VM7WC '16 #2 Silver - GG

Time limit: 0.1s **Memory limit:** 64M
Java: 0.5s
Python: 0.5s

G's are delivered through G-strings, which are made mostly of G's, and are not to be confused with the article of clothing. Sometimes, G's can be contaminated with other alphabetical characters—from old age or memory issues—and the integrity of the G-string becomes compromised.

Here is a prime G-string, made only of G's:

```
GGGGGGGGGGGGGGGGGG
```

Here is a contaminated G-string, with some other characters among the G's:

```
SKGCGUSGGGGOGGELGG
```

G-strings may be contaminated to the point that they contain no G's at all!

G-strings are only as good as the longest run of G's in the G-strings. Define the integrity of a G-string as the number of G's that are present in the string. For example, the integrity of the first string, `GGGGGGGGGGGGGGGGGG`, is 18. The integrity of the second string, `SKGCGUSGGGGOGGELGGG`, is 11.

We can find the integrity of a substring of a G-string as well. Using zero-based indexing, the substring from positions 0 to 3, inclusive, of the second given G-string is:

```
SKGC
```

The integrity of this substring is 1, as there is only one G in the entire substring. Create a program that, when given a G-string and a list of ranges queries, can determine the integrity of the substring within each range.

Input Specification

The input is a single line, containing the G-string to be examined. The length of the G-string is at most 10^5 characters long and is comprised of only capital letter alphabetical characters.

On the second line is a single positive integer Q ($1 \leq Q \leq 10^5$), the number of queries the program must process. The next Q lines each contain two non-negative integers i and j ($0 \leq i \leq j < N$), where N is the length of the given G-string.

Output Specification

For each query and on separate lines, print the integrity of the substring between positions i and j in the G-string, inclusive, and using zero-based indexing.

Sample Input 1

```
GGGGGGGGGGGGGGGGGG
4
1 3
2 9
3 4
0 17
```

Sample Output 1

```
3
8
2
18
```

Sample Input 2

```
KGCGUSGGGGOGGELGGG
5
1 3
2 9
3 4
0 0
0 17
```

Sample Output 2

```
2
5
1
0
11
```