

A. Greg and Array

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Greg has an array $a = a_1, a_2, \dots, a_n$ and m operations. Each operation looks as: l_i, r_i, d_i , ($1 \leq l_i \leq r_i \leq n$). To apply operation i to the array means to increase all array elements with numbers $l_i, l_i + 1, \dots, r_i$ by value d_i .

Greg wrote down k queries on a piece of paper. Each query has the following form: x_i, y_i , ($1 \leq x_i \leq y_i \leq m$). That means that one should apply operations with numbers $x_i, x_i + 1, \dots, y_i$ to the array.

Now Greg is wondering, what the array a will be after all the queries are executed. Help Greg.

Input

The first line contains integers n, m, k ($1 \leq n, m, k \leq 10^5$). The second line contains n integers: a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^5$) — the initial array.

Next m lines contain operations, the operation number i is written as three integers: l_i, r_i, d_i , ($1 \leq l_i \leq r_i \leq n$), ($0 \leq d_i \leq 10^5$).

Next k lines contain the queries, the query number i is written as two integers: x_i, y_i , ($1 \leq x_i \leq y_i \leq m$).

The numbers in the lines are separated by single spaces.

Output

On a single line print n integers a_1, a_2, \dots, a_n — the array after executing all the queries. Separate the printed numbers by spaces.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams of the `%I64d` specifier.

Examples

input	Copy
<pre>3 3 3 1 2 3 1 2 1 1 3 2 2 3 4 1 2 1 3 2 3</pre>	
output	Copy
<pre>9 18 17</pre>	
input	Copy
<pre>1 1 1 1 1 1 1 1 1</pre>	

output	Copy
2	

input	Copy
4 3 6 1 2 3 4 1 2 1 2 3 2 3 4 4 1 2 1 3 2 3 1 2 1 3 2 3	
output	Copy
5 18 31 20	