Hamming Distance

You are given a string S, consisting of N small latin letters '`a`' and '`b`'. You are also given M queries to process. The queries are as follows:

- C l r ch: all the symbols in the string, starting at the l^{th}, ending at the r^{th} become equal to ch;
- S l_1 r_1 l_2 r_2: swap two consecutive fragments of the string, where the first is denoted by a substring starting from l_1 ending at r_1 and the second is denoted by a substring starting at l_2 ending at r_2;
- R l r: reverse the fragment of the string that starts at the l^{th} symbol and ends at the r^{th} one;
- W l r: output the substring of the string that starts at the l^{th} symbol and ends at the r^{th} one;
- H l_1 l_2 len: output the *Hamming distance* between the consecutive substrings that starts at l_1 and l_2 respectively and have the length of len.

Everything is 1-indexed here.

Input Format
The first line of input contains a single integer N — the length of the string.
The second line contains the initial string S itself.
The third line of input contains a single integer M — the number of queries.
Then, there are M lines, each denotes a query of one of the types above.

Output Format
For each query of the type `W` or the type `H` output an answer on the separate line of output.

Constraints
$1 \leq N \leq 50000$
$1 \leq M \leq 75000$
Total number of characters printed in W-type queries will not exceed $2 \cdot 10^6$
For C-type, R-type, W-type queries: $1 \leq l \leq r \leq N$; ch equals either `a`, or `b`
For S-type queries: $1 \leq l_1 \leq r_1 < l_2 \leq r_2 \leq N$
For H-type queries: $1 \leq l_1, l_2 \leq N$; $l_i + len - 1 \leq N$; $1 \leq len \leq N$.

Sample Input

```
10
aabbbabbab
10
R 1 5
W 3 8
C 4 4 a
H 2 1 9
S 5 9 10 10
H 1 2 9
C 3 3 b
H 7 8 2
C 2 9 a
S 7 9 10 10
```

Sample Output

```
baaabb
4
5
1
```

This challenge is only available for programming in C, C++, Java, C#. Standard TL apply as given in Environment