E. Hellish Constraints

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Katya recently started to invent programming tasks and prepare her own contests. What she does not like is boring and simple constraints. Katya is fed up with all those "N does not exceed a thousand" and "the sum of a_i does not exceed a million" and she decided to come up with something a little more complicated.

The last problem written by Katya deals with strings. The input is a string of small Latin letters. To make the statement longer and strike terror into the people who will solve the contest, Katya came up with the following set of k restrictions of the same type (characters in restrictions can be repeated and some restrictions may contradict each other):

- The number of characters c_1 in a string is not less than l_1 and not more than r_1.
- ...
- The number of characters c_i in a string is not less than l_i and not more than r_i.
- ...
- The number of characters c_k in a string is not less than l_k and not more than r_k.

However, having decided that it is too simple and obvious, Katya added the following condition: a string meets no less than L and not more than R constraints from the above given list.

Katya does not like to compose difficult and mean tests, so she just took a big string s and wants to add to the tests all its substrings that meet the constraints. However, Katya got lost in her conditions and asked you to count the number of substrings of the string s that meet the conditions (each occurrence of the substring is counted separately).

Input

The first line contains a non-empty string s, consisting of small Latin letters. The length of the string s does not exceed 10^5.

The second line contains three space-separated integers k, L and R $(0 \le L \le R \le k \le 500)$.

Next k lines contain Katya's constrictions in the following form "c_i l_i r_i". All letters c_i are small Latin letters, l_i and r_i are integers $(0 \le l_i \le r_i \le |s|$, where $|s|$ is the length of string s). Letters c_i are not necessarily different.

Output

Print a single number — the number of substrings that meet the constrictions.

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cout` stream or the `%I64d` specificator.

Examples

input

```
codeforces
2 0 0
o 1 2
e 1 2
```

output

```
7
```

input

```
codeforces
2 1 1
```

```
o 1 2
o 1 2
```

output

```
0
```

Note

In the first test we should count the number of strings that do not contain characters "e" and "o". All such strings are as follows (in the order of occurrence in the initial string from the left to the right): "c", "d'", "f", "r", "rc", "c", "s".

In the second test we cannot achieve fulfilling exactly one of the two identical constrictions, so the answer is 0.