Power Transmission

The Problem

DESA is taking a new project to transfer power. Power is generated by the newly established plant in Barisal. The main aim of this project is to transfer Power in Dhaka. As Dhaka is a megacity with almost 10 million people DESA wants to transfer maximum amount of power through the network. But as always occurs in case of power transmission it is tough to resist loss. So they want to use some regulators whose main aim are to divert power through several outlets without any loss.

Each such regulator has different capacity. It means if a regulator gets 100 unit power and it's capacity is 80 unit then remaining 20 unit power will be lost. Moreover each unidirectional link(Connectors among regulators) has a certain capacity. A link with capacity 20 unit cannot transfer power more than 20 unit. Each regulator can distribute the input power among the outgoing links so that no link capacity is overflown. DESA wants to know the maximum amount of power which can be transmitted throughout the network so that no power loss occurs. That is the job you have to do.

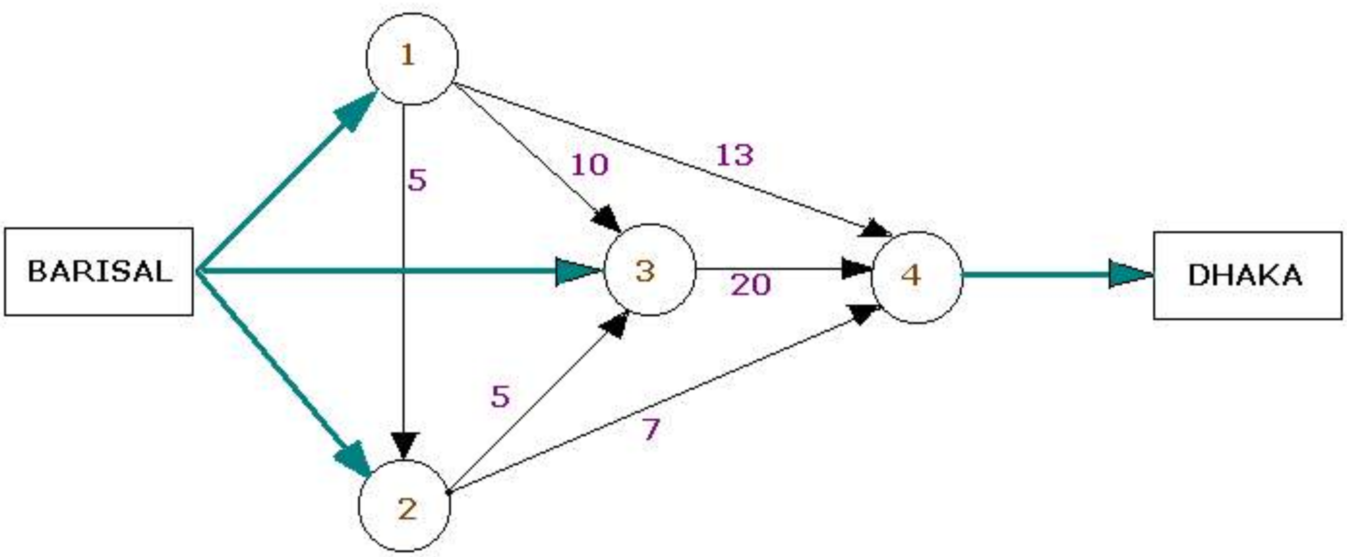


Figure : Sample Input - 1

(Do not try to mix the above description with the real power transmission.)

The Input

The input will start with a postive integer N (1<=N<=100) indicates the number of regulators. The next few lines contain N positive integers indicating the capacity of each regulator from 1 to N. The next line contains another positive integer M which is the number of links available among the regulators. Following M lines contain 3 positive integers (i j C) each. 'i' and 'j' is the regulator index (1<=i,j<=N) and C is the capacity of the link. Power can transfer from i'th regulator to j'th regulator. The next line contains another two positive integers B and D. B is the number of regulators which are the entry point of the network. Power generated in Barisal must enter in the network through these entry points. Simmilarly D is the number of regulators connected to Dhaka. These links are special and have infinite capacity. Next line will contain B+D integers each of which is an index of regulator. The first B integers are the index of regulators connected with Barisal. Regulators connected with Barisal are not connected with Dhaka.

Input is terminated by EOF.

The Output

For each test case show the maximum amount of power which can be transferred to Dhaka from Barisal. Use a seperate line for each test case.

Sample Input

```
4
10 20 30 40
6
1 2 5
1 3 10
1 4 13
2 3 5
2 4 7
3 4 20
3 1
1 2 3 4
2
50 100
1
1 2 100
1 1
1 2
```

Sample Output

```
37
50
```

Md. Kamruzzaman

Sabotage

The regime of a small but wealthy dictatorship has been abruptly overthrown by an unexpected rebellion. Because of the enormous disturbances this is causing in world economy, an imperialist military super power has decided to invade the country and reinstall the old regime.

For this operation to be successful, communication between the capital and the largest city must be completely cut. This is a difficult task, since all cities in the country are connected by a computer network using the Internet Protocol, which allows messages to take any path through the network. Because of this, the network must be completely split in two parts, with the capital in one part and the largest city in the other, and with no connections between the parts.

There are large differences in the costs of sabotaging different connections, since some are much more easy to get to than others.

Write a program that, given a network specification and the costs of sabotaging each connection, determines which connections to cut in order to separate the capital and the largest city to the lowest possible cost.

Input

Input file contains several sets of input. The description of each set is given below.

The first line of each set has two integers, separated by a space: First one the number of cities, n in the network, which is at most 50. The second one is the total number of connections, m, at most 500.

The following m lines specify the connections. Each line has three parts separated by spaces: The first two are the cities tied together by that connection (numbers in the range 1 - n). Then follows the cost of cutting the connection (an integer in the range 1 to 40000000). Each pair of cites can appear at most once in this list.

Input is terminated by a case where values of n and m are zero. This case should not be processed. For every input set the capital is city number 1, and the largest city is number 2.

Output

For each set of input you should produce several lines of output. The description of output for each set of input is given below:

The output for each set should be the pairs of cities (i.e. numbers) between which the connection should be cut (in any order), each pair on one line with the numbers separated by a space. If there is more than one solution, any one of them will do.

Print a blank line after the output for each set of input.

Sample Input

```
5 8
1 4 30
1 3 70
5 3 20
4 3 5
4 5 15
5 2 10
3 2 25
2 4 50
5 8
1 4 30
1 3 70
5 3 20
4 3 5
4 5 15
5 2 10
3 2 25
2 4 50
0 0
```

Sample Output

```
4 1
3 4
3 5
3 2

4 1
3 4
3 5
3 2
```

Problem setter: Jesper Larsson, Lund University, Sweden

Collector's Problem

Input: standard input
Output: standard output
Time Limit: 5 seconds

Some candy manufacturers put stickers into candy bar packages. Bob and his friends are collecting these stickers. They all want as many different stickers as possible, but when they buy a candy bar, they don't know which sticker is inside.
It happens that one person has duplicates of a certain sticker. Everyone trades duplicates for stickers he doesn't possess. Since all stickers have the same value, the exchange ratio is always 1:1.
But Bob is clever: he has realized that in some cases it is good for him to trade one of his duplicate stickers for a sticker he already possesses.
Now assume, Bob's friends will only exchange stickers with Bob, and they will give away only duplicate stickers in exchange with different stickers they don't possess.
Can you help Bob and tell him the maximum number of different stickers he can get by trading stickers with his friends?

Input

The first line of input contains the number of cases T ($T<=20$).
The first line of each case contains two integers n and m ($2<=n<=10, 5<=m<=25$). n is the number of people involved (including Bob), and m is the number of different stickers available.
The next n lines describe each person's stickers; the first of these lines describes Bob's stickers.
The i-th of these lines starts with a number $k_i<=50$ indicating how many stickers person i has.
Then follows k_i numbers between 1 and m indicating which stickers person i possesses.

Output

For each case, print the test case number together with the maximum number of different stickers Bob can get.

Sample Input

```
2
2 5
6 1 1 1 1 1
3 1 2 2
3 5
4 1 2 1 1
3 2 2 2
5 1 3 4 4 3
```

Sample Output

```
Case #1: 1
Case #2: 3
```

Explanation of the sample output:

In the first case, no exchange is possible, therefore Bob can have only the sticker with number 1.
In the second case, Bob can exchange a sticker with number 1 against a sticker with number 2 of the second person,
and then this sticker against a sticker with number 3 or 4 of the third person, and now he has stickers 1, 2 and 3 or 1, 2 and 4.

Problem setter: Adrian Kuegel

F

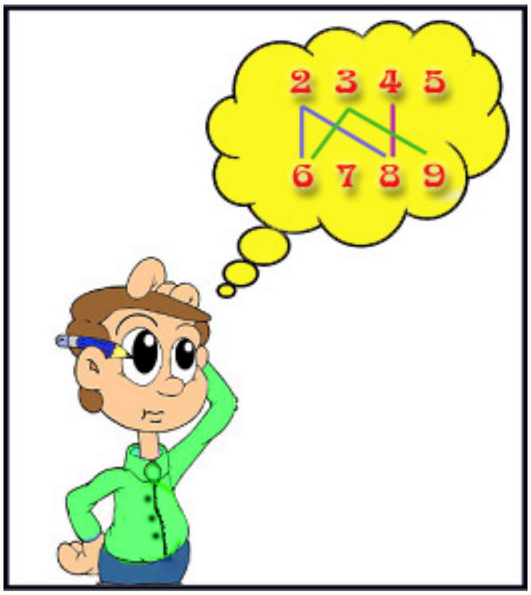
Next Generation Contest 3

Factors and Multiples

Time Limit: 2 seconds

You will be given two sets of integers. Let�s call them set **A** and set **B**. Set **A** contains **n** elements and set **B** contains **m** elements. You have to remove k_1 elements from set **A** and k_2 elements from set **B** so that of the remaining values no integer in set **B** is a multiple of any integer in set **A**. k_1 should be in the range **[0,n]** and k_2 in the range **[0,m]**.
You have to find the value of (k_1+k_2) such that (k_1+k_2) is as low as possible.

P is a multiple of **Q** if there is some integer **K** such that **P = K * Q**.



Suppose set **A** is **{2,3,4,5}** and set **B** is **{6,7,8,9}.** By removing **2** and **3** from **A** and **8** from **B**, we get the sets **{4,5}** and **{6,7,9}**. Here none of the integers **6**, **7** or **9** is a multiple of **4** or **5**.

So for this case the answer is **3** (**2** from set **A** and **1** from set **B**).

Input

The first line of input is an integer **T(T<50)** that determine the number of test cases. Each case consists of two lines. The first line starts with **n** followed by **n** integers. The second line starts with **m** followed by **m** integers. Both **n** and **m** will be in the range **[1,100]**. All the elements of the two sets will fit in **32** bit signed integer.

Output

For each case, output the case number followed by the answer.

Sample Input	Output for Sample Input
2 4 2 3 4 5 4 6 7 8 9 3 100 200 300 1 150	Case 1: 3 Case 2: 0

Problem Setter: Sohel Hafiz
Special Thanks: Jane Alam Jan

Problem B

Buy one, get the rest free.

Time Limit: 3 seconds

"Whoa! It feels like I'm flying!"
Lrrr

It's year 2258, and the age of airplanes is coming to an end. Everyone is using teleporters now. In an effort to stay competitive, the last remaining air travel company, GetsJo, is offering the following deal to its customers. Instead of buying one plane ticket, you can rent a whole flight from A to B. Each flight can carry a certain number of people and costs a certain amount of money. If you do that, then you can rent all of the other flights of equal or lesser cost for free!

For example, if there are 4 flights with costs $10000, $25000, $30000 and $40000, and you rent the $30000 flight, then you get the $10000 and $25000 flights for free. The total cost to rent these 3 flights is $30000.

You want to organize a large programming competition and would like to invite all of the participants to city **n**, where the competition will be held. Being a nice person, you decide to pay for everyone's airplane tickets. Given the locations of the participants and the list of available flights between now and the day of the competition, what is the cost of renting enough flights to get all of the participants to city **n** in the next **d** days?

Input

The first line of input gives the number of cases, **N**. **N** test cases follow. Each one starts with a line containing the number of cities (1<=**n**<=30), the number of days (1<=**d**<=10) until the competition and the number of flights (0<=**m**<=1000). **m** lines follow, each one containing 5 integers: **u**, **v**, **c**, **p** and **e** (1<=**u**,**v**<=**n**, 1<=**c**<=100, 0<=**e**<**d**). This means that a flight that can carry **c** passengers and costs **p** dollars leaves city **u** on day **e** in the evening and arrives next day in the morning to city **v**. Day 0 is today, and all of the participants need to be in city **n** in the evening of day **e**. Finally, **n** integers ($\mathbf{z_1}$, $\mathbf{z_2}$, ..., $\mathbf{z_n}$) follow, meaning that there are $\mathbf{z_i}$ participants in city **i** on day 0 (0<=$\mathbf{z_i}$<=100). The maximum cost of a flight is 100000. There will never be two flights with the same **u**, **v** and **e** values.

Output

For each test case, output one line containing "Case #**x**:" followed by the minimum required cost of flying all of the participants to city **n** before the end of day **d**. If no amount of money is enough, print "Impossible" instead.

Sample Input	**Sample Output**

2 5 4 5 1 5 100 30000 0 2 4 10 10000 0 2 4 10 10000 1 4 5 25 25000 2 2 5 100 40000 3 1 20 0 5 100 2 1 1 1 2 99 10400 0 100 0	Case #1: 30000 Case #2: Impossible

---

Problemsetter: Igor Naverniouk
Alternate solution: Yury Kholondyrev