E - Alternating String Editorial by en_translator

For a sequence $A=(A_1,A_2,\ldots,A_{N-1})$ of length $(N-1)$, let $A_i=0$ if $S_i=S_{i+1}$ and $A_i=1$ if $S_i \neq S_{i+1}$.

Then a query `1 L R` of the first type modifies A as $A_{L-1} \leftarrow (1-A_{L-1})$ and $A_R \leftarrow (1-A_R)$.
Here, if $L=1$ or $R=N$, the former or latter update is unneeded, respectively.

On the other hand, a query `2 L R` of the second type is `Yes` if $A_L=A_{L+1}=\cdots=A_{R-1}=1$, and `No` otherwise.
Noticing that each A_i is 0 or 1, one can decide the answer to be `Yes` if $A_L+A_{L+1}+\cdots+A_{R-1}=R-L$, and `No` otherwise.

These operations can be achieved with a segment tree.
Both queries can be processed in $O(\log N)$ time each, the problem can be solved in a total of $O(Q\log N)$ time, which is fast enough.
Thus, the problem has been solved.

When implementing the algorithm above, beware of possibly necessary exception handling when $N=1$, where the length of A is 0. One can either code exceptional procedure, or allocate a bit longer A.

One can also solve this problem in the same time complexity by managing i such that $S_i=S_{i+1}$ in an order set.

Sample code C++:

```cpp
#include <bits/stdc++.h>
#include <atcoder/segtree>
using namespace std;
using namespace atcoder;
int op(int a, int b) { return (a+b); }
int e() { return 0; }

int main(void){
        int n,q;
        string s;
        int x,l,r;

```

```
    cin>>n>>q;
    cin>>s;
    segtree<int, op, e> seg(n+1);
    for(int i=0;i<n-1;i++)if(s[i]!=s[i+1])seg.set(i+1,1);
    for(int i=0;i<q;i++){
        cin>>x>>l>>r;
        if(x==1){
            seg.set(l-1,1-seg.get(l-1));
            seg.set(r,1-seg.get(r));
        }
        else{
            if(seg.prod(l,r)==(r-l))cout<<"Yes"<<endl;
            else cout<<"No"<<endl;
        }
    }
    return 0;
}
```