

Catch Fish

[English](#)

[Vietnamese](#)

During his childhood, Mirko liked to play “Sea battle” a lot, but once he got bored of it and invented a new game imaginatively called “Ships catch fish on a river”.

The game board consists of N fields lined up in a row and numbered 1 to N in ascending order from left to right. These fields represent a river on which M ships are to be placed. For each field the amount of fish swimming in that part of the river is known. Every ship has a specific length, i.e. it occupies the specific number of consecutive fields. A ship must somewhere drop an anchor but is allowed to do that in a certain field only. That means that one of the fields a ship will occupy is predetermined.

There can be only one ship or a part of only one ship on a single field of the board. We define the total amount of fish caught as the sum of amounts of fish swimming in all fields occupied by ships. Goal of the game is to place all ships on the river in a way that the total amount of fish caught is maximal.

While playing an instance of this game, Mirko is in doubt whether the way he placed ships is optimal. Therefore he asked you to help him calculate the maximum possible amount of fish caught.

Input

First line of the input file contains an integer N , the number of fields on board, $1 \leq N \leq 100000$.

In the next line there are N integers, separated by whitespaces, which represent the amounts of fish swimming in appropriate field, given in kilograms. These numbers will not be less than 1 nor greater than 100.

The next line contains an integer M , number of the ships, $1 \leq M \leq N$.

Each of the following M lines consists of two integers B and D separated by a whitespace. That means that the appropriate ship should drop the anchor in a field numbered with B and that its length is D .

Note: input data will always be such that the solution will exist

Output

The only line of the output file should contain the maximum possible total amount of fish caught.

Sample

brodovi.in

```
11
2 5 3 4 7 6 2 1 3 8 5
2
8 3
3 2
```

brodovi.out

20

brodovi.in

```
13
3 2 4 7 2 1 3 6 1 2 6 4 1
2
5 7
11 4
```

brodovi.out

38

brodovi.in

```
11
1 1 6 4 4 1 1 3 10 1 1
3
2 3
6 4
10 2
```

brodovi.out

31