k Alternating Sum

Sameen has:

1. An array having N integers
2. Q friends

His friends are curious about the array. So, each of his friends asks Sameen a question about the array. Every question is described by 3 integers: i, j and k. In reply to a question, Sameen has to say the "k alternating sum" of the subarray starting at position i and ending at position j [1 based indexing]

"k alternating sum" of a subarray starting at position i and ending at position j can be calculated in the following way:

Add the first k numbers[starting from position i]

Subtract the second k numbers[starting from position i+k]

Add the third k numbers[starting from position i+2*k]

Subtract the fourth k numbers[starting from position i+3*k]

And so on till adding/subtracting the j-th number…

(j-i+1) will be divisible by k.

[See sample Input/output and explanation section for more details]

Can you help Sameen in answering the questions?

Input

The first line of input contains two integers N and Q. The next line contains N integers, the numbers in the array. Then each of the following Q lines contains 3 integers i, j & k.

Output

For each query output an integer in a separate line, the answer for that query. Queries should be answered in the order given in the input.

Constraints:

$1 \leq k \leq 100000$

$1 \leq N \leq 100000$

$1 \leq Q \leq 100000$

$-1000000000 \leq$ Value of a number in the array ≤ 1000000000

(j-i+1) will be divisible by k.

Example

Input:

6 6

4 1 -2 -3 4 5

2 5 2

1 6 1

1 6 3

1 6 6

3 3 1

3 4 1

Output:

-2

3

-3

9

-2

1

Explanation:

In the first query, the subarray is [1, -2, -3, 4].

So "2 alternating sum" is equal to: [1-2]-[-3+4] = -2

For the second query, we get [4]-[1]+[-2]-[-3]+[4]-[5] = 3

N.B: Dataset is huge. Use faster I/O method.