1

Explanation: Pattern ATA occurs in positions 0 and 4 in the first text (ATAGATACA); It does not occur in the second text (ATC); it is found in position 1 in the third text (GATA)

Problem 2: Rabin-Karp's algorithm
Implement Rabin Karp's algorithm to solve the same pattern matching problem. The input and output specifications is the same as in Problem 1.