A. Window

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

"Anas" is a good citizen lives in "Lateria". Lateria is a city full of naughty children, you can not sleep, eat nor relax! -poor Anas-. Anas is my friend and knows that I can get your help in solving problems so he asked me for this favor and I will not say no, here is Anas's problem: One of Lateria children broke Anas's rectangular window, Anas knew that naughty child and asked his parents to replace the broken window, and gave them the length and width of the broken window (X and Y in order), when the child's parent brought him the new one, he discovered that he told them the dimensions reversed Y X instead of X Y, what a bad luck! Now he asks you to tell him the area of the new window with the reversed dimensions!! eg: Anas will give you the dimensions X Y you should tell him the area formed by the rectangular window with dimensions Y X.

Input

the first line of input is an integer $0 < T < 100$ then T lines, each line is a test case contains two integers $1 \leq X,Y \leq 100000$ in order separated by a single space.

Output

for each test case, print one line contains a single integer represent the the rectangle area of dimensions Y and X.

Examples

input

```
3
5 4
1 2
33 33
```

output

```
20
2
1089
```

B. Paper Game

time limit per test: 1 second

memory limit per test: 64 megabytes

input: standard input

output: standard output

Hasan and Hussain are playing a game with W × H paper, the game consist of turns; Hasan starts first. On each turn, the player chooses one of the pieces of paper and cuts it into two papers such that the cut must be a straight line parallel to one of the paper sides and must cut the paper to two pieces of paper with integer dimensions.

The loser is the one who can't make a move (i.e when all pieces of paper become 1 × 1). Assuming that Hasan and Hussain are playing optimally, who will win the game?

Input

The first line of input is an integer $T (1 \le T \le 101)$ which represents the number of test cases. Each of the next T lines consists of two space-separated integers W,H $(1 \le W,H \le 100,000)$ the width and the height of the paper.

Output

For each test case, output a single line representing its answer; if Hasan is the winner output "Hasan" (without quotes) otherwise "Hussain" (without quotes).

Examples

input

```
3
1 1
2 1
2 2
```

output

```
Hussain
Hasan
Hasan
```

C. Rectangles

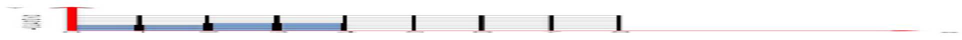
time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

You have n rectangles, each of which is described by three integers i, j and k. This indicates that the lower-left corner of the rectangle will be located at the point (i, 0) and the upper-right corner of the rectangle will be located at the point (j, k). For example, if you have a description of four rectangles like this: 0 2 3 0 3 1 1 2 2 2 4 4 The area occupied by these rectangles will be as follows:



The total area in this case will be 14. You are given n and n triples (i, j, k), your task is to compute the total area occupied by the rectangles which are described by the n triples.

Input

The first line will contain a single integer T indicates the number of test cases. Each test case will consist of n + 1 lines, the first one will contain the number of rectangles n and the following n lines will be the description of the rectangles. Each description will be a triple (i, j, k) as mentioned above. The Constraints: 10 <= T <= 20 1 <= n <= 100 0 <= i < 100 1 <= j < 100 i != j 1 <= k <= 100

Output

For each test case, print a single integer in a separated line indicates the total area occupied by the rectangles.

Examples

input

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

output

```
14
```

D. Sequences

time limit per test: 1.5 seconds

memory limit per test: 4 megabytes

input: standard input

output: standard output

One of the most wonderful qualities of an ACMer is to be multi interests so he combines multiple qualifications and hobbies not just coding. Hussain is one of the most qualified ACMers to mention when talking about hobbies and other domains of personality training despite of his qualifications in ACM problem solving and math. It's very known about Hussain his obsession in hunting and shooting, he used to pass hours training on empty cans in his childhood. These days, Hussain became a professional and still challenge others in this game, but for his bad luck he accidentally challenged a professional ACMer, without mentioning the name, so this ACMer made a game for Hussain. He numbered N targets for Hussain with random numbers and challenged him to shoot the minimum number of targets so the remaining numbers will form a sequence of increasing (by one) numbers in their current order. Example: if there is 6 targets numbered as follow: 2 5 7 3 2 4 Hussain will shoot 5,7 and the second 2 remaining for 2 3 4. Now, Hussain will focus on shooting, we will help him and focus on the targets he must shoot. But No! Hussain is an very good ACMer, we will make it hard for him and just tell him the number of the remaining targets in the sequence.

Input

First line contain an integer T represents the number of test cases 0 < T < 100, each test case consists of two lines, the first one is an integer 0< N < 20000 represents the number of targets, then followed by the second line that contains N numbers each number 0 < Xi < 20000 represents the number written on the i'th target.

Output

For each test case print one number represents the remaining sequence's length can be created by the input where it should be the maximum length and each number of it follow its previous by 1.

Examples

input

```
4
6
2 5 7 3 2 4
7
2 18 65 33 11 5 4
5
2 7 5 9 3
5
9 8 7 10 11
```

output

```
3
1
2
3
```

Note

Please consider a large input file.

E. Napoléon

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

In the world of chess, there is a good plan to start your game with which is “Napoléon plan”. “Napoléon plan” is commonly based on two pieces, (the queen and a bishop), using these two pieces you can guarantee the win after some moves if you are facing a beginner opponent or any way you can start your game with it to explore your opponent qualifications. Further, this plan can be used and applied in proceeding states of your game so you can disperse your opponent before you surprise him/her with a successful apply of “Napoléon plan” and saying your mortal word “checkmate”.

The plan; substantially, says that you can checkmate your opponent king by constricting it with its allies, threatening it with your queen which is guarded; commonly, by your bishop.

The picture displays a successful apply of “Napoléon plan”.

Our problem goes in a further state of the game, when you need to get your bishop to the right state guarding your queen for a successful apply of the plan. Given the Current location of the bishop and its Target state in the form Xc, Yc, Xt, Yt starting the count from zero, tell the queen: the minimum number of squares that the bishop pass in the minimum moves assuming an empty chess board.

Input

The first line of input is the number of test cases 0<T<5000, the second line is an integer for the chessboard size 0<N<=20, then (2*T) lines follow. Each test case is two lines; the first line contains two integers 0<=Xc,Yc<N representing the current location of the bishop, the second line contains another two integers 0<=Xt,Yt<N representing the target location of the bishop.

Output

For each test case, print a line containing the minimum number of squares passed by the bishop on its way to the target location taking the minimal possible moves or print “can't reach!” in case of impossible to reach the target from the current location.

Examples

input

```
3
8
0 0
7 1
5 5
5 5
0 5
6 7
```

output

```
7
0
6
```

F. The Best Strategy

time limit per test: 2 seconds

memory limit per test: 64 megabytes

input: standard input

output: standard output

Carlo Ancelotti "Real Madrid's coach" is so Sad and disappointed, and Florentino Perez fired him after getting knocked out from the UEFA Champions League against Juventus. Carlo is so good in algorithms and strategies (he is an engineer), and he heard about the ACM competition and decided to train a team to qualify to ICPC. After 2 years of training Carlo became really experienced with his team, and now he knows how much time his team needs to solve a problem (read it and code it correctly). Given the required solving time for each problem, help Carlo to determine the highest number of problems his team can solve with the minimum possible penalty.

Input

Your program will be tested on one or more test cases. The first line of the input contains a single integer T ($1 \leq T \leq 100$) the number of test cases. Followed by T test cases. Each test case consists of two lines. The first line contains a single integer N ($8 \leq N \leq 15$), the number of problems .The next line consists of N integers separated by a single space: p_i ($4 \leq p_i \leq 300$) which refers to the solving time for each problem.

Output

For each test case print a single line containing "Case n:" (without the quotes) where n is the test case number (starting from 1) followed by a single space, followed by the number of problems his team can solve and the minimum possible penalty.

Examples

input

```
2
8
252 244 6 109 294 31 67 270
8
218 48 273 69 281 224 250 193
```

output

```
Case 1: 4 360
Case 2: 2 165
```

Note

The total penalty is the sum of penalties for all solved problems. The penalty for a solved problem is the time of the accepted submition. And the period of the contest is 300 minutes.

G. Cutie Pie

time limit per test: 1 second

memory limit per test: 64 megabytes

input: standard input

output: standard output

Consider a NxM small-letters grid. SVU asked you to check whether this grid is a Cutie Pie or not A grid is a cutie pie if you can find the word "pie" in any direction (vertical, horizontal, and radial). Your program should output "Cutie Pie!" if the grid contains the word "pie" or "Sorry Man" otherwise

Input

The first line contains T 1<=T<=10000 the number of test cases. The followed T lines represent the test cases, each one contains two integers $0 < N,M \leq 20$ then N lines each of them contains M English small-letter separated by space characters. There is a blank line between each two successive test cases.

Output

For each test case output "Cutie Pie!" if the grid in the test case contains the word "pie" or "Sorry Man" otherwise.

Examples

input

```
2
3 5
o p r d t
i i i i e
f f s e d

4 3
o p r
o k r
i i u
f f s
```

output

```
Cutie Pie!
Sorry Man
```

H. Weekend

time limit per test: 2 seconds

memory limit per test: 250 megabytes

input: standard input

output: standard output

You and your friends are going to camping next weekend , because you are the best driver and you are living in this town for ten years , you have to drive the car from your home and pick up each of your friends to take them with you ,then finally go to the camping site. You have the map of the town and the length of every street ,and homes of friends will be at some intersections. you want to start from your home then pick up all of your friends then go to the camping site as fast as you can so you have to find the shortest path to do that. you can pick up your friends in any order but you have to make sure you pick up all of them. NOTE: you can pass by any intersection more than once including home and camping site .

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cases ($1 \le T \le 50$). Followed by the test cases, the first line of each test case contains three integers N,M,F ($3 \le N \le 100$, $0<M \le (N*(N-1))/2$, $1 \le F \le 9$) representing the number of intersections , the number of streets and number of friends respectively. Followed by M lines, each line contains 3 integers separated by a single space X Y Z ($1 \le X, Y \le N$), ($1 \le Z \le 1000$) which represent a street between intersection X and intersection Y with length Z in both direction (X and Y will be different). then follow F (1<fr<N) deferent integer representing friends home number(number of intersection where home is located). your home is at intersection number one and camping site is at intersection number N.

Output

For each test case print a single line containing "Case n:" (without quotes) where n is the test case number (starting from 1) followed by a space then the shortest path to go from your home to camping site after picking up all your friends .

Examples

input

```
2
4 4 2
1 4 3
1 2 2
2 3 2
3 4 2
2 3
6 8 2
1 2 1
2 3 2
3 6 3
1 4 11
4 5 2
5 6 20
5 3 5
4 3 10
4 5
```

output

```
Case 1: 6
Case 2: 20
```

Note

in the second test case the shortest path is : 1 > 2 > 3 > 5 > 4 > 5 > 3 > 6

I. Playing With Strings

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Dani and Mike are two kids ,They are playing games all day and when they don't find a game to play they invent a game . There is about an hour to arrive to school, because they love playing with strings Dani invented a game , Given a string and the winner is the first who form a palindrome string using all letters of this string according to the following sample rules : 1- player can rearrange letters to form a string . 2- the formed string must be palindrome and use all letters of the given string. 3- if there is more than one string chose the lexicographically smallest string . EX: string is "abacb" player can form : "abcba" and "bacab" ,but "abcba" is the lexicographically smallest. Mike asked you to write a Program to compute the palindrome string so he can beat Dani.

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cases ($1 \leq T \leq 1000$). Every test case on one line contain one string ,the length of the string will not exceed 1000 lower case English letter.

Output

For each test case print a single line containing the lexicographically smallest palindrome string according to the rules above. If there is no such string print "impossible"

Examples

input

```
4
abacb
acmicpc
aabaab
bsbttxs
```

output

```
abcba
impossible
aabbaa
bstxtsb
```

Note

Palindrome string is a string which reads the same backward or forward.

Lexicographic order means that the strings are arranged in the way as they appear in a dictionary.

J. Good Coins

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

It was a beautiful day, the sun was shining, the sky was clear and king Omar was listening to the birds peacefully under his favorite apple tree. Suddenly, an apple fell down and hit his head, and an idea came to him, he decided to reduce the types of coins in his kingdom to two types only. To take a wise choice, he had to choose good coins. We call a group of two coins good if we can exchange any integer amount of money by using them. For example, you can exchange any amount of money by using coins which have the values 3 and 4 so the group {3, 4} is good but you can't exchange the amount 3 if the two coins have the values 2 and 6, so the group {2, 6} isn't good. You are about to help king Omar to choose a good group of coins to be the national coins from a set of choices. You program have to test T group of two members where 0<T<5000, and print "GOOD" if the group is good, and "NOT GOOD" if it is not.

Input

The first line of the input contains T the number of test cases, followed by T lines, each line contains tow integers x,y separated by a space and 0 < x,y $\leq$ 10 ^{} 7

Output

For each line of the input print "GOOD" if {x,y} is good and "NOT GOOD" if it is not.

Examples

input

```
4
3 4
2 6
19739 6101
3636 351
```

output

```
GOOD
NOT GOOD
GOOD
NOT GOOD
```

Note

If someone X wants to give another one Y the amount 4 and the available coins have the values 3 and 5, then X can give Y 8 coins of the value 5 and Y give X 12 coins of the value 3 . 8x5-12x3=4

K. Clash Of Snakes

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Two lovely snakes Kiki and Susu like to play in a rectangular field divided into equal-sized squares. For us, it doesn't matter what the rules of the game play are, but we are interested in the rules of the starting. These rules are: 1- Each snake must lie in either a vertical or a horizontal line. 2- Both snakes must be inside the garden (their whole bodies are inside). 3- No snake can share a square in the field with the other snake. The following figure shows some valid and invalid cases of starting (in case of a field of size 5×5, Kiki's length is 3 and Susu's length is also 3):

Kiki and Susu are very confused and they don't know how to start the game, so they are asking you for help. Kiki and Susu will tell you what is the size of the grid and what are the lengths of them themselves, and you have to tell them in how many ways they can start the game. Note: the case in which a snake's head is at (x_1, y_1) and its tail is at (x_2, y_2) is considered different from the case in which the snake's head is at (x_2, y_2) and its tail is at (x_1, y_1).

Input

Your program will be tested on one or more test cases. The first line of the input contains T (1 $\leq$ T $\leq$ 1000) the number of test cases following. Each test case consists of a single line containing 4 space separated integers n,m,k,s representing the length of the field, the width of the field, the length of Kiki and the length of Susu respectively. For each test case, these constraints hold: (2 $\leq$ n,m $\leq$ 100000) and (2 $\leq$ k,s $\leq$ min(n,m)).

Output

For each test case print a single line containing "Case c: " without quotes where c is the number of the test case, and then the number of ways in which the two snakes can start their game modulo 1000000007.

Examples

input

```
4
2 2 2 2
3 3 3 3
2 3 2 2
4 4 3 2
```

output

```
Case 1: 16
Case 2: 48
Case 3: 88
Case 4: 1056
```