

二进制 (binary)

【题目描述】

pupil 发现对于一个十进制数，无论怎么将其的数字重新排列，均不影响其是不是 3 的倍数。他想研究对于二进制，是否也有类似的性质。于是他生成了一个长为 n 的二进制串，希望你对于这个二进制串的一个子区间，能求出其有多少位置不同的连续子串，满足在重新排列后（可包含前导 0）是一个 3 的倍数。两个位置不同的子区间指开始位置不同或结束位置不同。由于他想尝试尽量多的情况，他有时会修改串中的一个位置，并且会进行多次询问。

【输入格式】

从文件 *binary.in* 中读入数据。

输入第一行包含一个正整数 n ，表示二进制数的长度。

之后一行 n 个空格隔开的整数，保证均是 0 或 1，表示该二进制串。

之后一行一个整数 m ，表示询问和修改的总次数。

之后 m 行每行为 1 i ，表示 pupil 修改了串的第 i 个位置（0 变成 1 或 1 变成 0），或 2 $l\ r$ ，表示 pupil 询问的子区间是 $[l,r]$ 。

串的下标从 1 开始。

【输出格式】

输出到文件 *binary.out* 中。

对于每次询问，输出一行一个整数表示对应该询问的结果。

【样例 1 输入】

```
4
1 0 1 0
3
2 1 3
1 3
2 3 4
```

【样例 1 输出】

```
2
3
```

【样例 1 解释】

对于第一个询问，区间 $[2,2]$ 只有数字 0，是 3 的倍数，区间 $[1,3]$ 可以重排成 $011_{(2)} = 3_{(10)}$ ，是 3 的倍数，其他区间均不能重排成 3 的倍数。

对于第二个询问，全部三个区间均能重排成 3 的倍数（注意 00 也是合法的）。

【子任务】

对于 20% 的数据， $1 \leq n, m \leq 100$ 。

对于 50% 的数据， $1 \leq n, m \leq 5000$ 。

对于 100% 的数据， $1 \leq n, m \leq 100000$ ， $l \leq r$ 。