Coding Competitions Farewell Rounds - Round D

Genetic Sequences

Genetic Sequences

Margaret researches genetic sequences. She is analysing two sequences $\mathbf{A}$ and $\mathbf{B}$ from a new kind of life that does not use the typical four letter genetic alphabet. The code for the genetic sequences conveniently requires 26 letters represented by the uppercase English letters `'A'` through `'Z'`.

Margaret wants to compare the sequences $\mathbf{A}$ and $\mathbf{B}$. The best way to do this is to do a series of sequence analysis tests. Each test involves taking a prefix from $\mathbf{A}$ containing only the first $\mathbf{P}$ letters from $\mathbf{A}$, which is called the $\mathbf{A}$-prefix. Each test also involes taking a suffix from $\mathbf{B}$ containing only the last $\mathbf{S}$ letters from $\mathbf{B}$, which is called the $\mathbf{B}$-suffix. Margaret then needs to compare the $\mathbf{A}$-prefix to the $\mathbf{B}$-suffix. A substring is a subsequence of contiguous letters. A substring from the $\mathbf{A}$-prefix matches the $\mathbf{B}$-suffix if the $\mathbf{B}$-suffix starts with that substring. That is, the substring is a prefix of the $\mathbf{B}$-suffix. The result of a test is the length of the longest substring from the $\mathbf{A}$-prefix that matches the $\mathbf{B}$-suffix.

Margaret needs some software to determine the outcome of a batch of $\mathbf{Q}$ sequence analysis tests. Note that each test is independent. Margaret has many copies of $\mathbf{A}$ and $\mathbf{B}$ and a new one is used for each test.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow. Each test case begins with a line containing two strings and an integer, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{Q}$ respectively. Each test case ends with $\mathbf{Q}$ lines, the i-th of which contains two integers $\mathbf{P_i}$ and $\mathbf{S_i}$, which are the prefix and suffix sizes for the i-th sequence analysis test.

Output

For each test case, output one line containing `Case #`x`:` y_1 y_2 $\dots$ $y_\mathbf{Q}$, where x is the test case number (starting from 1) and y_i is the answer to the i-th query in the input.

Limits

Time limit: 20 seconds.
Memory limit: 2 GB.
$1 \le \mathbf{T} \le 100$.
$1 \le \mathbf{P_i} \le$ the length of $\mathbf{A}$.
$1 \le \mathbf{S_i} \le$ the length of $\mathbf{B}$.

Test Set 1 (Visible Verdict)

$1 \le$ the length of $\mathbf{A} \le 3000$.
$1 \le$ the length of $\mathbf{B} \le 3000$.
$1 \le \mathbf{Q} \le 3000$.

Test Set 2 (Hidden Verdict)

$1 \leq$ the length of $\mathbf{A} \leq 10^5$.
$1 \leq$ the length of $\mathbf{B} \leq 10^5$.
The sum of the lengths of $\mathbf{A}$ over all test cases is $\leq 5 \times 10^5$
The sum of the lengths of $\mathbf{B}$ over all test cases is $\leq 5 \times 10^5$
$1 \leq \mathbf{Q} \leq 10^5$.

Sample

Sample Input

```
3
ABCABAC CABABA 3
3 6
7 6
6 5
BANANA HABANA 2
5 4
5 5
ABC ABD 1
2 1
```

Sample Output

```
Case #1: 1 4 3
Case #2: 4 1
Case #3: 0
```

In Sample Case #1, there are 3 tests. The prefix `ABC` from $\mathbf{A}$ and the complete suffix `CABABA` from $\mathbf{B}$ are compared in the first test. The answer is 1, since `C` is the longest substring that is contained in `ABC` and is a prefix of `CABABA`. In the second test, `ABCABAC` is tested against `CABABA` and the longest match is `CABA`. In the third test, `ABCABA` is tested against `ABABA` and the longest match is `ABA`.

In Sample Case #2, there are 2 tests. In the first, `BANAN` is tested against `BANA`, and the longest match is `BANA`. In the second, `BANAN` is tested against `ABANA`, and the longest match is `A`.

In Sample Case #3, there is one test. In it, `AB` is tested against `D`. Since there is no match the answer is 0.