Competitive Programming Contest: Greedy

A. Frenster Week

1.5 seconds, 256 megabytes

Raam, a diligent university student residing in a college dormitory, faces a challenge during the penultimate week of the semester due to a concert being held in the parking lot. Raam has a meticulous list detailing the duration of each academic task, down to the millisecond. Unfortunately, the loud music from the concert disrupts his focus, making it impossible for him to concentrate on his studies.

To mitigate this, the event organizers provide Raam with precise intervals of quiet time during which no music will be played. These intervals are specified by their start and end times, also down to the millisecond.

Raam's goal is to complete as many tasks as possible during these quiet intervals. However, he cannot pause working on a task when music plays, as he loses his train of thought, and the duration of each task and the quiet intervals are such that he can only complete one task per interval.

Given a list of times t_i (in milliseconds) that each task will take and a list of times l_j (in milliseconds) specifying the lengths of the intervals when no music is being played, what is the maximum number of tasks that Raam can complete?

Input

The first line of input contains a pair of integers n and m where n is the number of tasks and m is the number of time intervals when no music is played.

The second line consists of a list of integers $t_1, t_2, \ldots, t_n$ indicating the length of time of each task that Raam needs to complete.

The final line consists of a list of times $l_1, l_2, \ldots, l_m$ indicating the length of time of each quiet interval when Raam is at work.

You may assume that $1 \leq n, m \leq 10^6$, and that $100,000 \leq t_i, l_j \leq 199,999$ for each task i and each quiet interval j.

Output

Output consists of a single line containing a single integer indicating the number of tasks that Raam can accomplish from his list during this week.

Scoring

Group	Points	Constraints
1	8	$1 \leq n, m \leq 2 \cdot 10^4$
2	8	$1 \leq n, m \leq 2 \cdot 10^5$
3	4	No additional constraints

input

```
5 4
150000 100000 160000 100000 180000
190000 170000 140000 160000
```

output

```
4
```

input

```
4 4
180000 185000 199999 100000
199999 180000 170000 120000
```

output

```
3
```

input

```
3 3
199999 180000 170001
199999 170000 180000
```

output

```
2
```

In Example 1:

- Raam has 5 tasks to complete.
- There are 4 quiet intervals available during which Raam can work without disturbance.
- The durations of the tasks are: 150000, 100000, 160000, 100000, and 180000 milliseconds.
- The durations of the quiet intervals are: 190000, 170000, 140000, and 160000 milliseconds.

Based on the statements above, there are 4 tasks that Raam can complete as below:

1. Raam can complete the 1^{st} task (150000 ms) within the 2^{nd} quiet interval (170000 ms).
2. Raam can complete the 2^{nd} task (100000 ms) within the 3^{rd} quiet interval (140000 ms).
3. Raam can complete the 3^{rd} task (160000 ms) within the 4^{th} quiet interval (160000 ms).
4. Raam can complete the 5^{th} task (180000 ms) within the 1^{st} quiet interval (190000 ms).

Therefore the output is: 4

In Example 2:

There are 3 tasks that Raam can complete as below:

1. Raam can complete the 1^{st} task $(180000$ ms) within the 2^{nd} quiet interval $(180000$ ms).
2. Raam can complete the 2^{nd} task $(185000$ ms) within the 1^{st} quiet interval $(199999$ ms).
3. Raam can complete the 4^{th} task $(100000$ ms) within the 3^{rd} quiet interval $(170000$ ms).

Therefore the output is: 3

B. Gungnir Throwing Game

5 seconds, 256 megabytes

Feeling a bit restless in the grand halls of Asgard, Odin decided to start his spear-throwing training. He picked up his legendary spear, Gungnir, and aimed at various targets, hitting them every time. Odin wandered through the realms, tossing Gungnir with varying force. The gods and goddesses watched in amusement as he turned a routine training into a celestial pastime. Loki, intrigued by Odin's training, suddenly glowed up with a mischievous gleam in his eye, and suggested that Odin should make this into a game. He then wrote down some rules to play the game with:

1.Each game consists of n linear tiles. Odin starts at the first tile, and he can only move towards the final tile and to the tile where Gungnir landed.

2.The maximum throwing power of the mighty weapon, p_i, is now the number shown on the ith tile, where the ith tile is the tile Odin is stepping on currently.

3.The objective is to determine if Odin can reach the final tile. If yes, determine also the minimum times Odin would need to throw Gungnir.

Odin wielding the Gungnir and preparing to throw it.

Odin does not want to be made fun of by Loki, so he tries his best to win every game with minimum throws. Your objective is the same as Odin's, so you think of a clever way to help Odin...

Input

The first line of input contains an integer n $(1 \leq n \leq 1.5 \cdot 10^7)$ — the number of tiles.

The second line of input contains n integers, $p_1, p_2, \ldots, p_n$ ($0 \leq p_i \leq 1500$) — the maximum throwing power of Gungnir at that tile, p_i.

Output

Output the minimum number of times Odin needs to throw Gungnir if he can make it to the final tile. If Odin cannot make it to the final tile, output "no way".

Scoring

Group	Points	Constraints
1	5	$n \leq 250$, $1 \leq p_i \leq 25$
2	11	$n \leq 20000$, $p_i \leq 100$
3	19	No additional constraints

input

```
4
1 5 1 1
```

output

```
2
```

input

```
8
2 3 0 1 1 0 6 1
```

output

```
no way
```

input

```
5
1 1 1 1 0
```

output

```
4
```

C. The Boba Dilemma

3 seconds, 1024 megabytes

My, oh, my! Another boba store has opened up shop here in SS15, Subang Jaya. Being a massive success, hordes of customers have been queuing around the block for a chance to taste the shop's sweet, tapioca goodness, with the lines forming up until the late evening.

Unfortunately for the customers, the shop has to close at some point in the day. However, the shop owner, with their intuitive entrepreneurial skills, plans to hold a rush hour event where some of the customers can be selected to skip to the front of the queue, in order to maximize shop revenue.

The idea is promising, but the store owner knows most customers cannot wait to be selected forever. Some of them must leave after a certain amount of time to avoid being caught in worsening traffic in the area.

The boba shop owner is very efficient in that they only require one minute for them to serve one customer. A customer i, though with big pockets, cannot stay longer than t_i to avoid the rough nearby traffic.

The store owner is also very risk averse, and thus would also like to run multiple calculations or tests before deciding to hold the event. Can you help them to calculate the possible maximum revenue that can be generated by the store in such an event?

Input

The first line of input contains an integer k $(1 \leq k \leq 300)$, the number of test cases.

The first line of each test case contains two integers n $(1 \leq n \leq 5 \cdot 10^3)$ and t $(1 \leq t \leq 10^3)$, the number of customers queuing up for the boba store and the remaining amount of time in minutes until the boba store closes.

After the first line in each test case, there are also n lines of input, each of which consists of two integers, RM_i $(1 \leq RM_i \leq 10^5)$ and t_i $(0 \leq t_i < t)$, meaning the amount of ringgit customer i has and the maximum amount of time in minutes where customer i can wait before they will leave the queue to go home if they have yet to be selected.

Output

For each test case, output one line with the maximum amount of ringgit the boba store owner can get, m, from the customers queuing for the rush hour event before the boba store closes for the day.

Scoring

Group	Points	Constraints
1	22	$k \leq 100$, $n \leq 1000$, $t \leq 500$
2	14	$k \leq 200$, $n \leq 2500$, $t \leq 750$
3	9	No additional constraints

input

```
1
4 4
1000 1
2000 2
500 2
1200 0
```

output

```
4200
```

input

```
1
3 4
1000 0
2000 1
500 1
```

output

```
3000
```

input

```
2
4 4
1000 1
2000 2
500 2
1200 0
3 4
1000 0
2000 1
500 1
```

output

```
4200
3000
```