

# Groups

A4 / 250



Ivancho develops new software. He needs a module that by given list of N numbers - A can perform fast two operations:

1) **Update**: In this query the program must update the value of all elements between elements with numbers l and r including them. For each element  $A[i]$  it's new value is:  $((A[i]) \text{ XOR } (K+i)) \bmod 64$  where K is value given in the input for each query.

Note: We assume that the array elements are numerated from 1 to N:

2) **Get**: In this query your program must print single number – how many groups of K equal elements there are in the interval from l to r including them. We assume that each element is unique although their values are equal. This number can be very big so you have to print it modulo 18181.

Ivancho is your best friend and you want to help him. You have to write the program **groups**, that by given array A with size N and Q queries from type 1 or 2 finds the answer of each query of type 2.

## Input

On the first row of the input file `groups.in` are entered 2 numbers - N and Q.

On the second row are entered N numbers from 0 to 63.

Then there are Q rows, consisting of 4 numbers, denoting: The type of the query, l, r and K. The interval **includes** l and r.

## Output

On each row of the output file `groups.out` you must print 1 number – the answer for each query of type 2.

## Constraints:

$0 \leq A[i] \leq 63$ .

$1 \leq N \leq 300\,000$

$1 \leq Q \leq 10\,000$

$R-L < 20$  for query from type 1

$R-L < 300\,000$  for query from type 2

Note: The ML is 32 MB.

TL 1 sec

## Example

| Input (groups.in)   | Output (groups.out) |
|---------------------|---------------------|
| 10 5                | 1                   |
| 3 3 1 3 2 1 5 8 8 8 | 3                   |
| 2 1 3 2             | 5                   |
| 2 1 4 2             | 2                   |
| 2 1 9 2             |                     |
| 1 1 10 1            |                     |
| 2 1 9 2             |                     |