Retrovirus

Little Petar, hacker and bioinformatician, has successfully achieved his greatest hacking feat to date - he successfully broken into the highly guarded database of the DBW (Department of Biotechnological Weaponry), which is suspected of creating many pandemic-causing retroviridae in the past. Among many secret projects, he has discovered a prototype of a brand-new kind of retrovirus, potentially capable of causing a new, incurable disease.

Retroviridae store all their genetic information used for attacking the host cell in a single strand of ribonucleic acid (RNA). An RNA strand consists of a string of nucleotides, that may be either Adenine (A), Uracil (U), Cytosine (C) or Guanine (G).

Petar found out that the prototype is actually a mutated version of a known retrovirus - as such, it is expected that there are regions (substrings) where these two viruses are highly "similar". The similarity of two regions of same length is defined as the number of indices where their nucleotides match; e.g. the similarity between "ACAGU" and "AGAGA" is 3 (the nucleotides match on the first, third and fourth position).

Petar has identified potentially similar regions - it is now up to you to write the program to calculate how similar they actually are.

Input

The first line of the standard input contains a natural number N, the length of the RNA strand of both the old and the new retrovirus. The second and third line contain two strings, RV_1 and RV_2, representing their RNA strands.

The fourth line contains a natural number Q, representing the number of similarity queries that need answering. Each of the following Q lines contains three natural numbers X, Y and L - this means that the similarity of the regions of length L starting from the Xth position in RV_1 and the Yth position in RV_2 should be calculated. It is guaranteed that this region can extend for at least L characters in both strings.

Output

For each of the Q queries, output the answers in order, each one in a separate line of the output.

Example

Input:

```
7
AUGCAAG
GGAUGCG
2
1 3 4
6 1 2
```

Output:

```
4
1
```

Explanation

Let A[i..j] be the substring of A starting on index i and ending on index j.

The first query asks for the similarity of RV_1[1..4] = "AUGC" and RV_2[3..6] = "AUGC", which is 4 (as this is a complete match).

The second query asks for the similarity of
RV_1[6..7] = "AG" and RV_2[1..2] = "GG", which is 1 (as only the second index matches).

Constraints

- 1 <= N <= 1000
- 1 <= Q <= 10^6