

## #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  $t_i$  ( $1 \leq i \leq T$ ), you need to find if it's a prime number or not. Look at the input output section for more clarification.

### Limits

$1 \leq T \leq 100$

$1 \leq t_i \leq 10^9$

### 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, ' $A' \leq P_i, T_i \leq 'Z'$ '.

$1 \leq |P| \leq 1000$

$1 \leq |T| \leq 10000000$

### Test Cases:

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

### #Problem 3: Calculate Prefix Function of a String

Description:

In this problem, you will be given a string P. You need to calculate the prefix function's value of P. For each length prefix  $P_i$  (1 length, 2 length, 3 length, etc. ) of P, prefix function calculates the length of the maximum prefix that matches with the suffix of  $P_i$ .

Let the length of the given string be m. Then the output will have a single line containing m integer values separated with a single space denoting the length of the maximum prefix that matches with the suffix for each length prefix of  $P_i$ .

Limits

P will always contain only English upper case letters, 'A' <=  $P_i$ ,  $T_i$  <= 'Z'.

$1 \leq |P| \leq 100000$

Test Cases:

Input	Output
ABABABABCA	0 0 1 2 3 4 5 6 0 1
ABABABABABABABABCA	0 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 0 1
AAACBBBCCAACAABBABCCACAAACBC CC	0 1 2 0 0 0 0 0 0 1 2 0 1 2 0 0 1 0 0 0 1 0 1 2 3 4 5 0 0 0