Deliveries

This question is graded for 1%.

Problem Statement

Tom has created a new app available locally which allows people to sign up and register to deliver packages for monetary rewards. Users can also sign up on the app to make delivery requests for a huge variety of items.

Surprisingly, despite the abundance of alternatives, Tom's app has received a tremendous amount of traction. The app differs from other delivery apps in the modus operandi; delivery requests are collated on the first day of every month and compiled into a list. New delivery requests are not allowed to be added in the middle of each month. Every delivery request must be paid for in advance to prevent the occurrences of pranks.

As the chief software developer, Tom is the first person who has access to the new list of delivery requests compiled on the first day of every month. Each entry on the list contains the amount of money paid for the delivery request (which is also the amount of amount earnt by the delivery rider, since the app does not charge commissions and only relies on advertising for revenue), and the deadline in seconds by which the pizza must be delivered. If the delivery request is fulfilled after the deadline, the request will not be considered to be fulfilled and no money goes to the delivery rider. Each package takes one second to deliver and only one can be delivered per second. There is no downtime in between delivering packages and taking on the next delivery request.

Tom is interested in raking in some extra income on the side, and since he is the first person to view the list of delivery requests, he will like to select all the requests he can fulfill and remove them from the list before releasing to the delivery personnel. Tom will like to maximise the amount of money he can make from a given list. Can you help Tom with this task?

Note: For the purposes of this lab, you MUST use the PriorityQueue data structure or your own implementation of a binary heap to solve the question. Failure to comply will result in a zero.

Input

28*24*3600

The first line contains a single integer N ($1 <= N <= 200000$) , the number of delivery requests on the given list . It is followed by N lines, each containing an integer T ($1 <= T <= 2419200$) representing the deadline by which the delivery request must be fulfilled, followed by a space, followed by an integer C ($1 <= C <= 100000$) representing the amount of money paid for the delivery.

Output

Output a single positive integer, which is the maximum amount of money Tom can get for fulfilling as many requests in the list as possible before the deadline.

Sample Input 1

```
4
1 10
1 20
2 10
2 20
```

Sample Output 1

```
40
```

Sample Input 2

```
5
1 10
2 11
3 12
4 13
5 14
```

Sample Output 2

```
60
```

Explanation

For Sample Input 1, it is not possible to fulfill all delivery requests; the second delivery request and the fourth delivery request can be fulfilled to maximise earnings. For Sample Input 2, all delivery requests can be fulfilled.