

E. Linear Kingdom Races

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are a car race organizer and would like to arrange some races in Linear Kingdom.

Linear Kingdom has n consecutive roads spanning from left to right. The roads are numbered from 1 to n from left to right, thus the roads follow in the order of their numbers' increasing. There will be several races that may be held on these roads. Each race will use a **consecutive** subset of these roads. Also, each race will pay some amount of money to you if this race is held. No races overlap in time, so some roads can be used in several races.

Unfortunately, some of the roads are in a bad condition and they need repair. Each road has repair costs associated with it, you are required to pay this cost to repair the road. A race can only take place if all the roads used in the race are renovated. Your task is to repair such roads (possibly all or none) that will maximize your profit. Your profit is defined as the total money you get from the races that are held minus the total money you spent to repair the roads. Note that you may decide not to repair any road and gain zero profit.

Print the maximum profit you can gain.

Input

The first line contains two single-space separated integers, n and m ($1 \leq n, m \leq 2 \cdot 10^5$), denoting the number of roads and the number of races, respectively.

Then n lines follow, each line will contain a single non-negative integer not exceeding 10^9 denoting the cost to repair a road. The costs are given in order from road 1 to road n .

Finally, m lines follow. Each line is single-space-separated triplets of integers. Each triplet will be given as lb , ub , and p ($1 \leq lb \leq ub \leq n$, $1 \leq p \leq 10^9$), which means that the race these three integers describe will use all the roads from lb to ub , inclusive, and if it's held you get p .

Output

Print a single integer denoting the maximum possible profit you can gain.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is recommended to use `cin`, `cout` stream (also you may use `%I64d` specifier).

Examples

input
7 4 3 2 3 2 1 2 3 1 2 5 2 3 5 3 5 3 7 7 5
output
4

input

2 1 0 3 1 2 5
output
2

input
3 1 10 10 10 1 3 10
output
0

Note
In the first sample the optimal solution is to repair roads 1, 2, 3, and 7. Three races will take place which nets you 15. The road repair costs 11, hence your profit is 4.