Challenges Week 5

SOBCH - 25 points

The CEO of SOBCH was extremely impressed with the way you helped them with their FACTORY problem in your first week. So he is asking for your help again!

This time around, the CEO wants to determine the best employee from a list of employees dependent on the amount of work they did. And all of SOBCH engineers are clueless on how to achieve this, so you will write them a program that will achieve this.

Figure 1: Again, please note that this logo is completely original

Input: The first line contains two integers n and p. n is the number of records, and p is the number of employees The next n lines contain a string s, which represents the name of the employee (all lowercase letters) and an integer c, which represents the amount of work they did for that one record. This is followed by a line with one integer m. Following, there are m lines of strings, representing the names employees the CEO is interested in knowing their total amount of work.

Output: the sum of the total amount of worked hours of the requested employees. Followed by the name of the employee who worked the most and their amount of work
Constraints:
$0 \leq n \leq 10^7$
$0 \leq p \leq 10^6$
$0 < length(s) < 50$
$0 \leq c \leq 10^4$
$0 \leq m \leq 10^7$
$0 \leq a \leq 10^9$

Example

Input	Output
11 7 mohammad 200 channa 5 mohammad 72 mralshack 69 anrold 0 erblin 16 anrold 0 alex 40 channa 10 erblin 200 niels 39 6 anrold mohammad channa mralshack erblin alex	612 mohammad: 272

COVID - 45 points

A new COVID version just came out! Scientific research has confirmed that getting infected by it turns you into a zombie :o

The chance of you becoming a zombie after getting infected is 100%, therefore it is extremely important to get your 70th vaccine shot. Thankfully, you have to finish your 4 assignments of AADS and can't go outside to get infected. So you will write a program to simulate the latest version of COVID from the comfort of your gaming chair. Your simulation will help the virus form the biggest possible army of zombies across a $2D$ space.

You have the $2D$ space (grid), $n \times m$ where each cell could represent a **country** or an **ocean**. The virus will start at the top left corner and can only move east or south from that point.

Each country cell can have x people in it. These people could either get infected (they will be represented as a positive number, say 10), or could be already vaccinated (the vaccinated ones will be represented as a negative number, say -3). Once the virus and its army move into a cell with people that can get infected, they infect them (the number of newly infected people gets added to your zombie army). Otherwise, the zombie army will have to fight the people that can't get infected. In that case, every immune person takes down one soldier of your zombie army and dies in the process. If you just have a one-man army, you should avoid entering an immune territory because you will be killed in battle. Finally, you must leave one of your zombie soldiers behind to guard the old land before going into the new one.

You can't traverse the ocean because your zombie army lacks boats. A number 0 indicates an ocean on a map.

Your goal is to reach the lower right corner, and when you do, you want your zombie army to be as massive as possible. Write a program that prints the maximum number of soldiers you could have in your COVID army when you reach the lower right corner using the contents of the War grid. It is guaranteed that the map allows you to reach your destination, with an army size of at least 1.

Input: The first line will contain two integers indicating the grid size, n and m. The next n lines will contain m signed integers each, w_{ij}, representing the size of an army, or ocean, in a cell (i, j).

Output: A single integer representing the maximum number of zombies in your army, after reaching the lower right corner.

Constraints:

$0 < n \leq 10^4$
$0 < m \leq 10^4$
$-10^5 \leq w_{ij} \leq 10^5$ for any indices i and j on the map
$w_{00} > 0$, and also $w_{n-1m-1} \neq 0$ (if we take the map grid to be 0-indexed)
There will always be a path over countries from the upper left cell to the lower right cell, and there will always be a path which leaves you with a final army of at least 1 warrior.

Example

Input	Output
4 4 10 -3 -2 3 -6 0 -1 0 2 -1 1 1 0 0 3 -1	2