1 Palindromic Bracket Sequence

1.1 Problem

Bob wants to learn a new programming language. After he wrote a `Hello World` program he is ready for something more advanced. He wants to write a program which recognizes palindromic bracket sequences.

A palindrome is a string that is the same if read forward and backward. For example `"aba"` and `"aa"` are palindromes but `"Hello"` and `"db"` are not.

The following rules (completely) define whether a string is a valid bracket sequence:

1. The empty string is valid.
2. If A is valid, then `(`A`)`, `{`A`}` and `[`A`]` are also valid.
3. If the strings A and B are valid, then AB is valid.

Bob has a string s. There are three operations he can perform on this string:

(a) Given i $(1 \le i \le |s|)$, remove the bracket at position i.

(b) Given i and b $(1 \le i \le |s| + 1, b \in \{$ `(`, `)`, `{`, `}`, `[`, `]` $\})$, insert b at position i (move all brackets after i).

(c) Given i and j $(1 \le i, j \le |s|)$, swap $s[i]$ and $s[j]$.

To help Bob you have to write a program that performs the following queries:

1. Print `"yes"` if s is a palindrome and a valid bracket sequence or `"no"` if not.
2. Print the minimum number of operations required to transform s into a palindrome and a valid bracket sequence
3. Given i $(1 \le i \le |s|)$, perform operation (a) on s.
4. Given i and b $(1 \le i \le |s| + 1, b \in \{$ `(`, `)`, `{`, `}`, `[`, `]` $\})$, perform operation (b) on s.
5. Given i and j $(1 \le i, j \le |s|)$, perform operation (c) on s.

For queries of type 3, 4, or 5, it is guaranteed that the given parameters describe a valid operation on the current value of s.

1.2 Input

The input contains multiple testcases. The first line contains a single integer t $(1 \le t \le 10)$, the number of testcases.

Each testcase begins with two numbers n and q $(1 \le n, q \le 10^5)$, the length of the initial string and the number of queries. The next line contains a string s $(|s| = n, s[i] \in \{$ `(`, `)`, `{`, `}`, `[`, `]` $\})$. The next q lines each contain an integer x $(1 \le x \le 5)$, the type of the operation you need to perform, and possibly some more parameters as described above.

1.3 Output

For each query of type 1 or 2 give the correct answer in a new line.

1.4 Sample Data

Input	Output
2 2 3 (] 1 3 2 2 2 4) (1 5 1 2 4 2 [1	no 1 no no

4