

Problem Caesar Is Back

C header `caesar.h`
C++ header `caesar.h`

Your favourite emperor CAESAR is back! He gives you the following problem. He defines a *1-step transformation* in the following way: a 1-step transformation transforms an 'a' into a 'b', a 'b' into a 'c', ..., a 'y' into a 'z', and finally a 'z' into an 'a'. Furthermore, for any non-negative integer k , he defines a *k-step transformation* as a 1-step transformation applied k times. For example, a 3-step transformation transforms an 'a' into a 'd'. Note that a 0-step transformation does nothing i.e. it transforms an 'a' into an 'a', a 'b' into a 'b', and so on.

CAESAR provides you with two strings A and B , each of length n . These are both **indexed from 0**. Furthermore, he provides you with q intervals $[l, r]$ where $0 \leq l \leq r < n$. For each interval $[l, r]$, he wants you to find the number of pairs (x, y) such that $l \leq x \leq y \leq r$ and there exists a k such that, for all $x \leq i \leq y$, we have that B_i is the k -step transformation of A_i .

For example, if $n = 3$, $A = \text{aac}$, $B = \text{bbc}$, $l = 0$ and $r = 2$ then the valid pairs are $(0, 0)$, $(0, 1)$, $(1, 1)$ and $(2, 2)$. For $(0, 0)$, $(0, 1)$, $(1, 1)$ we take $k = 1$, and for $(2, 2)$ we take $k = 0$.

Interaction Protocol

The contestant must implement two functions:

```
void init(int n, int q, char A[], char B[]);  
long long query(int l, int r);
```

The function `init` will be called **exactly once**, at the beginning of the interaction. The function will be supplied with the values n and q and with the two strings, A and B . Then, the committee will call the function `query` q times. It will be supplied with the values l and r , representing a query. The contestant must return one integer, the answer for the interval $[l, r]$, according to the statement.

Attention! The contestant must not implement the main function, and must #include the caesar.h header! Contestants are allowed to use global variables and other functions.

Restrictions

- $1 \leq n \leq 1\,000\,000$.
- $1 \leq q \leq 1\,000\,000$.
- A and B contain lowercase English letters only.

#	Points	Restrictions
1	5	$A = \text{aaa}\dots, B = \text{bbb}\dots$
2	9	A and B contain only 'a' and 'n'
3	10	$n \leq 100, q \leq 1\,000$
4	15	$n \leq 1\,000, q \leq 300\,000$
5	30	$q \leq 100\,000$
6	31	No further restrictions

Examples

Input	Output
<pre>init(3, 1, "aac", "bbc") query(0, 2)</pre>	4
<pre>init(5, 3, "abcde", "bcdyz") query(1, 3) query(0, 2) query(4, 4)</pre>	4 6 1
<pre>init(20, 20, "aggccdaloxgnakfivqd", "ckjdfgdnszczhpdmilxrh") query(2, 9) query(8, 10) query(2, 11) query(3, 4) query(9, 15) query(6, 12) query(8, 10) query(8, 10) query(2, 5) query(5, 14) query(8, 13) query(5, 11) query(0, 1) query(6, 14) query(0, 5) query(2, 2) query(0, 3) query(9, 14) query(3, 12) query(8, 11)</pre>	11 4 14 2 8 8 4 4 5 12 7 9 2 10 7 1 4 7 14 5

Explanations

First example For the interval $[0, 2]$ the valid pairs are $(0, 0)$, $(0, 1)$, $(1, 1)$ and $(2, 2)$. For the first three pairs we take $k = 1$ which transforms the letters 'a' into letters 'b'. For the last one we take $k = 0$ which leaves the letter 'c' as it is.

Second example For the interval $[1, 3]$ we have the valid pairs $(1, 1)$, $(1, 2)$, $(2, 2)$ and $(3, 3)$. For $(1, 1)$, $(1, 2)$ and $(2, 2)$ we choose $k = 1$ which transforms the letter 'b' into 'c' and the letter 'c' into 'd' respectively. For $(3, 3)$ we choose $k = 21$, because it transforms the letter 'd' into 'y'. Therefore, the answer is 4. For the interval $[0, 2]$ every possible pair is valid. For all of them we choose $k = 1$, which makes the letter 'a' into 'b', the letter 'b' into 'c' and the letter 'c' into 'd' respectively. Therefore, the answer is 6. For the interval $[4, 4]$ the only pair that satisfies the statement is $(4, 4)$, for which we choose $k = 21$, which transforms the letter 'e' into 'z'. Therefore, the answer is 1.