Problem 1 - Satellites

Elon Musk has received a contract from NASA in which he must launch N satellites into orbit in the sequence stated on a list. The weight of the satellites is also specified on the same list.

NASA also wants all the satellites being delivered on the same day, otherwise the deal would be terminated and transferred to Jeff Bezos (which Elon Musk cannot permit no matter the costs).

Elon Musk has at his disposal only K rocket launching platforms but no rockets, so he will have to construct them. To manufacture K rockets as quickly as possible, they all must be identical (only one design will be made) and as compact as possible.

What is the smallest payload limit for a rocket that would allow Elon to send the N satellites into orbit in at most K consecutive launches?

Data Format

Input

The input file is called “*satellites.in*”.

The first line contains the numbers **N** and **K**.

The second line contains **N** integers denoting the weight $\mathbf{w_i}$ for each satellite.

Output

The output file is called “satellites.*out*”.

It contains only one line with one integer **T**, which is the minimum payload the designed rocket must carry in order to send all **N** satellites into orbit in at most **K** consecutive launches.

Data Limits

1 <= K <= N <= 100.000

$1 <= w_i <= 10^{12}$

For 40% of the tests: 1 <= N <= 1000; 1 <= K <= 100; $1 <= w_i <= 100$

Time limits on vmchecker: C++ (0.15s), Java (1.0s)

Example:

satellites.in	**satellites.out**
5 3 1 2 2 1 2	3

Explanation

He must launch 5 satellites into orbit, and he has to build 3 rockets.

Solution:

Rocket 1 will deliver satellites 1 and 2.	---- 1 + 2 = 3
Rocket 2 will deliver satellites 3 and 4.	---- 2 + 1 = 3
Rocket 3 will deliver satellite 5.	---- 2

The minimum payload the designed rocket must carry is 3.

Problem 2 - ICE

Mr. Robot heard that Elon Musk is busy constructing rockets for a NASA contract so has woken up to equip all Tesla automobiles with internal combustion engines (he does not like EVs)

There is a total of N automobiles parked in a straight line in the factory parking lot.

The main issue Mr. Robot has is that, after modifying R vehicles, he will remain without energy. Thankfully, some Tesla cars still have some energy left in their batters, which Mr. Robot can recover without losses. However, this recovery process takes a specific amount of time for each car.

Help Mr. Robot in equipping all the electric vehicles with internal combustion engines with a minimum amount of time spent on energy extraction from batteries.

Data Format

Input

The input file is called “*ice.in*”.

The first line contains three numbers, **R** (the maximum number of electric vehicles Mr. Robot can modify before he remains without energy), **N** (the total amount of cars) and **E** (the number of cars that still have some energy remained in their batteries).

Each of the next **E** lines contains a tuple of integers <d, t, r>:

- **d** represents the distance from the starting position to the car with energy still available
- **t** represents the time elapsed in order to recover the energy from the car
- **r** represents the maximum number of cars Mr. Robot may rebuild using the recovered energy from the automobile

The list of tuples is sorted by **t**, ascending.

Output

The output file is called “*ice.out*”.

It contains a single number that represents the minimum amount of time Mr. Robot will spend retrieving energy to rebuild all the vehicles. If Mr. Robot is unable to repair all the vehicles, the number will be -1.

Data Limits

1 <= E <= **100000**

1 <= R <= N

1 <= d <= N <= 1.000.000

1 <= t <= 1.000.000

For 40% of the tests, N <= 1000

Time limits: C++ (0.6s), Java (7s)

Example:

ice.in	**ice.out**

7 30 5
6 50 8
7 20 7
12 40 9
13 50 11
23 10 7
	80

Solution explanation:

Mr. Robot can modify at the beginning a maximum of 7 cars. As a result, he can only reach the 6th and 7th car. For both cars, if he recovers the energy from them, he could go as far as car 14. The only difference between these two cars is that the energy from car 7 can be recovered faster than the energy from car 6, so it would be better for him to recover the energy from the car 7.

From 7th car, Mr. Robot can reach cars 12 and 13, that also have some energy remained into their batteries. We can see that the energy from the car 12 can be recovered faster that the energy from the car 13, but its energy won't get Mr. Robot to the next car that still has some energy into its battery (car 23) as he can only modify all the cars as far as 12 + 9 = 21. However, the other car available that still has some energy into it (car 13), would permit Mr. Robot to modify all the cars as far as 13 + 11 = 24, which makes it possible to reach to the last car that still has some energy remained into its battery (car 23).

The last car offers Mr. Robot enough energy to modify 7 more cars, thus reaching car 30. We display the total amount of time elapsed for the recovery process.