Find Substring

Problem Description

Given 2 strings, determine whether a particular substring is inside those strings:

- Output 0 if the substring does not belong to both strings.
- Output 1 if the substring is in first string but not second string.
- Output 2 if the substring is in second string but not first string.
- Output 3 if the substring is in both strings.

Input

The first line of the input contains **N** (1 <= **N** <= 500) and **K** (1 <= **K** <= 6), where **N** is the length of those strings and **K** is the length of the substring that we are interested in. The second line contains the first string of length **N**. The third line contains the second string of length **N**. The next line contains an integer **Q** (1 <= **Q** <= 1,000), denoting the number of substrings that we are interested in. The next **Q** lines contain the substrings of length **K**.

Output

There are **Q** lines in the output. Each line contains the number of occurrences of the substrings.

Sample Input	Sample Output
6 2 ACGTAC ACTGCA 6 AC CG AT GC TA CA	3 1 0 2 1 2

Explanation

```
Find AC: ACGTAC      ACTGCA
Find CG: ACGTAC      ACTGCA
Find AT: ACGTAC      ACTGCA
Find GC: ACGTAC      ACTGCA
Find TA: ACGTAC      ACTGCA
Find CA: ACGTAC      ACTGCA
```

Marking

1. You will only gain a maximum of 70% for this section if you solution is worse than big-Oh of O(N*K + Q*K).
2. You can gain a maximum of 100% for this section if a O(N*K + Q*K) or better solution is implemented correctly. You must answer each query in O(K) time.

Hint

- To get 100% solution, think of a data structure that you have learnt that can search/query a particular key in O(1). Also note that the valid characters are only ‘A’, ‘G’, ‘C’, and ‘T’.
- **K** is at most 6.

Note

The main Java class must be called **Find**, and be in the source file **Find.java**.