SHOPEE CODE LEAGUE 2020

Shopee Programming Contest #1

Competition Details

You may **start the competition anytime between 1pm (GMT+7) / 2pm (GMT+8) to 2:40pm (GMT+7) / 3:40pm (GMT+8).** Once you start the competition, the **countdown will begin** and **you will have 3 hours 15min to submit your codes.**

Duration: 3 Hours 15 Minutes (Additional 15 minutes is given for your team to familiarise with the competition platform)

Table of Contents

Sample output 17

Item Stock

Problem statement

Items in Shopee can have their stocks derived from other items. For example, 1 stock of item A can be derived from 2 stock of item B + 3 stock of item C. We say that item B and item C are parents of item A. For this problem, we are only interested when an item can only have 1 parent item. In this case, we can see the structure of stock derivation will form a rooted tree.

There are 2 kinds of derivations:

1. Dynamic stock derivation. Suppose that 1 stock of item A equals to Qty stock of item B. Then, the stock of item A will be equal to floor(item_B_stock / Qty).
2. Fixed stock derivation. Suppose that 1 stock of item A equals to Qty stock of item B, and we initially have S stock of item A. Then, item A will deduct stocks from its lowest ancestor which is fixed stock, to make sure that item A will have sufficient stock. It can be assumed that the root of the tree (1st item) will always be fixed stock. Note that the number of reserved stocks depends on the multiplication of the Qty from the path of item A to that ancestor, not just the Qty to item B. Please refer to the example input for clarity.

At first, we only have item 1, which initially has **M** stock. Then, we add **N-1** items one-by-one, possibly changing the stock of some items at each step. In the end, what will be the stock of each item?

Input

The first line contains 2 integers **N** ($1 \le \mathbf{N} \le 100{,}000$) and M ($1 \le \mathbf{M} \le 1{,}000{,}000{,}000$), denoting the number of items and the initial stock of the 1st item.

The next **N-1** lines contain the description of the i-th item (starting from 2), which can be in one of the 2 following formats:

1. **P**i **Qty**i ($1 \le \mathbf{P}i < i$, $1 \le \mathbf{Qty}i \le 10$), which means the i-th item has dynamic stock with parent item Pi and 1 stock of it equals to Qtyi stock of its parent
2. **P**i **Qty**i **S**i ($1 \le \mathbf{P}i < i$, $1 \le \mathbf{Qty}i \le 10$, $1 \le \mathbf{S}i \le 1{,}000{,}000{,}000$), which means the i-th item has fixed stock with parent item **P**i, 1 stock of it equals to **Qty**i stock of its parent, and has initial stock of **S**i.

It is guaranteed that at the end, the stock for each item will be non-negative.

Output

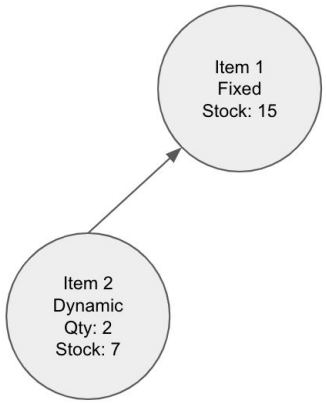
Output N lines, each containing an integer. The integer in the i-th line denotes the stock of the i-th item.

Sample explanation

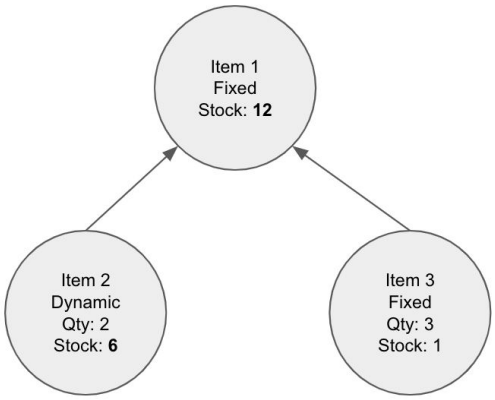
Below are the states after each item additions:

1. Initial state

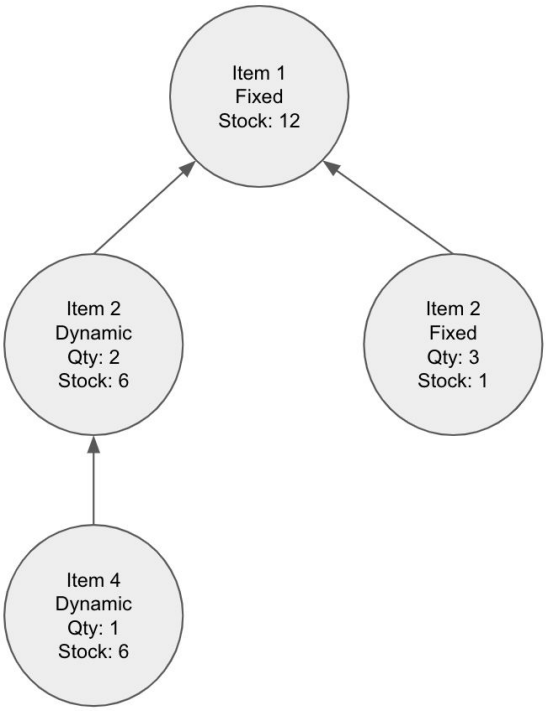
2. Adding 2nd item, stock is floor(15/2) = 7



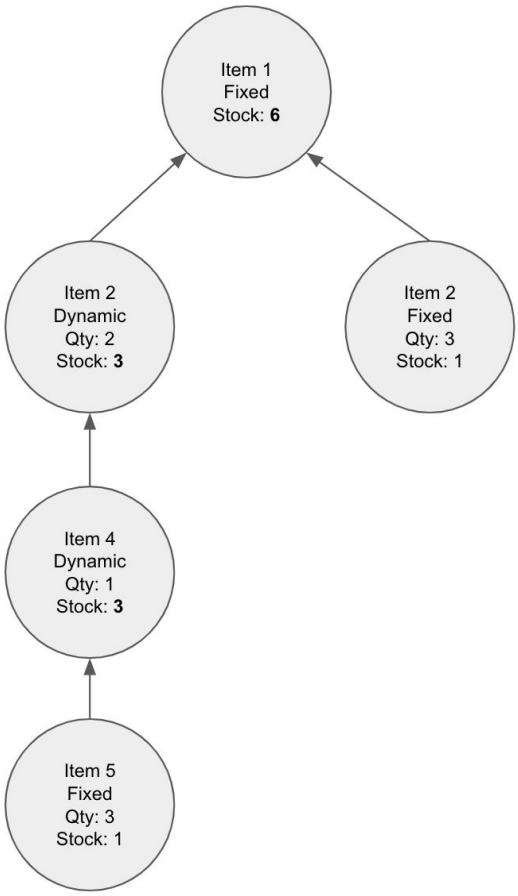
3. Adding 3rd item, taking 1 * 3 stock from the 1st item. Note that Item 2 stock is also changed because of this.



4. Adding 4th item, stock is floor(6/1) = 6



5. Adding 5th item, taking 2*1*3 (Qty) *1 (stock) stock from the 1st item as it is its lowest fixed stock ancestor. Note that Item 2 and item 4 stock are also changed because of this.



Sample input

5 15

1 1 2

2 1 3 1

1 2 1

2 4 3 1

Sample output

6

3

1

3

1

Search Engine

Problem statement

Who doesn't like to search and see these unexpected search suggestions floating just below the search bar. Everyone likes it!!! As we all know Shopee, one of the largest E-commerce platforms, also has a search bar where users can search for all kinds of items. Shopee wants to build a new search engine. And you are to help Shopee to implement this new engine.

You are given a data set that contains all the item's names, and an item's name is represented as an ordered sequence of strings separated by a single space and the strings contain only lowercase English alphabets(**a-z**) and digits(**0-9**). for example, a valid name could be, **"apple iphone se 2**". Queries for the new search engine will be a sequence of alphanumeric strings separated by space. For example, "**se 2**" or "**11 pro max**" and the search engine has to answer how many different items are there in the data set containing the query sequence in their name in exact order. For example, "**se 2**" matches the item "**apple iphone se 2**", however "**app**" doesn't match this item.

Input

Input starts with an integer **T (1 ≤ T ≤ 15)**, denoting the number of test cases. The first line of each test case will contain two integers **N (1 ≤ N ≤ 104)** and **Q (1 ≤ Q ≤ 104)**. Here, **N** is the number of items in the database and **Q** is the total number of queries. Each of the next **N** lines will contain an item's name as described. Each of the next **Q** lines will contain a search query as described. You can safely assume that each item's name will contain at most 10 spaces and the total length will be between 1 to 50.

Output

For each case, print the case number in a single line. Then for each query **Q** print the number of different names in the database who contains the query sequence in their name in exact order.

Constraints

Total number of characters in the dataset will be not more than 7×105

Sample explanation

For the first test case, both "limes avocado" and "apple lettuce" match both 1st and 3rd items, "limes" and "apple" match in all three items, "app" doesn't match any item and "apple limes" matches the second item.

Sample Input

2

3 6

apple lettuce limes avocado

onion cranberries apple limes

escarole corn28corn apple lettuce limes avocado

limes avocado

apple lettuce

limes

apple

app

apple limes

3 3

apple iphone se 2

iphone 11 max pro

iphone 11 pro max

apple iphone

max pro

Iphone

Sample Output

Case 1:

2

2

3

3

0

1

Case 2:

1

1

3

Judging Servers

Problem statement

As we all know, you are the chief judge for the upcoming Shopee Code League hosted by our favorite E-Commerce platform Shopee. You have already selected **N** problems from hundreds of thousand of problems from your quality Problem Bank. Since you want to make every contestant happy even if he/she got Wrong Answers on every problem during the contest, you have decided to judge each problem on a different server.

Now you have to buy **N** judging servers from SEA Server Limited which is a reputed company. SEA Server Limited has total S servers in a row numbered from **1 to S** and you have to choose **N** servers from these **S** servers. The **i'th** server has a price tag of **P**i where **1 ≤ i ≤ S**. You cannot rent these servers on an hour or day basis. However, SEA Server Limited has a lifetime offer for you. If you buy any server then you get one of the adjacent servers for free if you wish. If you choose to buy the **i'th** server then you can get the **(i-1)'th** or **(i+1)'th** server for free if you want to take it for free. The contest date has a tag of coming soon and the contest organizers want to know the total cost for the problem set and judging servers from you. Since you are the ultimate chief judge who wants to maximize your profit and as well as make every contestant happy. You have to choose **N** servers with lowest cost possible to maximize your profit.

Input

Input starts with an integer **T (1 ≤ T ≤ 50)**, denoting the number of test cases.

Each case starts with two integers **S (1 ≤ S ≤ 1000)** and **N (1 ≤ N ≤ S)**. Next line contains **S** integers separated by space and the **i'th** integer of this line represents the price tag **P**i of the **i'th** server **(0 ≤ Pi ≤ 109)**.

Output

For each case, print the case number and the minimum cost to buy the **N** servers.

Sample explanation

In the second case, you can pay for the 3rd and the 4th servers with a cost of 115 and take the 2nd or 5th server for free.

Sample input

2

3 2

15 14 15

5 3

1000 560 30 85 100 900

Sample output

Case 1: 14

Case 2: 115

Lucky Winner

Problem statement

Ding Ding Ding! You have been chosen to be the one and only winner of SHOPEE LUCK LEAGUE.

We're giving you **K tokens** to pick items on our shopee search results page for FREE!

Each token can get two adjacent items on the grid (horizontally or vertically).

Given that shopee search results page is a grid of **N rows and 3 columns** (actually it's 5, but we specially change our UI to make your life easier). Each item on the grid has a price (the price can be negative).

The rule are you have to use **all of your tokens**, one item can only be covered by one token (no overlapping here, you can't buy one item twice). Your goal is to find the maximum worth of items you can bring home.

Input

First line, number N (number of rows) (1 <=N <= 1000), and K (number of tokens) (1 <= K <= 1000)

N next lines, each line contain 3 numbers, the values of the board (abs(a[i,j]) <= 1e6)

Output

A single number of the maximum worth of items that can be cover with exactly K non-overlapping tokens

Sample explanation

Second example: It's optimal to use 3 tokens on [100, -1], [2, 5], and [3,4]

Sample input

5 3

100 -9 -1

-1 3 2

-9 2 3

2 5 1

3 3 4

Sample output

113

Sequences

Problem statement

You are on a company visit to Shopee. During the office tour, you noticed that there seems to be a random scribbling on one of the walls. After looking at it closely, you noticed it is actually an algorithm question! Below is the question:

You are given **N** functions f(i, j) with parameters **A**i, **B**i, **C**i, where the value of f(i, j) is equal to **A**i x j2 + **B**i for each 1 ≤ j ≤ **C**i. Find how many sequences (i1, j1), (i2, j2), … , (iM, jM) of length **M** are there in which the following holds:

f(i1, j1) + f(i2, j2) + … + f(i**M**, j**M**) is divisible by **K**

Two sequences are different if there is at least one index k, such that ik ≠ ik' or jk ≠ jk'

You quickly take note of the question, as maybe it is a draft for an interview question. Solve the question to increase your chance of acing the future interview at Shopee!

Input

The first line contains 3 integers **N** (1 ≤ **N** ≤ 5,000), **M** (1 ≤ **M** ≤ 1,000,000,000), and **K** (1 ≤ **K** ≤ 2,000).

The next **N** lines each contains 3 integers **A**i, **B**i, (0 ≤ **A**i, **B**i < **K**) and **C**i (1 ≤ **C**i ≤ 1,000,000,000), denoting the parameters for the i-th function.

Output

One line containing a single integer, the number of the sequence. Since this number can be very large, output its value modulo $10^9 + 7$.

Sample explanation

Below are all the possible sequences:

1. (1, 1), (1, 1)
2. (1, 1), (1, 2)
3. (1, 1), (2, 1)
4. (1, 2), (1, 1)
5. (1, 2), (1, 2)
6. (1, 2), (2, 1)
7. (2, 1), (1, 1)
8. (2, 1), (1, 2)
9. (2, 1), (2, 1)
10. (2, 2), (2, 2)
11. (2, 3), (3, 1)
12. (3, 1), (2, 3)

Sample input

3 2 6

0 3 2

1 2 3

2 5 1

Sample output

12