

# 1011 10010

## Problem Description

对于一个二进制数  $x(x > 0)$ ，设  $2^y = \text{lowbit}(x)$ ，设  $z = \lfloor x/2^{y+1} \rfloor$ 。

定义  $f$  函数： $f(0) = 0$ ，当  $x > 0$  时

$$f(x) = \begin{cases} y & z = 0 \\ f(z) + 2 & z \neq 0 \wedge \text{lowbit}(z) = \text{lowbit}(x) \times 2 \\ y & z \neq 0 \wedge \text{lowbit}(z) \neq \text{lowbit}(x) \times 2 \end{cases}$$

给出一个长度为  $n$  的 01 序列  $a$ 。有  $m$  次操作，每次操作都形如以下的两种：

- 1 l r：查询序列  $a$  的区间  $[l, r]$  组成的二进制数  $x$  的  $f(x)$  值。
- 2 x：令  $a_x$  反转（0 变成 1，1 变成 0）。

†  $\text{lowbit}(i)$  表示  $i$  在二进制表示下最低位的 1 及其后面所有的 0 构成的数值。

†  $\wedge$  表示逻辑且。

## Input

每个测试点中包含多组测试数据。输入的第一行包含一个正整数  $T(1 \leq T \leq 110)$ ，表示数据组数。对于每组测试数据：

第一行两个正整数  $n, m(1 \leq n \leq 5.1 \times 10^5, 1 \leq m \leq 5 \times 10^5)$ ，分别表示字符串长度和操作次数。

第二行一个长度为  $n$  的字符串，表示初始的 01 序列  $a$ 。

接下来  $m$  行，每行若干个整数，第一个数  $\text{opt}$  表示操作类型：

- 若  $\text{opt} = 1$ ，则后面两个数  $l, r (1 \leq l \leq r \leq n)$ ，表示查询的区间。
- 若  $\text{opt} = 2$ ，则后面一个数  $x (1 \leq x \leq n)$ ，表示修改的位置。

保证所有测试数据中  $n$  之和不超过  $6.2 \times 10^5$ ， $m$  之和不超过  $7.1 \times 10^5$ 。

## Output

对于每组测试数据：对于每一个  $\text{opt} = 1$  的操作，输出一行一个数表示答案。

## Sample Input

```
1
11 3
10001001000
1 2 9
2 10
1 6 10
```

## Sample Output

```
4
3
```