

Introduction:

Collaborating as a team can make work much more enjoyable and productive. By pooling our unique skills, we can enhance creativity and efficiency, leading to greater overall output. Equally important is the opportunity for shared learning and experience when working together.

In this project, your task will be to assign contributors to various project roles that align with their skill sets, with the goal of optimizing the completion of projects and achieving a high score. The contributors have a range of already mastered skills, while the projects require different skill sets. Additionally, contributors can enhance their skills by successfully completing projects and can help mentor each other to excel in roles that may initially be challenging. Your challenge is to strategically assign contributors to roles that will utilize their strengths and ultimately maximize the number of completed projects and the associated score.

Problem description:

- Contributors:

A group of N contributors is defined, with each contributor having a name and at least one skill that has been assigned a specific level, ranging from 0 to higher values. If a contributor does not possess a particular skill, it is assumed that their skill level for that skill is 0.

Example:

- ★ Jihane: Haskell level 7
- ★ Achraf: Java level 5, Kotlin level 7
- ★ Ayman: C level 10, CSS level 2, HTML level 5

- Projects:

There are M projects, each with a name, a duration in days, a score for completion, and a "best before" time in days. The "best before" time indicates the last day of work for the project, and if work is completed before that day, the project earns the full score. If work continues after the "best before" day, the score decreases by one point per day, but never goes below zero. Each project requires contributors to fill specific roles, with each role requiring a specific skill level. Each contributor **can fill only one role on a single project**.

For example, a project called "Intranet" could have the following roles:

- Role 0 requiring NodeJs level 5
- Role 1 requiring Javascript level 6
- Role 2 requiring React level 9

- Roles and Mentorship:

Each contributor can be assigned to a project for a specific role, but only if they possess the required skill level or higher. Alternatively, a contributor can be assigned to a role requiring a skill level that is exactly one level higher than their current skill level, provided that another contributor on the same project has the required skill level or higher and can mentor them. A single contributor can mentor multiple team members for the same skill and can also be mentored by others at the same time. If a contributor does not possess a required skill, it is assumed they have a skill level of 0 for that skill. Therefore, a contributor could work on a project and be assigned to a role requiring Python level 1 even if they have no knowledge of Python as long as another team member knows Python at level 1 or higher and can mentor them.

- Assignment:

Each contributor can begin working on day 0 and can work on a maximum of one project at a time. Once work begins on a project, the contributors assigned to that project will work on it for a duration equal to the project's length in days, and then become available to work on other projects.

- Learning:

Completing a project is a valuable learning opportunity, especially for contributors who are pushing the boundaries of their existing abilities. When a project is completed, contributors who worked in roles requiring **a skill level equal to or higher than their current level** will improve their skill level by one level. Other contributors will maintain their current skill level. It's important to note that mentoring someone does not result in an increase in skill level for the mentor.

Input File:

- File Format:

Each input dataset is provided in a plain text file with lines ending in a single '\n' character, containing only ASCII characters. The first line of the dataset consists of two integers separated by a space character:

C: the number of contributors ($1 \leq C \leq 100000$).

P: the number of projects ($1 \leq P \leq 100000$).

This is followed by C sections, each describing an individual contributor. Each contributor is described by the following lines:

The first line contains the contributor's name, an ASCII string of at most 20 characters consisting of uppercase or lowercase English letters (a-z and A-Z) or digits (0-9).

The second line contains an integer N ($1 \leq N \leq 100$), indicating the number of skills the contributor possesses.

The next N lines describe each individual skill of the contributor, each consisting of:

The name of the skill, an ASCII string of at most 20 characters consisting of uppercase or lowercase English letters (a-z and A-Z), digits (0-9), dashes (-), or pluses (+).

An integer L_i ($1 \leq L_i \leq 10$), indicating the level of proficiency the contributor has in that skill.

Following the sections for individual contributors, there are P sections, each describing an individual project. Each project is described by the following lines:

The first line contains:

The name of the project, an ASCII string of at most 20 characters consisting of uppercase or lowercase English letters (a-z and A-Z) or digits (0-9).

An integer D_i ($1 \leq D_i \leq 100000$), indicating the number of days required to complete the project.

An integer S_i ($1 \leq S_i \leq 100000$), indicating the score awarded for completing the project.

An integer B_i ($1 \leq B_i \leq 100000$), indicating the "best before" day for the project.

An integer R_i ($1 \leq R_i \leq 100$), indicating the number of roles required for the project.

The next R_i lines describe each required skill for the project, each consisting of:

The name of the skill, an ASCII string of at most 20 characters consisting of uppercase or lowercase English letters (a-z and A-Z), digits (0-9), dashes (-), or pluses (+).

An integer L_k ($1 \leq L_k \leq 100$), indicating the required level of proficiency for that skill.

Example:	
3 3	3 contributors, 3 projects
Fatima 1 Java 2	contributor Fatima has Java skill at level 2
Ahmed 2 Python 5 Kotlin 5	contributor Ahmed has Python skill at level 5 has Kotlin skill at level 5
Karim 1 Ruby 3	contributor Karim has Ruby skill at level 3
BlogApp 5 10 5 1 Java 3	project BlogApp needs 1 contributor that needs to have Java at level ≥ 3 (2 with mentoring)
Ecommerce 7 10 7 2 Python 3 Java 2	project Ecommerce needs 2 contributors first contributor needs to have Kotlin at level ≥ 3 (2 with mentoring) second contributor needs to have Python at level ≥ 2 (1 with mentoring)
MobileApp 10 20 20 2 Ruby 3 Python 3	project MobileApp needs 2 contributors first contributor needs to have Ruby at level ≥ 3 (2 with mentoring) second contributor needs to have ReactNative at level ≥ 3 (2 with mentoring)

Submissions:

The submission file should be a plaintext file containing only ASCII characters.

- File format:

Your submission describes which projects each contributor is going to work on and in which role.

The first line should contain the integer E ($0 \leq E \leq P$) the number of executed projects.

This should be followed by E sections each describing one completed project. Each project should be described by two lines:

A single line containing the name of the project (as it appears in the input file). Each project can be mentioned at most once in the submission file.

A single line containing the names of the contributors assigned to each of the project roles, separated by single spaces and listed in the same order as the roles appear in the input file.

Submission file:

3

Ecommerce

Ahmed Fatima

BlogApp

Fatima

MobileApp

Karim Ahmed

Scoring:

Each contributor can only work on one project at a time. If one contributor is assigned to multiple projects, the contributor will work on them in the same order as they appear in the **submission** file. Each project starts immediately on the first day on which all the assigned contributors are available.

Projects execution timeline based on the input data set and the submission from the previous sections.

If some project assignment is invalid because the assigned contributor does not have the required skill level for the project after finishing all previously assigned projects, the submission is considered invalid and will not be scored.

Each project that is completed successfully receives its assigned score, as defined in the input file, minus penalty points for any delay. If a project is completed after its “best before” time, it gets one point less for each day it is late (but no less than zero points). Note that even if a project scores zero points, the assigned contributors will work on it (and may improve their skills thanks to it).

The total score is the sum of scores for all correctly completed projects.

The example submission results in this timeline:

Day 0 to day 6: Ahmed and Fatima are working on project Ecommerce (they both have the skills required).

As of project completion, Fatima levels up in Java: level 2 → 3;

Ahmed doesn't level up because his Python skill (level 5) is of a higher level than required for the project (level 3).

Project Ecommerce's last day of work is day 6, so it completes strictly before its "best before" day 7 and receives the full score: 10 points.

Day 7 to 11: Fatima is working on the project BlogApp(she has sufficient Java skill after completing project Ecommerce).

As of project completion, Anna levels up in Java: level 3 \rightarrow 4;

Project BlogApp's last day of work is 11 (so it's completed strictly before day 12), while its "best before" day was 5. It is late by $(12-5=7)$ days and receives a score of: $(10-7=3)$ points.

Day 7 to 16: Karim and Ahmed are working on project MobileApp

As of project completion, Karim levels up in Python: level 3 \rightarrow 4;

Bob doesn't level up in Python because his skills are of a higher level than required for the project (Python level 5, required 3)

Project MobileApp's last day of work is day 16, while the "best before" day is 20, so it receives the full score: 20 points.

In the end, projects Ecommerce (10 points), BlogApp (3 points) and MobileApp (20 points) are completed, resulting in a total score of 33 points.