Problem A:

Everybody hates Raymond. He's the largest stamp collector on planet earth and because of that he always makes fun of all the others at the stamp collector parties. Fortunately everybody loves Lucy, and she has a plan. She secretly asks her friends whether they could lend her some stamps, so that she can embarrass Raymond by showing an even larger collection than his.

Problem

Raymond is so sure about his superiority that he always tells how many stamps he'll show. And since Lucy knows how many she owns, she knows how many more she needs. She also knows how many friends would lend her some stamps and how many each would lend. But she'd like to borrow from as few friends as possible and if she needs too many then she'd rather not do it at all. Can you tell her the minimum number of friends she needs to borrow from?

Input

The first line contains the number of scenarios. Each scenario describes onecollector’s party and its first line tells you how many stamps (from 1 to1000000) Lucy needs to borrow and how many friends (from 1 to 1000) offer hersome stamps. In a second line you'll get the number of stamps (from 1 to 10000) each of her friends is offering.

Output

The output for every scenario begins with a line containing "Scenario#i:", where i is the number of the scenario starting at 1. Then print asingle line with the minimum number of friends Lucy needs to borrow stampsfrom. If it's impossible even if she borrows everything from everybody, write impossible. Terminate the output for the scenario with a blank line.

Sample Input

```
3
100 6
13 17 42 9 23 57
99 6
13 17 42 9 23 57
1000 3
314 159 265
```

Sample Output

```
Scenario #1:
3
Scenario #2:
2
Scenario #3:
impossible
```

Problem B:

A friend of you is doing research on the Traveling Knight Problem (TKP) where you are to find the shortest closed tour of knight moves that visits each square of a given set of n squares on a chessboard exactly once. He thinks that the most difficult part of the problem is determining the smallest number of knight moves between two given squares and that, once you have accomplished this, finding the tour would be easy.Of course you know that it is vice versa. So you offer him to write a program that solvesthe "difficult" part. Your job is to write a program that takes two squares a and b as input and then determines the number of knight moves on a shortest route from a to b.

Input Specification

The input file will contain one or more test cases. Each test case consists of one line containing two squares separated by one space. A square is a string consisting of a letter (a-h) representing the column and a digit (1-8) representing the row on the chessboard.

Output Specification

For each test case, print one line saying "To get from xx to yy takesn knight moves.".

Sample Input

e2 e4
a1 b2
b2 c3
a1 h8
a1 h7
h8 a1
b1 c3
f6 f6

Sample Output

To get from e2 to e4 takes 2 knight moves.
To get from a1 to b2 takes 4 knight moves.
To get from b2 to c3 takes 2 knight moves.
To get from a1 to h8 takes 6 knight moves.
To get from a1 to h7 takes 5 knight moves.
To get from h8 to a1 takes 6 knight moves.
To get from b1 to c3 takes 1 knight moves.
To get from f6 to f6 takes 0 knight moves.

Problem C:

Programming training is an uneasy task which makes the contestants exhausted and staving after a busy training day. UESTC ACM ICPC team has been suffering from go to a chill dinner together because every one may have their own taste of food. For instance, hhb doesn't like spicy food while lxhgww hate dishes containing sugar, love8909 and zhymaoiing despise pepper. To be ridiculous, Hysramp never eat duck. To deal with it, they would never choose the dishes whose name contains "spicy", "sugar", "pepper", "duck". It is Mr. Gou's treat for dinner today and he claims that he has brought only Nyuan in RMB. Everybody wants to use up Mr. Gou's money as greedy as possible without exceed. In addition, every dish could be chosen most once. After the waiter shows the menu, a discussion about order dishes has begun. Can you help them to determine the maximum money can they spend while obeying their taste.

Input

The first line of the input contains one integer T, which indicate the number of test cases. The first line of each test case contains one real number N,($0 \le N \le 10^9$) the money Mr. Gou has, and one integer M,($1 \le M \le 20$) indicating the number of dishes they can choose. Following M lines are in format of a string S and a real number P.($0 \le P \le 10^8$) S stands for the name of the dish and the $P$is the price of which. The name of the dishes will be unique and guaranteed only include lowercase English letters with the length no more than 20.

Output

One line for each test case contains only one real number accurate to two digits after the radix point indicating the answer, the maximum money can they spend while obeying their taste.
each line may contains more than one input.

Sample Input

```
1
98.6 6
spicyfish 10.0
supersugar 9.0
riseonly 5.0
lovelyduckling 32.5
kaohongshu 90.0
mengjichi 8.5
```

Sample Output

```
98.50
```

Problem D

A hacker, who has stolen a lot of important information, wants to encrypt them in his computer so that only he can access them. He has devised an algorithm he thinks is unbreakable because no one would expect something like it.

To ***"Decrypt"***a text in his system, one should start from the first character directly after the one and only "*" character of the Encrypted string, which is also the first character of the original text. Every time, we find the ASCII code of the current character, (for e.g. 60), start from the beginning, and go that number of times forward until we get to a new character (So we would be on the 60th character of the encrypted string in the previous example). If during this process we get to the end of the string, we resume counting from the beginning of the text (Which means in the previous example, if the length of the string is 13 and the character we read had the ASCII code of 60, in the end we would be looking at the 9th character of the string). After each process, the character we reach is the next character of the original string, and also we should start the next process using this one's ASCII code. If during this, we reach a character we had read before or we went back to the "*" character (For example, if we are going to read the 9th character for the second time), we should instead find the first character we have not read yet directly after the current one and resume the process using it.

One important thing to note is that if we remove the "*" from the encrypted string, and see it as a number in base 256 (Which means that each character in the string counts as a digit, with the ASCII code being the value), it is the largest possible such number that, if decrypted, results in the original string.

Now, write a program that, given the ***original text***, gives the ***encrypted one*** as output.

Input:

The program should get N (N < 32768), which is the number of input strings, and then, get that number of strings consisting solely of letters (a, b, c, …, z, A, B, C, … Z).

Output:

The program should print the encrypted text for each of these inputs, separated by new lines.

Sample Input:

3

salam

pashmak

kolahghermezi

Sample Output:

salam*

s*pamhak

zirmh*kgeoaehl

Problem E:

Our sad tale begins with a tight clique of friends. Together they went on a trip to the picturesque country of Molvania. During their stay, various events which are too horrible to mention occurred. The net result was that the last evening of the trip ended with a momentous exchange of "I never want to see you again!"s. A quick calculation tells you it may have been said almost 50 million times!

Back home in Scandinavia, our group of ex-friends realize that they haven't split the costs incurred during the trip evenly. Some people may be out several thousand crowns. Settling the debts turns out to be a bit more problematic than it ought to be, as many in the group no longer wish to speak to one another, and even less to give each other money.

Naturally, you want to help out, so you ask each person to tell you how much money she owes or is owed, and whom she is still friends with. Given this information, you're sure you can figure out if it's possible for everyone to get even, and with money only being given between persons who are still friends.

Input

The first line contains two integers, n ($2 \leq n \leq 10000$), and m ($0 \leq m \leq 50000$), the number of friends and the number of remaining friendships. Then n lines follow, each containing an integer o ($-10000 \leq o \leq 10000$) indicating how much each person owes (or is owed if $o < 0$). The sum of these values is zero. After this comes m lines giving the remaining friendships, each line containing two integers x, y ($0 \leq x < y \leq n-1$) indicating that persons x and y are still friends.

Output

Your output should consist of a single line saying "POSSIBLE" or "IMPOSSIBLE".

Sample Input

5 3
100
-75
-25
-42
42
0 1
1 2
3 4

Sample Output
POSSIBLE

Sample Input 2
4 2
15
20
-10
-25
0 2
1 3

Sample Output 2
IMPOSSIBLE

Problem F:

The people of the city want to elect a mayor from among some candidates. Each person has only one vote. There is a man behind the scenes who wants to change the result of the election. He wants to do that by making people meet each other. It is known that if two groups of people meet, they start to debate about the election and change the minds of other group's members in the way described below:

1: If member x1 is in the first group and he wants to vote for p1, if there is at least one other person like y1 in the other group who also wants to vote for, p1, his opinion doesn't change.

2: If member x2 is in the first group and he wants to vote for p1, if there is no other person voting for p1 in the other group, one of these happens:

2.1: If a candidate like p2 has more votes than any other candidate in the other group, x2 changes his vote to p2.

2.2: If there isn't a definite candidate who has the most votes in the other group (For example if p3 and p4 both have the same number of votes, which is more than any other candidate, in that group), p2's opinion doesn't change.

3: People won't state their new candidate choice before the meeting is completely finished, so, if someone changed his vote, other people won't know that before the meeting has finished. This means that to change votes, people look at the state of the other group's members' votes ***before*** the start of the meeting.

4: After a meeting, members of the two groups will merge and form a single new group consisting of all of them.

5: At the start, there is no group with more than one members.

6: Each person is always a member of one, and only one group.

To facilitate a meeting, the man behind the scenes invites two people to meet each other, knowing well that both of them will bring their respective groups to the meeting.

Your goal is to write a program that, given the initial opinion of the people, and the man behind the scene's plan of meetings, prints each person's votes after all of the meetings have taken place.

Input

Input consists of number N ($N < 10000$), which is the number of the people of the city, followed by each person's vote (which is a non-negative integer less than 10). After that, it gets M, which is the number of planned meetings, followed by pairs of numbers, representing the index of the people who are going to meet. The meetings are sequential, which means that each one happens after the previous one is finished.

Output

Output is the vote of each person after the meetings, which are separated by a new line.

INPUT SAMPLE:

```
4
9
3
3
2
4
1 2
3 2
2 0
3 0
```

OUTPUT SAMPLE:

```
2
9
9
9
```

Problem G: Clever Police and Terrorist Troop.

A clever police lives in a city with *n* segments. These *n* segments are connected with a number of streets. A terrorist group in all segments of the city is playing. They are constantly changing their segment. A terrorist locating in the segment *i*, who wants to go to segment *j*, chooses the shortest path between segments *i* and *j*. Our clever police desires to write a program to find a segment with the maximum traffic of terrorists movements.

Input

The first line of the input contains one integer *n*, which indicates the number of test cases. For each test case, the next two lines show *v* (the number of segments) and *e* (the number of streets between segments) respectively. The *e* following lines contain three numbers: departure segment, destination segment, and the length of the street between the departure and the destination segments.

Output

The index of a segment, where the police must be located (the segment with the maximum traffic of terrorists).

INPUT SAMPLE:

```
1
7
9
0 2 3
0 3 2
0 5 1
1 6 18
2 4 3
3 6 17
4 6 15
4 5 3
6 5 14
```

OUTPUT SAMPLE:

```
6
```