NOI '23 P5 - String

Time limit: 2.0s **Memory limit:** 512M

Little Y is a college student who is currently doing researches related to strings. Little Y learned about the following definitions regarding strings:

- Given a string $s[1:n]$ of length n, we define its substring $s[l:r](1 \leq l \leq r \leq n)$ as the new string obtained by selecting $s[l], s[l+1], \ldots, s[r]$ in order and concatenating them.
- Given a string $s[1:n]$ of length n, we define its reversed result $R(s)$ as the string obtained by concatenating $s[n], s[n-1], \ldots, s[1]$ in order, which is the string obtained by reversing the original string.
- Given two strings $a[1:n]$ and $b[1:n]$ of equal length n, we define a to be lexicographically smaller than b if and only if there exists $1 \leq i \leq n$ such that for any $1 \leq j < i, a[j] = b[j]$, and $a[i] < b[i]$.

After understanding the above definitions, Little Y came up with the following problem:

Given a string $s[1:n]$ of length n, there are q queries, each query giving two parameters i and r. You need to find out how many values of l satisfy the following conditions:

- $1 \leq l \leq r$.
- $s[i:i+l-1]$ is lexicographically smaller than $R(s[i+l:i+2l-1])$.

Little Y would like to ask for your help in solving this problem.

Input Specification

This problem has multiple test data sets.

The first line of the input contains two integers c and t, which represent the test case number and the number of test data sets. $c = 0$ represents that this test case is a sample test.

Then, each set of test data is given as input in order. For each set of test data:

The first line of input contains two positive integers, n and q, which represent the length of the string and the number of queries respectively.

The second line of input contains a string s of length n that only consists of lowercase letters.

Then q lines follow, each containing two positive integers, i and r, representing a query. It is guaranteed that $i + 2r - 1 \leq n$.

Output Specification

For each query of each set of test data, output a line containing an integer, representing the number of ls satisfying the requirements.

Sample Input 1

```
0 2
9 3
abacababa
1 4
2 4
3 3
9 3
abaabaaba
1 4
2 4
3 3
```

Sample Output 1

```
4
0
3
2
0
2
```

Explanation for Sample Output 1

For the first set of test data in the sample:

- When $l = 1$, $s[i : i + l - 1] =$ a, $R(s[i + l : i + l + l - 1]) =$ b.
- When $l = 2$, $s[i : i + l - 1] =$ ab, $R(s[i + l : i + l + l - 1]) =$ ca.
- When $l = 3$, $s[i : i + l - 1] =$ aba, $R(s[i + l : i + l + l - 1]) =$ bac.
- When $l = 4$, $s[i : i + l - 1] =$ abac, $R(s[i + l : i + l + l - 1]) =$ baba.

In all four cases, $s[i : i + l - 1]$ is lexicographically smaller than $R(s[i + l : i + 2l - 1])$, so the answer is 4.

Additional Samples

Sample inputs and outputs can be found here.

- Sample 2 (`ex_string2.in` and `ex_string2.ans`) corresponds to test case 5.
- Sample 3 (`ex_string3.in` and `ex_string3.ans`).
- Sample 4 (`ex_string4.in` and `ex_string4.ans`) corresponds to test cases 24-25.

Constraints

For all test data, it is guaranteed that: $1 \le t \le 5, 1 \le n \le 10^5, 1 \le q \le 10^5, 1 \le i + 2r - 1 \le n$. The string s only consists of lowercase letters.

Test ID	$n \le$	q	Additional Constraints
1	≤ 5	≤ 5	A
2	≤ 10	≤ 10	
3	≤ 20	≤ 20	
4	≤ 50	≤ 50	
5	$\le 10^2$	$\le 10^2$	
6	$\le 10^3$	$\le 10^3$	None
7	$\le 2\,000$	$\le 2\,000$	
8	$\le 3\,000$	$\le 3\,000$	
9	$\le 4\,000$	$\le 4\,000$	
10	$\le 23\,333$	$\le 23\,333$	A
11	$\le 5 \times 10^4$	$\le 5 \times 10^4$	
12	$\le 75\,000$	$\le 75\,000$	
13	$\le 9 \times 10^4$	$\le 9 \times 10^4$	
14	$\le 10^5$	$\le 10^5$	
15	$\le 23\,333$	$\le 23\,333$	B
16	$\le 75\,000$	$\le 75\,000$	
17	$\le 9 \times 10^4$	$\le 9 \times 10^4$	
18	$\le 10^5$	$\le 10^5$	

19	$\leq 23\,333$	$\leq 23\,333$	None
20	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	
21	$\leq 75\,000$	$\leq 75\,000$	
22	$\leq 9 \times 10^4$	$\leq 9 \times 10^4$	
23	$\leq 95\,000$	$\leq 95\,000$	
24	10^5	10^5	
25			

Additional Constraint A: It is guaranteed that the input string only consists of `a` and `b`, and each character is uniformly chosen from `a` and `b` at random.

Additional Constraint B: It is guaranteed that every pair of adjacent characters in the input string are distinct.