#Problem 1: Basic Prime Checking

Description:
In this problem, first you will be given the number of test cases T. Then you will be given T numbers. For each number t_i (1<=i<=T), you need to find if it's a prime number or not. Look at the input output section for more clarification.

Limits
1<=T<=100
1<=t_i<=10^9

Test Cases:

Input	Output
3 2 3 12	PRIME PRIME NOT PRIME
7 10007000 1111113 3111391 100000003 100099897 300099973 244499972	NOT PRIME NOT PRIME NOT PRIME NOT PRIME NOT PRIME PRIME NOT PRIME

#Problem 2: Finding the Pattern's occurrences

Description:
In the first line, you will be given a text T and in the following line you will be given a pattern P. You need to find the occurrences of P in T. Print each occurrence (start and end index in T) in separate lines keeping the ascending order of the starting indexes. Look at the input output section for more clarification. The output will follow 0 based indexing.

Limits
P and T both will always contain only English upper case letters, 'A' <= P_i, T_i <='Z'.
1<=|P|<=1000
1<=|T|<=10000000

Test Cases:

Input	Output
ABCABCDABCADDD BCA	1 3 8 10
AABABABCABABABABBABAB ABAB	1 4 3 6 8 11 10 13 12 15 17 20
DDACBDBBBBBCBABBBCACBDDBBDBBBC BBB	6 8 7 9 8 10 14 16 26 28
CAABCBBBBCBBCCACCACAAAABBBCCBB CAABC	0 4

#Problem 3: State Transition Matrix

Description:
You will be given a text T and a pattern P . You need to calculate the transition matrix constructed from P which can be used to find its occurrence in T.

The output will contain |P|+1 lines, where |P| denotes the length of P. Each line will contain K values where K denotes the total number of unique characters found between P and T. In each line the transitions will be printed maintaining the ascending order (ASCII Value) of the characters, e.g,, first for ‘A’, then ‘B’, then ‘C’, etc.

Limits
P and T both will always contain only English upper case letters, ‘A’ <= P_i, T_i <=’Z’.
1<=|P|<=1000
1<=|T|<=10000000

Test Cases:

Input	Output
CABABABBCBACBCCABABBBBCCABABCA ABAB	1 0 0 1 2 0 3 0 0 1 4 0 3 0 0

ABABBABBAA ABBA	1 0 1 2 1 3 4 0 1 2
CAABCBBBBCBBCCACCACAAAABBBCCBB CAABC	0 0 1 2 0 1 3 0 1 0 4 1 0 0 5 2 0 1
DDACBDBBBBBCBABBBCACBDDBBDBBBC BBB	0 1 0 0 0 2 0 0 0 3 0 0 0 3 0 0