Problem 3 - Scheduler (scheduler)

You helped evaluate the final rankings. You helped choose the servers to rent. We're now almost ready to go.

As someone once didn't one say: *"With great **computing power** comes great responsibility"*.

All the available servers are not useful if we can't find a way to efficiently schedule the tasks to be performed.

Unfortunately, we noticed all the N rented servers are different from the others. In particular, each server takes $P[i]$ unit of time to power on and, once the startup is complete, manages to complete a task every $S[i]$ unit of time (for simplicity we assume all the tasks can be completed in the same amount of time for a server).

However, you will not be able to use all N servers simultaneously (as they will also be used for other jobs). Therefore, given M tasks you could choose a maximum of K servers to help you.

Please help the Reply Code Masters once again to find the minimum time required to execute all the tasks.

Input data

The first line of the input file contains an integer **T**, the number of test cases to solve, followed by **T** testcases, numbered from **1** to **T**.

In each test case, the first line contains the three integers **N**, **K** and **M**: the number of available servers, the maximum number of servers to use and the number of tasks.

The following **N** lines contain two integers: $\mathbf{P}_{in}$[i], $\mathbf{S}_{in}$[i], the time for the server to power-on, and the speed it completes a task in.

Output data

The output file must contain **T** lines. For each test case in the input file, the output file must contain a line with the words:

Case #t: c

where t is the test case number (from **1** to **T**) and c is the minimum time to execute all the tasks.

Constraints

- $1 \leq \mathbf{T} \leq 20$.
- $1 \leq \mathbf{N} \leq 100\,000$.
- $1 \leq \mathbf{K} < \mathrm{N}$.
- $1 \leq \mathbf{M} \leq 1\,000\,000\,000$.
- $1 \leq \mathbf{P[i]} \leq 100\,000$ for each $i = 0 \ldots N-1$.
- $1 \leq \mathbf{S[i]} \leq 100\,000$ for each $i = 0 \ldots N-1$.

Scoring

- **input 1** : $\mathbf{T} = 1$, $\mathbf{N} \leq 5$, $\mathbf{M} \leq 5$, $\mathbf{P}[\mathbf{i}]$ and $\mathbf{S}[\mathbf{i}] \leq 5$.
- **input 2** : $\mathbf{T} = 5$, $\mathbf{N} \leq 100$, $\mathbf{M} \leq 10$, $\mathbf{P}[\mathbf{i}]$ and $\mathbf{S}[\mathbf{i}] \leq 10$.
- **input 3** : $\mathbf{T} = 10$, $\mathbf{N} \leq 1\,000$, $\mathbf{M} \leq 1\,000$, $\mathbf{P}[\mathbf{i}]$ and $\mathbf{S}[\mathbf{i}] \leq 100$.
- **input 4** : $\mathbf{T} = 15$, $\mathbf{N} \leq 10\,000$, $\mathbf{M} \leq 1\,000\,000$, $\mathbf{P}[\mathbf{i}]$ and $\mathbf{S}[\mathbf{i}] \leq 1\,000$.
- **input 5** : $\mathbf{T} = 20$, $\mathbf{N} \leq 100\,000$, $\mathbf{M} \leq 1\,000\,000\,000$, $\mathbf{P}[\mathbf{i}]$ and $\mathbf{S}[\mathbf{i}] \leq 100\,000$.

Examples

input	output

`1`
`3 2 10`
`1 3`
`2 2`
`3 2`
	`Case #1: 13`

Explanation

We can use the second and the third server to execute 5 tasks each.

The second server takes *startup* + 5 × *speed* $= 2 + 5 \times 2 = 12$.
The third server takes *startup* + 5 × *speed* $= 3 + 5 \times 2 = 13$.

The total time is thus $\max(12, 13) = 13$ unit of time.
This solution is optimal as there is no way to complete 10 tasks in less than 13 unit of time.