Algorithms for Programming Contests
WS21 - Week 02

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Philipp Czerner, Martin Helfrich,
Christoph Welzel, Mikhail Raskin

Welcome to our practical course! This problem set is due by

Friday, 05.11.2021, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

This page is intentionally left blank.

Problem A
Marry Rich

Lea has a new game to play: *The Fugger Family IV*. In this game, Lea plays Jakob Fugger "the Rich", one of the richest persons of all time, and tries to grow his family's wealth and influence. At the time that Jakob Fugger lived, 1459 - 1525, one of the main ways to achieve this was marrying into a family having both. In real life, Jakob Fugger married Sibylla Artzt, Grand Burgheress to Augsburg, but Lea wants to do better in that game.

Lea compiled a list of all notable people in the world, the amount of money they have, and a list of all relations and marriages between them. To increase her influence, she wants Jakob to marry somebody who is not already connected to his family via relations or marriages, also via several steps. For example, she does not want Jakob to marry relatives of his uncle's wife. Also, it is not possible to marry someone who is already married.

As the game's publisher Moonflowers does not want to be criticized by activists, Lea may make Jakob marry any person in the game, including males. Among the possible candidates, she wants Jakob to marry the one with the most money. But who is this?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing three space-separated integers a, b and c, where a is the number of notable people in the world labelled from 1 to a with person a being Jakob Fugger, b is the number of family relations and c the number of marriages.

One line follows containing $a - 1$ space-separated integers $m_1, \ldots, m_{a-1}$ where m_i denotes the amount of money person i owns.

b lines follow describing the family relations. Each of these lines contains two space-separated integers d and e meaning that d is related to e. Note that relations that are given as a transition of other relations may be omitted.

c lines follow describing the marriages. Each of these lines contains two space-separated integers f and g meaning that f is married to g. Everybody is married at most once.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the biggest amount of money Lea can get into the family by marrying or "impossible" if there is nobody to marry satisfying her constraints.

Constraints

- $1 \leq t \leq 20$
- $1 \leq a \leq 10^4$
- $0 \leq b, c \leq 10^4$
- $0 \leq m_i \leq 10^6$ for all $1 \leq i \leq a - 1$
- $1 \leq d, e \leq a$
- $1 \leq f, g \leq a - 1$

Sample Input 1

```
2
7 1 1
2 1 3 4 6 5
6 7
3 5

7 2 1
1 2 3 6 4 5
6 7
5 4
4 6
```

Sample Output 1

```
Case #1: 4
Case #2: 3
```

Sample Input 2

```
5
3 1 0
105872 333624
1 1

4 3 0
819471 664525 500888
2 2
2 4
2 1

5 4 0
619991 362808 274506 846462
3 2
3 2
4 1
2 5

4 1 0
303800 2463 655780
3 2

4 1 0
454076 114316 1256
4 2
```

Sample Output 2

```
Case #1: 333624
Case #2: 500888
Case #3: 846462
Case #4: 655780
Case #5: 454076
```

Problem B
Drawing Lots

Lea is organizing a type of lottery where customers can, for a certain cost, draw lots from an urn. Each lot is either winning or losing, with a certain probability, and the customers win prizes if they draw winning lots. However, instead of the standard lottery where they draw one lot and win the prize if this one lot is winning, Lea puts a certain twist on the lottery: there are several prizes, numbered from 1 to n, with increasing value, and each customer draws one lot for the first prize, two lots for the second one, and so on, until they draw n lots for the last prize. They win a prize if and only if all the lots for that prize show up winning. They can win several prizes if the lots for all of these prizes show up winning. All the lots have the same probability of showing winning, and Lea ensures that this probability always stays the same during drawings (e.g. by appropriately replacing lots). There is one fixed cost to draw lots for all prizes.

As Lea is running the lottery, she wants to ensure that she does not make a loss. This means that the expected total payoff, which is the sum of the value of the prizes times the expected winning probability to win that prize, minus the cost to draw lots, should not be positive. Still, she wants the winning probabilty to be as high as possible so the customers stay happy. Lea has already fixed the number of prizes, their values and the cost. Can you tell her what the maximal probability for a lot to show up winning should be such that she does not make a loss?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers n b, where n is the number of prizes and b is the cost to draw lots. A second line follows, containing n space-separated integers $a_1 \dots a_n$, where a_i is the value of prize i.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the maximal winning probability of a lot such that the expected total payoff is less or equal than 0 with an absolute error of up to 10^{-6}. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10$
- $1 \leq b \leq 10000$
- $1 \leq a_i \leq 10000$ for all $1 \leq i \leq n$
- $a_i \leq a_{i+1}$ for all $1 \leq i \leq n - 1$

Sample Explanation

In the first sample, in the first case, there are 3 prizes with values 1, 5 and 10. With a winning probability of 0.5, the expected winning probabilty is 0.5 for the first prize, $0.5^2 = 0.25$ for the second prize and $0.5^3 = 0.125$ for the third prize. The expected payoff is therefore $1 \cdot 0.5 + 5 \cdot 0.25 + 10 \cdot 0.125 = 3$. With cost 3, the total expected payoff is 0.

Sample Input 1

```
5
3 3
1 5 10

5 1
2 3 5 7 11

2 10
1 100

3 10
2 3 5

4 10
1 10 100 10000
```

Sample Output 1

```
Case #1: 0.5000000000
Case #2: 0.2856243953
Case #3: 0.3112672920
Case #4: 1.0000000000
Case #5: 0.1732375136
```

Sample Input 2

```
13
1 3
10

2 1
5 9

3 1
5 8 10

3 9
3 6 8

4 2
1 4 4 7

4 2
4 4 4 5

5 5
1 2 7 7 10

5 9
2 3 5 6 6

6 5
1 1 2 8 8 8

7 7
1 1 1 2 3 3 6

8 4
2 3 4 4 5 7 7 8

9 1
1 2 5 6 8 8 9 10 10

10 3
1 3 3 3 5 6 8 9 10 10
```

Sample Output 2

```
Case #1: 0.3000000000
Case #2: 0.1561249820
Case #3: 0.1544582860
Case #4: 0.7500000000
Case #5: 0.4590767260
Case #6: 0.3347431820
Case #7: 0.6197041759
Case #8: 0.7608172953
Case #9: 0.6652758048
Case #10: 0.8344954285
Case #11: 0.5677976774
Case #12: 0.3554659118
Case #13: 0.5547896661
```

Problem C
Cable Car

Lea is a great fan of wintersports. She always follows the winter olympics on TV and just loves to go skiing herself. This year, she booked a room in an expensive hotel in a very exclusive ski resort called "Slippery Slopes and Hills". One evening at the Après-Ski-Party, she met an interesting man - the architect who planned all the cable cars taking the tourists up the mountain. They talked for a bit and he described his latest problem to her.

The ski resort is trying to build a new cable car up a glacier. Through complicated computation, they even found out exactly how many posts are needed to support the cable car. Since glaciers move (albeit really slowly), the individual posts supporting the cable car have to be spaced as far apart from each other as possible. Additionally, the cable car should also span a canyon in the middle of the route. Now the architect is hard at work, trying to figure how to place the posts. Lea, who is always on the lookout for interesting problems, tells you about it. Can you help the architect?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line of four integers d, p, u, and v, where d is the length of the route (going from 0 to d) of the cable car, p is the amount of posts that should be placed, u is the beginning point of the canyon and v the end point. Posts may be placed anywhere between 0 and d, i.e. exactly on 0, d, u, and v, but not in between u and v.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the maximal minimum distance between two posts that can be achieved with an absolute error of up to 10^{-4}. This means the maximum x such that the architect can place all the posts and no two posts are less than x apart. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq d \leq 10^6$
- $2 \leq p \leq 2 \cdot 10^6$
- $0 \leq u \leq v \leq d$

Sample Input 1	Sample Output 1
4 2 3 1 2 3 3 0 1 9 10 5 6 9 10 5 7	Case #1: 1.0000000009 Case #2: 1.5000000007 Case #3: 1.0000000001 Case #4: 0.8333333338

Sample Input 2

```
20
8 2 3 7
9 5 6 6
6 2 1 6
5 5 3 3
5 3 4 5
9 2 3 9
5 5 3 4
7 3 6 6
6 5 2 4
8 3 7 7
6 3 5 6
7 4 2 5
10 3 2 2
7 2 4 4
7 4 3 6
6 3 0 5
8 5 4 5
7 2 6 7
5 3 5 5
8 5 1 3
```

Sample Output 2

```
Case #1: 8.0000000000
Case #2: 2.2500000005
Case #3: 6.0000000000
Case #4: 1.2500000006
Case #5: 2.5000000006
Case #6: 9.0000000000
Case #7: 1.0000000003
Case #8: 3.5000000008
Case #9: 1.0000000005
Case #10: 4.0000000009
Case #11: 3.0000000007
Case #12: 2.0000000006
Case #13: 5.0000000006
Case #14: 7.0000000000
Case #15: 1.5000000002
Case #16: 1.0000000005
Case #17: 2.0000000009
Case #18: 7.0000000000
Case #19: 2.5000000006
Case #20: 1.6666666670
```

Problem D
Room Finding

On her day off, Lea went to visit some of her friends in the hospital (who had an unlikely accident involving an anteater and a surfboard). Once there, she asks for the directions to her friends' rooms.

The hospital is split up into several different wings, each containing some rooms for patients and a station warden who reigns over his respective station with an iron fist. All the rooms in these stations are numbered consecutively. However, after the "Great Station Wars", a conflict over stolen wheelchairs, the stations stopped talking to one another. That was a long time ago and almost all of the wings have been renovated or even completely rebuilt since then. Nowadays, the room numbers between stations do not match at all anymore. Thus, station 1 could contain rooms 7 through 10 while station 2 could contain rooms 9 through 11 (and so, there would be more than one room with numbers 9 and 10, respectively).

At the entrance, Lea meets an old desk officer who tells her all of this. He also asks her to "stay a while and listen", so he could tell her more about how the hospital used to be. Lea just wants to visit her friends though, so he hands her a table with descriptions of how to get to all the rooms of the hospital.

Room Number	Station	Directions
7	1	From the entrance, follow the signs to the flower garden until you reach the emergency room. Enter the last door on the left before reaching it.
8	1	From the entrance, go up the stairs twice, pass over the second bridge, then down, down, left, right, left, right. Knock on the door labelled "B", proceed through, then follow the signs pointing to "A".
9	1	From the entrance, follow the signs pointing north for 337 (average human) steps. Turn left, enter the third door on the right.
9	2	From the entrance, enter the sixth corridor from the left, follow it until you come to a dead end. Make a U-turn, and enter the second door on the right.
10	1	From the entrance, follow the smell of anesthetics until you reach the narcotics department. From there, ask for someone called "Dr. Fuzzy". He will show you the way.
10	2	From the entrance, go up the stairs twice, to the first bridge. Bribe the troll with some fish. Follow the signs pointing to the "Pointy Needle Inn" (the staff's break room). Pass straight through it, and enter the third door on the left.
11	2	From the entrance, ring the service personnel bell. Someone pushing a laundry cart will arrive. Follow him for 17 minutes, then enter the door on the right.

Figure D.1: Example Map

The map has been sorted lexicographically by (Room number, Station). Unfortunately, at some point in the past, the map has been ripped apart and the first column has been lost. Her friends knew this, so they told Lea their station and the line numbers with directions on the map, so she could find them more easily. However, Lea would still like to know their actual room numbers, so she would be able to follow signs. Given the line number on the map, can you tell her the room number?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers s f, where s denotes the number of stations in the hospital and f is the amount of friends Lea wants to visit. s lines follow. The i-th line contains two integers u_i v_i denoting that station i contains rooms numbered u_i through v_i. f lines follow. The i-th line contains an integer r_i stating the line number of her i-th friends room.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. Output f more lines. Line i should contain a single integer x_i where x_i is the room number of friend i.

Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq s \leq 1000$
- $1 \leq f \leq 1000$
- $1 \leq u_i \leq v_i \leq 2^{31} - 1$ for all $1 \leq i \leq s$
- $1 \leq r_i \leq 2^{31} - 1$ for all $1 \leq i \leq f$

Sample Input 1

```
2
2 3
7 10
9 11
2
4
5

3 4
6 10
9 12
4 8
3
7
9
1
```

Sample Output 1

```
Case #1:
8
9
10
Case #2:
6
8
9
4
```

Sample Input 2

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

3 4
4 5
4 5
4 5
3
1
2
3

3 3
3 5
3 4
3 5
5
5
5

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

Sample Output 2

```
Case #1:
3
3
3
4
3
Case #2:
4
4
4
4
Case #3:
4
4
4
Case #4:
3
4
3
3
4
```

This page is intentionally left blank.

Problem E
Diplomacy

International relations are hard. In the light of some recent international crisis, Lea's school decided that graduates should at least have a basic understanding of why that is. So, they set up a simulation of an international meeting for their graduates where every person represents a different country and has their own agenda. Lea decided to participate and so, she got her own agenda: As a representative of one of the larger countries, she wants to sway the course of history in her favour by rallying an alliance of countries that are able to overcome the resistance of all the other countries. For that, she gathered allies by making friends through diplomatic relations.

However, there are some countries that just hate one another because one country killed the king of the other country some hundreds of years ago. So, Lea has to pick up on the small, intricate signs of mutual hate or friendliness between the other countries' representatives. For example, she might pick up a small, intimate grin shared between the representatives of Templonia and Poorland and concludes that the two may be allied and plotting her downfall. Or she might see the representative of Beachistan slip some ominous powder into the drink of the representative of another country and infer that these two will probably not form an alliance anytime soon.

She also uses some basic properties of friendship and animosity:

- The friends of my friends are my friends as well.
  If x and y are allied and y and z, then x and z are allied as well.
- Friendship is mutual.
  If x is allied with y, then y is allied with x.
- Hatred is mutual.
  If x hates y, then y hates x.
- A common cause unites people.
  If x hates z and y hates z, then x and y form an alliance.
- An alliance has common enemies.
  If x is allied with y and x hates z, then y hates z as well.

After picking up on all these hints, can you tell her if her alliance gathered more than half of all the countries? This would establish a very powerful position for Lea (and her country, of course) from which she can use diplomacy to batter the rest of her enemies into submission (trade sanctions, for example).

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with a line consisting of two integers n, the number of countries, and m, the number of interactions between countries. The countries are numbered from 1 to n, with Lea representing country number 1. m lines follow, signaling a sign for either friendship or antipathy between the representatives of two countries. The i-th line is either `F` x y, signaling friendship or `A` x y, signaling antipathy between countries x and y.

Output

For each test case, output one line containing "Case #i: r" where i is its number, starting at 1, and r is "yes", if Lea's alliance has gathered more than half of all countries and "no" otherwise. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$

- $1 \leq n \leq 20000$
- $0 \leq m \leq 75000$
- $1 \leq x, y \leq n, x \neq y$
- The given relations will not be inconsistent (no two countries will be allies and enemies at the same time).

Sample Input 1

```
2
5 3
F 1 2
A 2 3
A 3 4

5 3
F 1 2
A 2 3
F 4 5
```

Sample Output 1

```
Case #1: yes
Case #2: no
```

Sample Input 2

```
5
3 3
A 3 1
F 2 1
A 2 3

5 1
F 5 1

3 1
F 3 2

5 4
F 3 5
F 5 1
F 1 2
F 1 5

4 2
A 2 4
A 4 2
```

Sample Output 2

```
Case #1: yes
Case #2: no
Case #3: no
Case #4: yes
Case #5: no
```