

# Plagia

Time Limit: 1.5s  
Memory Limit: 2048MB

## Task Statement

Limli is managing a December contest consisting of  $N$  participants, with each participant having received a total score of  $A_i$  (0-indexed) so far. Via complaints flooding the clarification system, it has come to light that certain submissions are not being graded - upon closer inspection, this was determined to be a result of “unfortunate hash collisions” between submissions causing the automated judge to ignore newer uploads as accidental resubmissions of older ones.

Unable to put an effective end to this problem within contest time, Limli instead decided to apply the following measure as a temporary solution: if participants  $x$  and  $y$  have solutions with identical hashes (thus causing a judge failure), she will perform a bitwise exclusive-or (xor) against the solution's score  $z$  on the scores of every participant with index between  $x$  and  $y$  inclusive.

Of course, Limli still needs to determine some data about the contestants' scores in order to gauge the suitability of the contest's difficulty; between some xor operations on the participants' scores, she will select a range of participants  $u$  to  $v$  inclusive and ask for the sum of current scores of all participants within the range. As such, given the initial scores of all contestants and a sequence of  $M$  range sum queries and range xor updates, evaluate these queries.

## Input

The first line of input consists of two integers,  $N$  and  $M$ .

The second line of input consists of  $N$  integers representing the array  $A$ .

$M$  lines will follow. The first character of each line will be  $T$ .

If  $T = 0$ , then it represents a query. Two integers will then follow,  $u$  and  $v$ .

If  $T = 1$ , then it represents an update. Three integers will then follow,  $x$ ,  $y$ , and  $z$ .

## Output

For each line of query, you are to output the sum of  $A[u]$  to  $A[v]$  inclusive, on a single line.

## Constraints

For all testcases,  $1 \leq N$ ,  $M \leq 100,000$ ,  $0 \leq u \leq v < N$ ,  $0 \leq x \leq y < N$ ,  $0 \leq A_i$ ,  $z \leq 10^9$ .

Subtask 1 (8%):  $1 \leq N$ ,  $M \leq 1,000$

Subtask 2 (17%): For each query,  $u = v$

Subtask 3 (17%): For each update,  $x = y$

Subtask 4 (14%):  $0 \leq A_i \leq 1$ ,  $0 \leq z \leq 1$

Subtask 5 (24%):  $0 \leq A_i \leq 3, 0 \leq z \leq 2$   
Subtask 6 (20%): No additional constraints.  
Subtask 7 (0%): Sample testcases.

## Sample Input

```
6 9
1 4 6 7 3 15
1 0 3 4
1 2 3 2
0 1 4
0 0 5
1 1 5 19
0 3 3
1 2 2 7
0 1 2
0 3 4
```

## Sample Output

```
4
24
18
39
34
```