

# Problem L

## What's Mine is Mine

Time Limit: 0.5 Seconds

The hot new video game “Mining Simulator” has just been released. In the game, rare earth mineral ores appear at certain times and you can mine them when they appear. After mining, you can later turn in the minerals for money. The quantity of mineral available during an appearance is proportional to the duration of the appearance and the price per unit of each mineral is decided beforehand.

The game contains a geological sensor that determines times when mineral ores will appear during a given day and at the beginning of each day you have a price list for each mineral. Assuming you mine out the mineral in exactly the amount of time that it is available, you cannot partially mine minerals (you must either not mine any of a given occurrence or mine all of it) and you can only mine one ore occurrence at a time.

Given a list of the prices of  $m$  minerals and  $n$  ore occurrences during a day, write a program to output the maximum amount of money you can earn from mining on that day. The mineral amount is the appearance time (end time – start time). You can mine an ore right after finishing the previous mining. In other words, one mined-ore's end time can be same as another mined-ore's start time. In the case depicted in Figure L.1, if you choose the mineral of type 1 that appears at time 2 and disappears at time 5, then the mineral amount is  $5 - 2 = 3$  and you earn  $3 \times 2 = 6$ . Next, if you choose the mineral of type 2 that appears at time 7 and disappears at time 11, then the mineral amount is  $11 - 7 = 4$  and you earn  $4 \times 3 = 12$ . Therefore, you earn 18 in total.

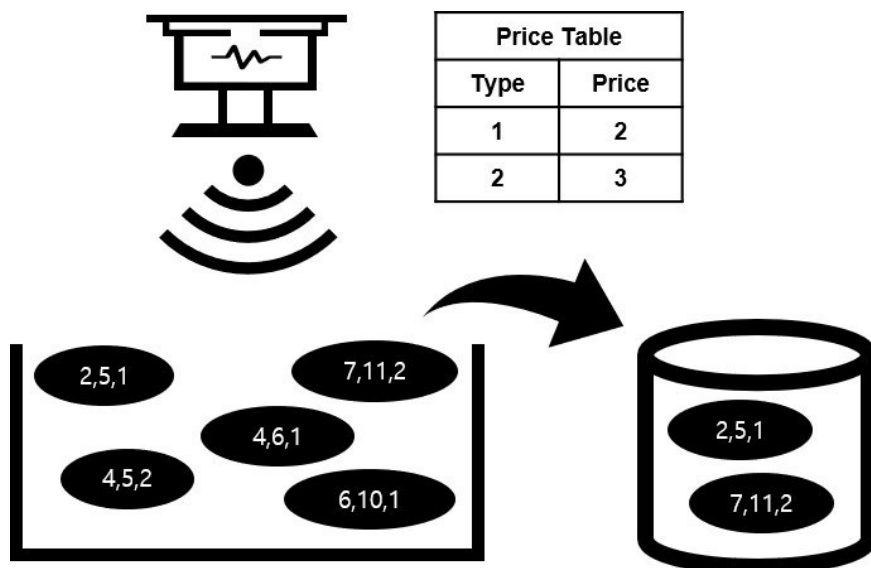


Figure L.1: A mining example. For each mineral  $(s, e, t)$ ,  $s$  is the start time,  $e$  is the end time and  $t$  is the mineral type. Therefore, the mineral amount is  $e - s$  and the possible earning is  $(e - s) \times t$ 's price.

### Input

Your program is to read from standard input. The input starts with a line containing two integers,  $m$  and  $n$  ( $1 \leq m \leq 100$ ,  $1 \leq n \leq 10,000$ ), where  $m$  is the number of types of minerals and  $n$  is the number of ore occurrences during the day. The mineral types are labeled from 1 to  $m$ . The following  $m$  lines contain a single

integer corresponding to the price of one unit of the  $i$ -th mineral type on that day (the price is between 1 and 10,000). The following  $n$  lines represent ore occurrences. Each line contains three integers,  $s$ ,  $e$ ,  $t$  where  $s$  is the start time,  $e$  is the end time and  $t$  is the mineral type in each ore occurrence with  $0 < s < e < 15,000$  and  $1 \leq t \leq m$ . The amount of mineral at each occurrence is  $e - s$ .

### Output

Your program is to write to standard output. Print exactly one line. The line should contain the maximum amount of money that can be earned on that day.

The following shows sample input and output for three test cases.

Sample Input 1	Output for the Sample Input 1
<pre>2 5 2 3 2 5 1 4 5 2 4 6 1 7 11 2 6 10 1</pre>	<pre>18</pre>

Sample Input 2	Output for the Sample Input 2
<pre>3 5 2 3 1 1 4 1 3 6 3 5 8 2 7 10 1 9 12 2</pre>	<pre>24</pre>

Sample Input 3	Output for the Sample Input 3
<pre>5 7 1 2 3 4 5 1 5 2 3 8 1 2 4 3 3 9 2 4 10 5 7 11 4 5 7 3</pre>	<pre>36</pre>