Problem R
Movie Collaborators
Problem ID: collaborators

Your solution to this problem must be implemented using an object-oriented approach by defining the classes presented below. Any other solution will receive zero points, even if the testing system judges it correctly.

The Internet Movie Database (IMDb) is a popular website containing detailed information about movies, going all the way back to the first days of cinema. While most people interact with IMDb through their website, IMDb also publishes abridged datasets that anyone can download and analyze.

We would like to use this data to find the frequent collaborators of a given person. For example, some directors tend to use the same performers, cinematographers, composers, etc. across many movies. So, given a number $N$, it would be interesting to find out who has collaborated with that director in at least $N$ movies.

For example, suppose we had three movies Jumanji, Titanic, and Avatar, with the following cast/crew associated with each movie:

- Jumanji: Sam, Alex, Billy
- Titanic: Sam, Alex, Jo
- Avatar: Sam, Billy

If we focus on Sam’s collaborators, and specifically on $N = 2$, we can see that Alex and Billy have each collaborated with Sam on two movies (not necessarily the same movies). If, on the other hand we were to look at $N = 3$, we would see that no one has worked with Sam on three movies. For simplicity, we are not going to keep track of the exact job a person did on a movie; we will only care about whether a person was involved in a movie (and not whether they were a director, a performer, etc.). Keep in mind that there are movies with the same titles (and similarly there are people with the same names). We will associate a unique identifier to help determine whether two movies are equal or distinct (and similarly for people).

You must implement the following class definitions along with the specified constructor, attributes (also known as instance variables) and methods. You must think about the type for each attribute and the type signature for each method. For this problem, the Input section helps clarify these types. Implement the following classes:

A class named Movie that models a single movie.

- **Constructor**
  - Implement a constructor that initializes a Movie instance and takes in three arguments: a unique identifier for the movie, the movie’s title, and the year the movie was released.

- **Attribute(s)**
  - Define an attribute that represents the unique identifier for the movie.
– Define an attribute that represents the title for the movie.
– Define an attribute that represents the year the movie was released.
– Define an attribute that keeps track of the number of people involved in the movie.

• Method(s)
  – Implement a method called add_person in the Movie class that takes in a Person object and adds them to the attribute that keeps track of the people in the movie.

A class named Person that models a person in a movie.

• Constructor
  – Implement a constructor that initializes a Person instance and takes in three arguments: a unique identifier for the person, the person’s name, and their year of birth.

• Attribute(s)
  – Define an attribute that represents the unique identifier for the person.
  – Define an attribute that represents the name of the person.
  – Define an attribute that represents the person’s year of birth.
  – Define an attribute that keeps track of the movies this person has been involved in.

• Method(s)
  – Implement a method called add_movie in the Person class that takes in a Movie object and adds them to the attribute keeps track of the movies this person has been involved in.

You may write additional helper methods and attributes; however, the above class definitions, constructor, attributes, and methods are required to be implemented and used in your solution.

As a remainder, your solution to this problem must be implemented using an object-oriented approach by defining the classes presented above. Any other solution will receive zero points, even if the testing system judges it correctly.

Input

The first input line is two integers separated by a single space; The first integer is $M$ ($1 \leq M \leq 100$), which represents the number of movies in the database. The second integer is $P$ ($1 \leq P \leq 50$) that represents the number of people in the database.

The second input line is two integers separated by a single space; This line represents the person you are trying to find collaborators with. The first integer is $T$ ($1 \leq T \leq P$), which is a unique identifier for a specific person. The second integer is $N$ ($1 \leq N \leq 50$), as defined above.

The input is followed by $M$ lines, each corresponding to a single movie. A movie line contains three components, each separated by a single space. The first component is the movie name. Each movie name contains only lowercase letters, no whitespace, and has a maximum length of 50 characters. The second component is an integer $Y$ ($1900 \leq Y \leq 2022$) that represents the year the movie was released. The third component is $I$ ($1 \leq I \leq M$), the unique identifier for the movie.
The remaining input is $P$ lines, each corresponding to a single person. Each person line contains the person’s name, an integer $D$ ($1 \leq D \leq P$) (the unique identifier for the person), their birth year $Y$, followed by $C$ integers. Each line component is separated by a single space. The $C$ integers are the unique movie identifiers that the person is involved with.

Output

The output is $K$ lines that represent each person that has collaborated on at least $N$ movies with person $T$. Each $K$ line outputs the collaborator’s unique identifier, name, and birth year. Each component is separated by a single space. The $K$ lines are ordered in ascending order by the unique identifier of the collaborator.

If no person has collaborated with person $T$ then output `NONE`

<table>
<thead>
<tr>
<th>Sample Input 1</th>
<th>Sample Output 1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Sample Input 2</th>
<th>Sample Output 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 4 1 3 jumanji 1995 1 titanic 1997 3 avatar 2009 2 sam 1 1974 1 3 2 billy 4 1980 1 2 alex 3 1988 3 1 jo 5 1964 3</td>
<td>NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample Input 3</th>
<th>Sample Output 3</th>
</tr>
</thead>
</table>