Problem A. As Easy As Possible

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

As we know, the NTU Final PK contest usually tends to be pretty hard. Many teams got frustrated when participating NTU Final PK contest. So I decide to make the first problem as "easy" as possible. But how to know how easy is a problem? To make our life easier, we just consider how easy is a string.

Here, we introduce a sane definition of "easiness". The easiness of a string is the maximum times of "easy" as a subsequence of it. For example, the easiness of "eeaseyaesasyy" is 2. Since "easyeasy" is a subsequence of it, but "easyeasyeasy" is too easy.

How to calculate easiness seems to be very easy. So here is a string s consists of only 'e', 'a', 's', and 'y'. Please answer m queries. The i-th query is a interval $[l_i, r_i]$, and please calculate the easiness of $s[l_i..r_i]$.

Input

The first line contains a string s. The second line contains an integer m. Each of following m lines contains two integers l_i, r_i.

- $1 \le |s| \le 10^5$
- $1 \le m \le 10^5$
- $1 \le l_i \le r_i \le |s|$
- s consists of only '`e`', '`a`', '`s`', and '`y`'

Output

For each query, please output the easiness of that substring in one line.

Examples

standard input	standard output
easy 3 1 4 2 4 1 3	1 0 0
eeaseyaesasyy 4 1 13 2 12 2 10 3 11	2 2 1 0