

Problem E

Ruben Spawns (Easy)

Being the head judge of a programming contest is no small thing. As with everything good in life, there is that inevitably long list of things that need to be done before it can happen. Fortunately for Ruben, he has recently acquired a machine that at the push of a button, can spawn a small minion for him to do part of his work. He has also hired an assistant to remind him to activate the machine.

There is one caveat with minions. If you have too many of them, they might get lost. So, simply put, fewer is better. After all, someone has to keep track of those minions. Each minion spawned from the machine can work a set amount of units, and then they are “spent” (there exists a recycling machine, but it is hidden in a deep, dark forest somewhere).

The machine itself creates a given number of minions whose work capacities are normally distributed with both μ and σ unknown. The number of minions the machine can spawn in a given time interval is Poisson distributed with intensity λ , also unknown.

What we are interested in is knowing the minimum amount of times Ruben would have to spawn a minion to be sure that all the work gets done. You are given a list of how many work units each of the N minions can work for. The machine will break down completely after having spawned N minions. The machine lets you choose which minion you want spawn next from the list of possible minions, but you can only spawn each once. All minions are unique in their own ways, but they might still have the same work capacity.



Input specifications

The first line of the input consists of a single integer T , the number of test cases. Each of the following T cases then consist of two lines. The first line has two integers: W , the number of work units Ruben needs completed, and M , the number of minions the machine can spawn. Then follows a line with M integers C_i , representing how many work units each minion can complete.

Output specifications

Output the minimum number of minions needed to complete the workload W , or output “no rest for Ruben” (without the quotes). Please note that you need a capital r in Ruben’s name.

Notes and Constraints

- $0 < T \leq 50$
- $0 < W \leq 10000$
- $0 < M \leq 100$
- $0 < C_i \leq 100$

Sample input

```
4
4 5
1 2 4 100 3
10 10
1 1 1 1 1 1 1 1 1 1
20 10
9 1 1 1 1 1 1 1 1 9
100 5
81 1 2 1 4
```

Output for sample input

```
1
10
4
no rest for Ruben
```