A. Dire Wolf

Time limit: 3s Memory limit: 128 MB

> Dire wolves, also known as Dark wolves, are extraordinarily large and powerful wolves. Many, if not all, Dire Wolves appear to originate from Draenor.
>
> Dire wolves look like normal wolves, but these creatures are of nearly twice the size. These powerful beasts, 8 - 9 feet long and weighing 600 - 800 pounds, are the most well-known orc mounts. As tall as a man, these great wolves have long tusked jaws that look like they could snap an iron bar. They have burning red eyes. Dire wolves are mottled gray or black in color. Dire wolves thrive in the northern regions of Kalimdor and in Mulgore.
>
> Dire wolves are efficient pack hunters that kill anything they catch. They prefer to attack in packs, surrounding and flanking a foe when they can.
>
> — Wowpedia, Your wiki guide to the World of Warcra

Matt, an adventurer from the Eastern Kingdoms, meets a pack of dire wolves. There are N wolves standing in a row (numbered with 1 to N from left to right). Matt has to defeat all of them to survive.

Once Matt defeats a dire wolf, he will take some damage which is equal to the wolf's current attack. As gregarious beasts, each dire wolf i can increase its adjacent wolves' attack by b_i. Thus, each dire wolf i's current attack consists of two parts, its basic attack a_i and the extra attack provided by the current adjacent wolves. The increase of attack is temporary. Once a wolf is defeated, its adjacent wolves will no longer get extra attack from it. However, these two wolves (if exist) will become adjacent to each other now.

For example, suppose there are 3 dire wolves standing in a row, whose basic attacks ai are (3, 5, 7), respectively. The extra attacks b_i they can provide are (8, 2, 0). Thus, the current attacks of them are (5, 13, 9). If Matt defeats the second wolf first, he will get 13 points of damage and the alive wolves' current attacks become (3, 15).

As an alert and resourceful adventurer, Matt can decide the order of the dire wolves he defeats. Therefore, he wants to know the least damage he has to take to defeat all the wolves.

Input

The first line contains only one integer T, which indicates the number of test cases. For each test case, the first line contains only one integer N ($2 \leq N \leq 200$).

The second line contains N integers a_i ($0 \leq a_i \leq 100000$), denoting the basic attack of each dire wolf.

The third line contains N integers b_i ($0 \leq b_i \leq 50000$), denoting the extra attack each dire wolf can provide.

Output

For each test case, output a single line '`Case #`x`:` y', where x is the case number (starting from 1), y is the least damage Matt needs to take.

Hint: In the first sample, Matt defeats the dire wolves from left to right. He takes $5 + 5 + 7 = 17$ points of damage which is the least damage he has to take.

Sample Input

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

Sample Output

```
Case #1: 17
Case #2: 74
```

B. Bounty Hunter II

Time limit: 5s Memory limit: 256 MB

Spike the bounty hunter is tracking another criminal through space. Luckily for him hyperspace ravel has made the task of visiting several planets a lot easier. Each planet has a number of Astral Gates; each gate connects with a gate on another planet. These hyperspace connections ure, for obvious safety reasons, one-way only with one gate being the entry point and the other gate being the exit point from hyperspace. Furthermore, the network of hyperspace connections nust be loop-free to prevent the Astral Gates from exploding, a tragic lesson learned in the gate ıccident of 2022 that destroyed most of the moon.
While looking at his star map Spike wonders how many friends he needs to conduct a search on every planet. Each planet should not be visited by more than one friend otherwise the criminal night get suspicious and flee before he can be captured. While each person can start at a planet of their choosing and travel along the hyperspace connections from planet to planet they are still bound by the limitations of hyperspace travel.

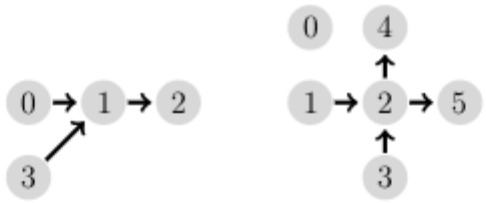

Figure B.1: Illustration of the Sample Inputs.

Input

The input begins with an integer N specifying the number of planets ($0 < N \leq 1000$). The planets are numbered from 0 to $N-1$. The following N lines specify the hyperspace connections. The i-th of those lines first contains the count of connections K ($0 \leq K \leq N-1$) *from* planet i followed by K integers specifying the destination planets.

Output

Output the minimum number of persons needed to visit every planet.

Sample Input 1	Sample Output 1
4 1 1 1 2 0 1 1	2

Sample Input 2	Sample Output 2
5 0 1 2	4

2 4 5
1 2
)
)

This page is intentionally left (almost) blank.

C. Pairs

Time limit: 2s Memory limit: 64 MB

In the secret book of ACM, it's said: ***"Glory for those who write short ICPC problems. May they live long, and never get Wrong Answers"*** . Everyone likes problems with short statements. Right? Let's have five positive numbers: X1,X2,X3,X4,X5. We can form 10 distinct pairs of these five numbers. Given the sum of each one of the pairs, you are asked to find out the original five numbers.

Input

The first line will be the number of test cases T. Each test case is described in one line which contains 10 numbers, these are the sum of the two numbers in each pair. Notice that the input has no particular order, for example: the first number doesn't have to be equal to {X1+ X2}. All numbers are positive integers below 100,000,000.

Output

For each test case, print one line which contains the number of the test case, and the five numbers X1,X2,X3,X4,X5 in ascending order, see the samples and follow the output format. There always exists a unique solution.

Examples

input
2 15 9 7 15 6 12 13 16 21 14 12 18 13 10 17 20 21 15 16 14
output
Case 1: 2 4 5 10 11 Case 2: 4 6 8 9 12

2015 Tishreen Collegiate Programming Contest

D. Cut Ribbon

Time limit: 1s Memory limit: 256 MB

Polycarpus has a ribbon, its length is n. He wants to cut the ribbon in a way that fulfils the following two conditions:

- After the cutting each ribbon piece should have length a, b or c.
- After the cutting the number of ribbon pieces should be maximum.

Help Polycarpus and find the number of ribbon pieces after the required cutting.

Input

The first line contains four space-separated integers n, a, b and c ($1 \le n, a, b, c \le 4000$) — the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly. The numbers a, b and c can coincide.

Output

Print a single number — the maximum possible number of ribbon pieces. It is guaranteed that at least one correct ribbon cutting exists.

Examples

input
5 5 3 2
output
2

input
7 5 5 2
output
2

Note

In the first example Polycarpus can cut the ribbon in such way: the first piece has length 2, the second piece has length 3.

In the second example Polycarpus can cut the ribbon in such way: the first piece has length 5, the second piece has length 2.

E. xortion

Time limit: 3s Memory limit: 128 MB

Hussain doesn't like long statements' problems, so he will describe his problem in a nutshell.

Hussain will give you an array $A[1....N]$ that consists of N positive integers.

Hussain will ask you Q Queries each one consists of an integer number X .

He wants you to find a number $P : 1 \le P \le N$ such that $(A[P]\text{^}X \ge A[I]\text{^}X)$ for each $(1 \le I \le N)$.

^ Refers to "XOR" operation in computer science .

In case of many possible values of P , take the minimum.

Input

The first line contains one number T – the number of testcases.

The second line contains two space-separated numbers, N and Q ($1 \le N \le 10^5$, $1 \le Q \le 3 \times 10^5$) — the size of the array and the number of queries .

The next line contains N space-separated integers the elements of the array A . All of them will fit into 32 bit signed integer.

Next Q lines contain one integer X (also fits in 32 bit signed integer) which was described above.

Output

For each testcase output Q lines . The I^{th} line will contain the answer of the I^{th} query.

Separate testcases by a blank line .

Examples

input
1 3 3 3 1 2 4 5 6
output
1 3 2

Note

Use fast I/O methods

2015 Damascus Collegiate Programming Contest (DCPC 2015)

F. Board Game

Time limit: 2s Memory limit: 64 MB

Feras bought to his nephew Saleem a new game to help him learning calculating. The game consists of a board with 4 rows and 4 columns with 16 cubes. Every cube has a number from 1 to 16. Let's define the power of a column as the sum of its elements. In the same way, the power of a row is the sum of its elements. Saleem should arrange the cubes in the board such that the power of all columns and all rows are equal. To make the game easier, the nice uncle, Feras, will help him arranging 7 cubes, and Saleem should arrange the rest of the cubes.

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 ($1 \le T \le 100$). Then the test cases. Each test case has four lines containing four integers. The j-th number in the i-th line describes the cell (i,j) of the board. If the number is -1 then the cell is empty and you have to fill it, otherwise, uncle Feras has already filled this cell.

Output

For each test case print a line in the following format: "Case c:" where c is the test case number starting from 1 then print the board in four lines every line has four numbers separated by space. If there is more than one solution print the solution that has the smallest order (See the notes below).

Examples

input
1 -1 -1 -1 -1 -1 -1 -1 -1 -1 5 13 12 3 8 9 14
output
Case 1: 11 6 10 7 16 15 2 1 4 5 13 12 3 8 9 14

Note

in the sample input there is more than one solution:

Solution1:

16 15 2 1

11 6 10 7

4 5 13 12

3 8 9 14

Solution2:

11 6 10 7

16 15 2 1

4 5 13 12

3 8 9 14

but we select solution2 because it has the smallest order when we write the rows in one line.

Solution1: 16 15 2 1 11 6 10 7 4 5 13 12 3 8 9 14

Solution2: 11 6 10 7 16 15 2 1 4 5 13 12 3 8 9 14

2015 Tishreen Collegiate Programming Contest

G. Tri-du

Time limit: 1s

Tri-du é um jogo de cartas derivado do popular jogo de Truco. O jogo utiliza um baralho normal de 52 cartas, com treze cartas de cada naipe, mas os naipes são ignorados. Apenas o valor das cartas,considerados como inteiros de 1 a 13, são utilizados.

No jogo, cada jogador recebe três cartas. As regras são simples:

- Um trio (três cartas de mesmo valor) ganha de uma dupla (duas cartas de mesmo valor).
- Um trio formado por cartas de maior valor ganha de um trio formado por cartas de menor valor.
- Uma dupla formada por cartas de maior valor ganha de uma dupla formada por cartas de menor valor.

Note que o jogo pode não ter ganhador em muitas situações; nesses casos, as cartas distribuídas são devolvidas ao baralho, que é embaralhado e uma nova partida é iniciada

Um jogador já recebeu duas das cartas que deve receber, e conhece seus valores. Sua tarefa é escrever um programa para determinar qual o valor da terceira carta que maximiza a probabilidade de esse jogador ganhar o jogo.

Entrada

A entrada consiste de uma ´unica linha que contém dois inteiros, **A** ($1 \leq \mathbf{A} \leq 13$) e **B** ($1 \leq \mathbf{B} \leq 13$) indicando os valores das duas primeiras cartas recebidas.

Saída

Seu programa deve produzir uma única linha com um inteiro representando o valor da carta que maximiza a probabilidade de o jogador ganhar a partida.

Exemplo de Entrada	Exemplo de Saída
10 7	10
2 2	2

XX Maratona de Programação da SBC 2015

Por Ricardo Anido, Universidade Estadual de Campinas Brazil

H. Sunlight

Time limit: 10s Memory limit: 1024 MB

A core right in Roman tenancy law was the availability of sunlight to everybody, regardless of status. Good sun exposure has a number of health benefits, many of which were known even in those ancient times.

The first act of a Roman city plan reviewer, then, is to survey the proposed structures to measure how well they distribute this precious resource. Given any one avenue of buildings arranged West-to-East, the number of hours for which each building is exposed to sunlight needs to be determined.

For the purpose of simplicity, the number of hours a building is in sunlight is proportional to the fraction of the 180 degrees of sky visible from its top. Thanks in no small part to the marvels of ancient engineering (and also to the strict nutritional regimes of old) you may assume each building is infinitesimally thin.

Input

- One line containing one integer N ($1 \leq N \leq 2 \cdot 10^5$): the number of buildings.
- N further lines each containing two space-separated integers X_i and H_i ($1 \leq X, H \leq 10^9$), the location and height respectively, in metres, of the i^{th}-most building from the west.

Output

On each of the N lines of output, write one real number to at least 4 decimal places of accuracy: the number of hours for which the peak of the i-th building is bathed in sunlight.

sample input 1	**sample output 1**
4 1 1 2 2 3 2 4 1	9.0000 12 12.00000 9.0

sample input 2	**sample output 2**
5 100 50 125 75 150 100 175 125 200 25	9.0 9.0 9.0 12.0 6.9357496

Robin Lee (UKIEPC 2015)

I. Kingdom

Time limit: 1s Memory limit: 1536 MB

Shin wants to be the “Great General of the Heaven”. It is his dream. But becoming a “Great General of the Heaven” is no ordinary task. For becoming a “Great General”, you have to be good in both fighting and tactical aspects of war.

Shin just received a new mission from headquarter. Enemies are planning to attack the Kingdom of Qin and he is ordered to stop them in any means necessary. The Kingdom of Qin can be considered as a connected graph of N nodes, where the different cities are nodes numbered from 1 to N and roads between them are the edges.

The enemies have already infiltrated the Qin. Their current position is not known yet, but soon the spies inside the enemy will alert Shin of their position. Meanwhile, the enemy is planning to attack headquarter of Qin’s Military Force. The position of headquarter in also unknown to Shin, but soon authorities of Qin will tell shin of its position.

So basically, even though Shin is not aware of the position of enemy or headquarter now, within few hours he will know both. Once he knows their position, he will march his army to stop the enemy. Let’s call the node in which the enemy is residing node E and headquarter node H.

Shin can only battle enemy in the middle of a road. Fighting inside the city needs to be avoided in order to avoid damage to citizens. But Shin needs to be careful when choosing road for the battle. There could be multiple paths between node E to H. If Shin waits in a road with his army and there is a path from E to H without using the road that Shin is guarding, then there is a possibility that the enemy will bypass Shin using that path and attack headquarter. So in order to avoid such situation, Shin has decided to guard a road such that the enemy will be unable to reach H from E without using that road. There could be multiple such roads and he calls them “Great Roads of Destiny”.

The real location of E and H have not arrived yet, so Shin is getting bored. In order to pass the time, he decided to train his tactical skills. He assumed a pair of integer as {E,H} and then tried to find out how many “Great Roads of Destiny” exists.

He now wants you to do the same. Given the graph of Kingdom of Qin, Shin will provide you with Q queries, where query will be a pair of integer representing E and H. Please tell Shin how many “Great Roads of Destiny” exists.

Input
The first line will contain an integer T (T<=5) indicating number of test cases.

For each test case, first line will contain a pair of integer N (2<=N<=100000) and R (1<=R<=200000) , where N is number of nodes and R is number of roads in the Kingdom of Qin. After that there are R lines each containing a pair of integer U,V (1<=U,V<=N) representing there is a road between node U and V. After that there is a single integer Q (1<=Q<=100000), which is number of queries Shin will ask you. Next Q lines will contain a pair of integer representing E and H, as explained in the problem.

You can safely assume that the graph will be connected. There will be no selfloops or multiple edges between nodes.

There could be blank lines, or extra spaces in input.

Output
For each case print a line “Case X:”, where X is the number of case. After that for each query print a line with an integer Y, where Y is the number of “Great Roads of Destiny”. See the sample input/output for better understanding.

Sample Input

```
3


3 2
1 2
2 3
2
1 2
1 3


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


5 5
```

1 2

2 3

3 1

2 4

3 5

1

1 4

Sample Output:

Case 1:

1

2

Case 2:

0

Case 3:

1

Explanation

In the first test case, Query 1 is 1 since the only road Shin should be guarding is the road between node 1 and 2. Query 2 is 2 since he can guard either the road 1->2 or 2->3. In the second case, Query 1 is 0 since there is no “Great Road of Destiny”. In the third case, for Query 1 the only road is 2->4.

Problem-setter: Mohammad Samiul Islam

Progkriya Contest January 2015

J. Fury Road

Time limit: 1s Memory limit: 1536 MB

Westland is full of war parties. There is always frequent war occurs in the fury road of Westland. Today is an another great day for the immortal Joe, because he is going to the war with his Kamakrazee war boys in the fury road. So, he sent Slit one of his war boys to count the number of war parties. In the fury road there are multiple war parties fight together, so it is difficult to count the number of war party but, Slit properly know that, if a group of war boys stand together, on the other word if they are adjacent (vertically, horizontally and diagonally) to each other, they would be counted as a single war party. As, there is large number of war parties in the fury road, it is not quite easy to count the number of war parties perfectly. So, Slit needs your help to determine the number of war parties in the fury road.

Now, you are given the number of war boys and their position in the Cartesian plane as (x, y) format which indicates the war boy's position. Now you have to determine the number of war parties.

Input

The input set starts with single line integer **T (1<=T<=50)** the number of test cases. Then following **T** cases starts with an integer **N (1<=N<=10000)** denoting the number of war boys in the fury road Then next **N** line contains a pair of integer **(X_i, Y_i) (0<=X_i, Y_i<=1000)** denoting the position of i-th war boy.

Output

For each case print "**Case X:** " (without the quotes) where **X (1≤X≤T)** is the case number. And then print the number of War parties in the fury road. Every new case should be printed in a new line.

Example

Input:

```
3
2
5 5
3 2
6
1 3
0 1
1 2
2 3
4 1
5 0
1
0 0
```

Output:

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

AIUB CS Fest 2015 (Ift Khan)

K. TWO SEQUENCES PROBLEM

Time limit: 1s Memory limit: 1536 MB

Problem statement:

Given two lists Aand B having the same length, find the length of longest subsequence of list A, whose sum is greater than or equal to the corresponding subsequence of list B. Corresponding subsequence means indices chosen in both of the lists must be the same.

Input format:

The first line contains an integer **T** , the number of test cases.

Then for each test cases, there are 3 lines.

The first line has an integer N, the number of elements in the lists A& B.

The second line contains N integers of the list A.

The third line contains N integers of the list B.

Output format:

For each test case, print the answer in a single line.

Constraints:

$1 < T < 50$

$1 \leq N \leq 10^5$
$0 \leq A[i] \leq 10^7$
$0 \leq B[i] \leq 10^7$

Sample input:

1

3

100 100 5

2 2 1000

Output:

2

L. Fake NP

Time limit: 1s Memory limit: 256 MB

Tavak and Seyyed are good friends. Seyyed is very funny and he told Tavak to solve the following problem instead of *longest-path*.

You are given *l* and *r*. For all integers from *l* to *r*, inclusive, we wrote down all of their integer divisors except 1. Find the integer that we wrote down the maximum number of times.

Solve the problem to show that it's not a *NP* problem.

Input
The first line contains two integers *l* and *r* ($2 \le l \le r \le 10^9$).

Output
Print single integer, the integer that appears maximum number of times in the divisors.

If there are multiple answers, print any of them.

Examples

input
19 29
output
2

input
3 6
output
3

Note
Definition of a divisor: https://www.mathsisfun.com/definitions/divisor-of-an-integer-.html (https://www.mathsisfun.com/definitions/divisor-of-an-integer-.html)

The first example: from 19 to 29 these numbers are divisible by 2: {20, 22, 24, 26, 28}.

The second example: from 3 to 6 these numbers are divisible by 3: {3, 6}.