

# Changing Bits



Let A and B be two N bit numbers (MSB to the left). You are given initial values for A and B, and you have to write a program which processes three kinds of queries:

- `set_a idx x`: Set `A[idx]` to `x`, where  $0 \leq \text{idx} < N$ , where `A[idx]` is `idx`'th least significant bit of A.
- `set_b idx x`: Set `B[idx]` to `x`, where  $0 \leq \text{idx} < N$ , where `B[idx]` is `idx`'th least significant bit of B.
- `get_c idx`: Print `C[idx]`, where `C=A+B`, and  $0 \leq \text{idx}$

## Input Format

First line of input contains two integers N and Q consecutively ( $1 \leq N \leq 100000$ ,  $1 \leq Q \leq 500000$ ). Second line is an N-bit binary number which denotes initial value of A, and the third line is an N-bit binary number denoting initial value of B. Q lines follow, each containing a query as described above.

## Output Format

For each query of the type `get_c`, output a single digit 0 or 1. Output must be placed in a single line.

## Sample Input

```
5 5
00000
11111
set_a 0 1
get_c 5
get_c 1
set_b 2 0
get_c 5
```

## Sample Output

```
100
```

## Explanation

- `set_a 0 1` sets 00000 to 00001
- $C = A + B = 00001 + 11111 = 100000$ , so `get_c[5] = 1`
- from the above computation `get_c[1] = 0`
- `set_b 2 0` sets 11111 to 11011
- $C = A + B = 00001 + 11011 = 011100$ , so `get_c[5] = 0`

The output is hence concatenation of 1, 0 and 0 = 100