A. Pizza Town

Description

You have just traveled to the Pizza Town. As the name implies, the most famous food in the town is pizza. There are n pizza restaurants, and each sells a unique pizza. You would like to try all of them, if possible, so you plan to eat the pizzas one-by-one. What you will do is the following:

1. Go to one pizza store and buy one pizza.
2. Go back to your hostel room and eat that pizza.
3. Go to another pizza store and buy another pizza.
4. ...

Then you find out that you may quickly get tired. You have e energy now. The round trip to the i-th restaurant costs you a_i energy, and eating the pizza from the i-th restaurant gives you b_i energy. Once you use up all e energy, you will get tired and cannot do any thing. However, if you arrive home with 0 energy, you can still eat the pizza that you bring back. How many pizza can you eat before running out of energy?

Input

The first line of the input file contains an integer T indicating the number of test cases. Each test case starts with a line containing two integers, n and e. The next line contains n integers, $a)1, \ldots, a_n$. Yet the next line also contains n integers, $b_1, \ldots, b_n$.

- $1 \leq T \leq 70$
- $1 \leq n \leq 10^5$
- $0 \leq e \leq 10^8$
- $0 \leq a_i, b_i \leq 10^8$

Output

For each test case output the maximum number of pizzas that you can buy and eat.

Sample Input

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

Sample Output

```
1
3
```

Note

For the first test case, you may go to either of the restaurants, but not both.

For the second test case, you can have 2 -> 3 -> 1, 2 -> 3 -> 4, 3 -> 2 -> 1, 3 -> 2 -> 4, or 3 -> 4 -> 2.

This page is intentionally left blank.

B. Pizza Milk Tea

Description

Long long time ago, when Tomato the Great was a young man, he worked in a pizza store. By then, he has fallen in love with milk tea. One day, he found a special store selling a specially avored milk tea - pizza milk tea! Since he loved both pizza and milk tea very much, he sworn to drink 1 liter of pizza milk tea everyday, no more, no less.

At the night of day 0, Tomato the Great has no money nor pizza milk tea. Every morning he receives his daily salary x dollars. At noon, he would drink 1 liter of pizza milk tea. If he has no pizza milk tea on hand to drink, he will buy it as many liters as possible. The cost per 1 liter of pizza milk tea is y dollars. Since Tomato the Great does not like fractions, he will only buy integral liters of pizza milk tea.

At the night of day z, he would like to check how much money he has. Can you calculate the correct amount of money for him?

Input

The first line of the input file contains an integer T indicating the number of test cases. Each test case is on one line, containing three integers, x, y and z.

- $1 \leq T \leq 50000$
- $1 \leq y \leq x \leq 10^9$
- $1 \leq z \leq 10^9$

Output

For each test case please output an integer in one line, denoting the correct amount of money.

Sample Input

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

Sample Output

```
1
6
1
```

This page is intentionally left blank.

C. Pizza Transportation

Description

The Pizza Festival is near! Every city has to send a pizza to the capital city in order to glorify the king. As a mayor, Tomato the Great has prepared a specially flavored pizza - milk-tea pizza! Then, he demands you to take this pizza to the capital city.

You would like to find the safest path to the capital city. But since you need to arrive the capital city before the festival, you cannot spend too long time on the way. Thus, you only want to consider the top-k shortest paths from your city to the capital city.

Now, given the map, please find out top-k shortest paths. Note that you do not mind to pass through a city more than once.

Input

The first line of the input file contains an integer T indicating the number of test cases.

The first line of each test case contains three integers, n, m and k, denoting the number of cities, the number of roads, and the number of shortest paths you want. The next line contains two integers, s and t, denoting the city you are in and the capital city. Each of the following m lines contains three integers, x_i, y_i and v_i, denoting a road from x_i to y_i which length is v_i.

The input guarantees that there is at least k paths from your city to the capital city.

- $1 \leq T \leq 15$
- $2 \leq n \leq 10000$
- $2 \leq m \leq 50000$
- $k \leq 10000$
- $1 \leq x_i, y_i \leq n$
- $1 \leq v_i \leq 1000$

Output

For each test case please output the length of top-k shortest paths in ascending order, each on one line.

Sample Input

```
2
4 4 4
1 4
1 2 10
2 3 10
3 4 10
3 2 10
2 2 5
1 2
1 2 5
2 1 7
```

Sample Output

```
30
50
70
90
5
17
29
41
53
```

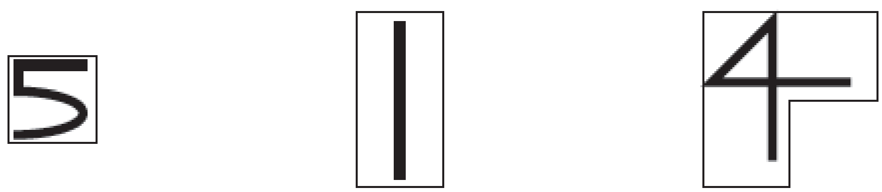
This page is intentionally left blank.

D. Pizza 514

Description

The Pizza Festival is coming. Everyone in the Pizza Kingdom is excited for this once- in-a-decade event. You have decided to present a jumbo pizza to God to thank him for everything and to honor his glory.

The jumbo pizza that you want to present is $n \times m$ in size and consists of jewelries and three kinds of small pizzas: Pizza-5, Pizza-1, and Pizza-4. Pizza-5 is 1×1 in size, with ham and pineapple toppings. Pizza-1 is 1×2 in size, with tomato, sliced mozzarella, and basil leaves toppings. Pizza-4 is 2×2 in size, L-shape, with sausage and olives toppings. You may rotate the small pizzas before putting into the jumbo pizza.



The priest has told you the pattern to place the jewelries in the jumbo pizza to bless your fortune. He also gave you a lucky sequence of 5, 1 and 4 to place the small pizzas. Following his suggestion, you would like to place the small pizzas in a way that when recording the pizza number from top to bottom and from left to right, the sequence is exactly the same as the lucky sequence. Every small pizza has to be placed completely inside the jumbo pizza, and there must be no empty space in the jumbo pizza. How many ways are there to place the small pizzas?

Input

The first line of the input file contains an integer T indicating the number of test cases.

Each test case starts with a line containing three integers, n, m and k, denoting the size of the jumbo pizza and the number of small pizzas in the jumbo pizza.

The next line contains k characters (5, 1, or 4) representing the lucky sequence. The following n lines presents the pattern of the jewelries. Each line contains m characters, '.' or '#'. All '#' will be filled by jewelries. You have to cover all '.' with small pizzas.

- $1 \leq T \leq 45$
- $1 \leq n, m \leq 18$
- $1 \leq k \leq 350$

Output

For each test case please output a line containing the number of ways to place the small pizzas satisfying all the constraints. Since the number may be large, please output the number modulo 0x514b122a.

Sample Input

```
3
4 4 6
514111
...#
#...
...#
...#
3 5 6
111514
.....
.....
#..##
4 3 5
```

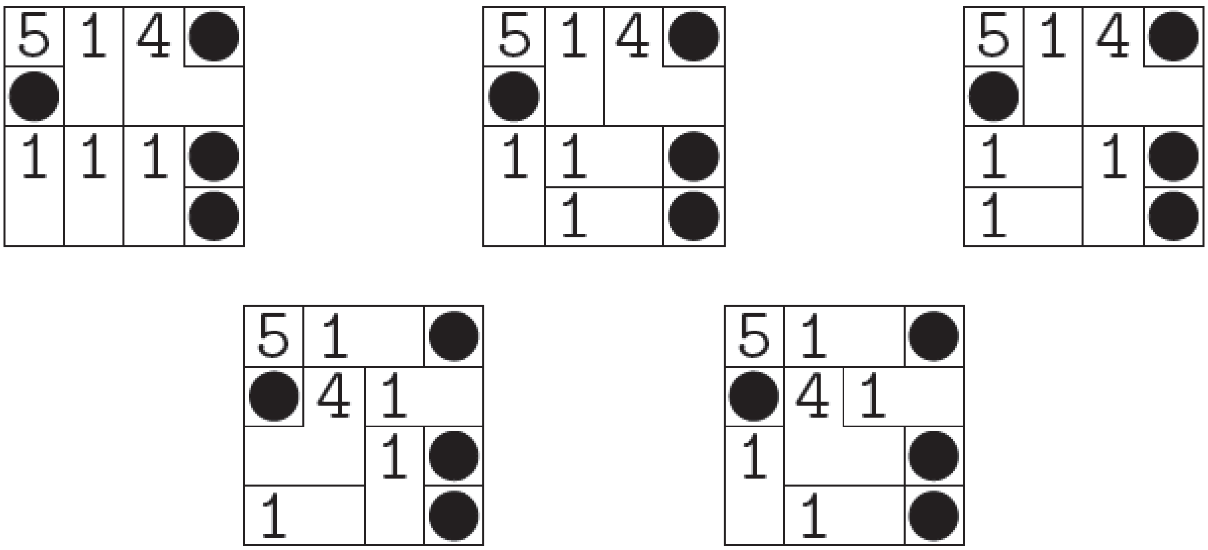
```
11154
...
...
#..
#..
```

Sample Output

```
5
1
4
```

Note

The solutions to the first test case are shown below.



E. Pizza Hut

Description

Every one knows the Pizza Hut, but seldom knows its special offer of composite pizzas. A composite pizza is a square pizza with n rows and n columns, and each cell consists of a small pizza. The customers may choose n kinds of sauces and n kinds of toppings, and then they will get one composite pizza with $n \times n$ small pizzas of all pairs of sauce and topping.

One day, Tomato the Great makes an extraordinary order of a 3×3 composite pizza. The three sauces he chooses are tomato sauce, cream sauce, and pesto sauce. And the three kinds of toppings he chooses are pineapple, mushroom, and sausage. Furthermore, he would like at least one of the small pizzas with tomato sauce to be on the first row of the composite pizza, the one with sausage topping and cream sauce to be on the main diagonal, and none of the ones with pineapple toppings to be on the left-most column or the last row.

The constraints are too complicated for the cook to make such composite pizza. He does not even know if all the constraints can be satisfied at the same time. Thus he turns to your help. Please write a program to check if all the constraints can be satisfied, and if so, please tell the cook how to make such composite pizza.

Input

The first line of the input file contains an integer T indicating the number of test cases.

The first line of each test case contains two integers, n and m. Then m lines follow; each describes one constraint. Each constraint contains three parts, separated by a white space: sauce type, topping type, and grid pattern. To satisfy the constraint, the small pizza with the given sauce type and topping type must be placed according to the grid pattern. The first n uppercase letters (ABC...) indicate n types of sauce, and the first n lowercase letters (abc ...) indicate n types of topping. An asterisk (*) in sauce type and/or topping type stands for wildcard.

Each grid pattern consists of a number of cell specications and the positional relationship between them in the following format:

```
[cell 1 spec][relation 1 2][cell 2 spec][relation 2 3][cell 3 spec]
```

There are three types of cell speci

cations: must-be-here (H), must-not-be-here (N), and reference (R). The reference cell specication is followed by a sauce type and a topping type (wildcards may also be used here). There are four types of positional relationships, up (^), down (v), left (<), and right (>), indicating that the following cell is in which direction from the previous cell.

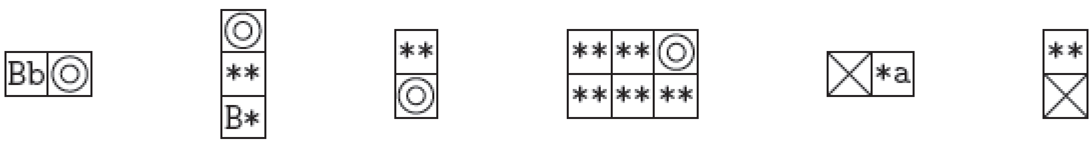
There are two kinds of grid patterns: positive patterns and negative patterns. A positive pattern contains exactly one must-be-here cell and one or more reference cells. The constraint is satisfied if such positive pattern presents in the composite pizza, i.e.. the small pizza with the given sauce and topping type is placed at the must-be-here cell with respect to the reference cells. If the given sauce or topping type or the reference cells contain wildcards, the wildcards can be replaced with any symbol for that type. There is at most one reference cell contains non-wildcard symbols.

A negative pattern contains at least one must-not-be-here cells, and zero or more reference cells. The constraint is satisfied if no such negative pattern shows up in the composite pizza. That is, the given small pizza must not be in any must-not-be-here cell, with respect to the reference cells (if any). (See the examples below.) There is at most one reference cell contains non-wildcard symbol for sauce or topping type. The input guarantees that each pattern contains at most one cell specication for the same position.

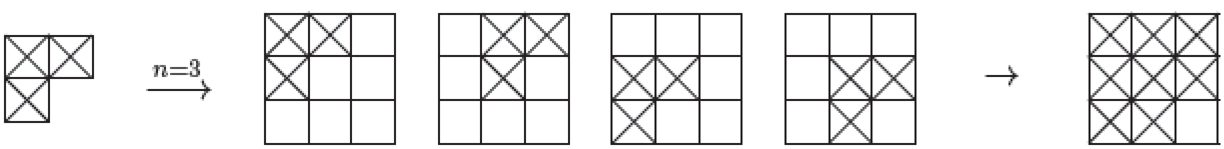
- $1 \le T \le 20$
- $1 \le n \le 3$
- $0 \le m \le n^2$

Examples:

- A a RBb>H: Pizza Aa must be at the right of the pizza Bb.
- * a HvR**vRB*: One topping-a pizza must be two rows above a sauce-B pizza.
- A a H^R**: Pizza Aa must be below a pizza, i.e. must not be on the first row.
- A * R**>R**>HvR**<R**<R**: One sauce-A pizza must not be on the first two columns, nor on the last row.



- A * N<R*a: No sauce-A pizza should be on the left of a topping-a pizza.
- A a R**vN: Pizza Aa must not be below any pizza, i.e. must be on the first row.
- A a N<NvN: Pizza Aa must be on the lower-right corner.



- A a N<N<NvR**>N: Pizza Aa must be on the lower-right or lower-left corner.

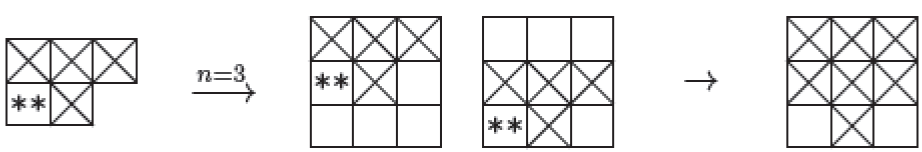


Figure 1: fig

Output

For each test case, please output the lexicographically smallest composite pizza satisfying all the constraint, or "IMPOSSIBLE"(without quote) if there is no solution.

The composite pizza should be printed in n lines. Each line should contain n pizza specications, separated by a white space. Each pizza specication consists of two characters, one for sauce, and the other for topping.

Sample Input

```
2
3 9
A a H>R**>R**vR**vR**
A b R**>R**vR**N>N^N^N
A c NvR**vN>R**>N^R**N
B a R**vHvR**>R**>R**
B b R**^R**^HN^R**>N
C a R**>R**>NvNN>N
C b R**R**vNvR**>N
3 8
A a H>R**vR**>R**vR**
B a H>R**>R**vR**H
A b H^RCa
B b RAb^R**>H
C b RBb^H
A c H^R**^RCb
B c HvR**>R**vRAc
```

Sample Output

```
Aa Ca Bb
Ba Ab Ac
Cc Cb Bc
Aa Bc Cb
Ba Ca Bb
Cc Ab Ac
```

This page is intentionally left blank.

F. Pizza Assembling

Description

Tomato the Great own a pizza factory, which produces pizzas from an assembly line. That is, each worker in your factory only does one single step of pizza production. One worker makes dough; another put toppings on the dough; yet another put cheeses on the dough; and someone else send the dough into the oven.

The Pizza Festival is coming, and Tomato the Great would like to prepare a great pizza for the great event. He would like the pizza consists of as many ingredients as possible. But he knows that some of the ingredients conflict with each other, and that putting those conflicting ingredients together results in a terrible pizza.

After carefully investigating, Tomato the Great finds out 31 key features of the ingredients. For example, the pineapple is sour and sweet, and the sausage is salty and contains meats. He uses binary string s_i of length 31 to describe each ingredient i. Each digit in the string represents a key feature. The j-th digit in the i-th binary string is 1 if the i-th ingredient has the j-th feature, or 0 otherwise.

Moreover, Tomato the Great finds out the key of conflict: Two ingredients conflict with each other if and only if their binary strings different only 1 digit. For example, 1 and 3 are conflict, but 1 and 4 are not conflict.

Input

The first line of the input file contains an integer T indicating the number of test cases.

Each test case consists of two lines. The first line contains an integer n, denoting the number of ingredients. The second line contains n integers, $a_1, \ldots, a_n$, where a_i is the binary value of the string s_i, i.e $\sum_{j=0}^{30} s_i[j] \times 2^j$

- $1 \leq T \leq 400$
- $1 \leq n \leq 50$
- $0 \leq a_i \leq 2^{31} - 1$

Output

For each test case please output two lines.

The first line should contain an integer m, indicating the maximum size of the ingredient set without conflicts.

The second line should contain m integers in any order representing the m ingredients in the set (in binary value format). If there are more than one such sets, you may output any one of them.

Sample Input

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

Sample Output

```
3
1 2 4
```

This page is intentionally left blank.

G. Pizza Oven

Description

You have a special pizza oven, named as Automatic Cooking Machine - Immediately Completing Pizza Cooker (ACM-ICPC). This pizza oven can bake one pizza at a time. No matter the amount of cheese or the size of pizza, the oven can automatically heat the pizza with the most suitable temperature, and finish baking almost immediately. You always turn on the pizza oven at time 0, and turn it off at time m.

Every thing works well except one flaw in the ACM-ICPC. The pizza oven will explode if no pizza is sent into it in k consecutive time units. Moreover, your cook has a weird habit that he only makes pizza in n certain time points: $a_1, \ldots, a_n$. That is, at each time a_i, your cook will make one pizza and send it into the pizza oven. Then, the pizza oven returns to your cool a well-baked pizza at time $a_i + \epsilon$ ($\epsilon < 1$).

In order to prevent the oven explodes, you have to pay for adjusting the pizza-making time. Your cook asks for $|a_i - a_i'|$ dollar to adjust a_i to a_i'. Also, your cook refuses to make more pizzas, and he cannot make two pizzas at the same time. What is the minimum cost you have to pay to keep the oven from explosion?

Input

The first line of the input file contains an integer T indicating the number of test cases.

For each test case, the first line contains three integers n, m and k. The following line contains n integers, $a_1, \ldots, a_n$. Note that all the adjusted a_i' must also be integers.

- $1 \leq T \leq 50$
- $1 \leq n \leq 250000$
- $1 \leq k \leq m \leq 10^9$

Output

For each test case please output the minimum cost as required, or `BOOM` if it is impossible to prevent explosion.

Sample Input

```
2
2 9 3
4 7
1 5 2
3
```

Sample Output

```
2
BOOM
```

This page is intentionally left blank.

H. Pizza Wave

Description

Tomato the Great will held a party at the night of Pizza Festival. Many friends of him will come to the party, so he has to prepare a huge wave of pizza! There are n pizza stores on the street, and each of them sells two kinds of pizzas. There are totally n kinds of pizzas which are sold on the street, and each kind of pizza is sold in exactly two pizza stores. Tomato the Great would like to collect at least one of each kind for the party.

However, the pizza stores also held a special event for Pizza Festival, called Absolutely Cool Mystery Invisible Content Pizza Container (ACM-ICPC). During the festival, one will not know in advance the taste of the pizza he is going to buy. That is, one will get a pizza in either of the two kinds sold in this store, but will not be sure about which kind of pizza exactly.

Tomato the Great knows the event, and he also know the number of pizzas of each kind sold in each store. Assume that all the pizzas are already made, and no more pizza will be produced. What is the minimum number of pizzas that he has to buy, to ensure that he can have at least one pizza of each kind? Note that he can know taste of these pizzas only after buying all pizza.

Input

The first line of the input file contains an integer T indicating the number of test cases.

Each test case starts with a line consisting of one integer n. The following n lines describes the n pizza stores. The i-th line contains four integers, a_i, x_i, b_i and y_i, indicating that the i-th pizza store has a_i boxes of Pizza x_i and b_i boxes of Pizza y_i.

- $1 \le T \le 100$
- $2 \le n \le 1000$
- $1 \le a_i, b_i \le 1000$
- $1 \le x_i, y_i \le n$

Output

For each test case please output an integer in one line, denoting the minimum number of pizza that Tomato the Great has to buy.

Sample Input

```
1
2
1 1 1 2
1 1 1 2
```

Sample Output

```
2
```

This page is intentionally left blank.

I. Pizza Equalization

Description

Have you heard the story of killing three warriors with two peaches? Now you have two pizzas and would like to give to three friends of yours. Of course, you do not want them to fight for the pizza, so you must divide the pizzas into three equal-sized portions.

Both pizzas you have are in convex quadrilateral shape, and they have one side in common. As shown in the figure below, ABCD and CDFE are your pizzas. You are also given two lines, L_1 and L_2. You have to choose a point on each line (like P on L_1 and Q on L_2), so that the area of region ABP, CPDQ, and QFE are exactly the same. Moreover, P should lie in ABCD and Q should lie in CDFE. Then, you may give each friend one of the three pieces.

How about other pieces, such as ACP and CQE? Those are for yourself!

Input

The first line of the input file contains an integer T indicating the number of test cases.

Each test case consists of three lines. The first line contains 12 integers $A_x, A_y, B_x, B_y, C_x, C_y, D_x, D_y, E_x, E_y, F_x$ and F_y, denoting the x and y coordinates of points A to F.

The second line contains 4 integers $L_{11,x} L_{11,y}, L_{12,x}, L_{12,y}$ denoting two points on L_1. The third line contains 4 integers $L_{21,x} L_{21,y}, L_{22,x}, L_{22,y}$ denoting two points on L_2.

All the coordinates are in the range $[0, 514]$.

- $1 \leq T \leq 100$

Output

For each test case output 4 numbers, P_x, P_y, Q_x and Q_y, denoting the two chosen points. If there are multiple solutions, output the one with smallest P_x. If there are still multiple solutions, output the one with smallest P_y, and then Q_x, and then Q_y. Otherwise, if there are no solution, output "No solution"(without quote) in one line. A solution with absolute error at most 10^{-6} will be accepted.

Sample Input

```
1
0 10 0 0 5 10 5 0 10 10 10 0
0 0 5 10
5 0 10 10
```

Sample Output

```
3.3333333 6.6666667 6.6666667 3.3333333
```

This page is intentionally left blank.

J. Pizza Islands

Description

In the Pizza Kingdom, there are bridges between some pair of islands. Moreover, there is exactly one path between every two different islands in the kingdom.

For the Pizza Festival, the people in each island vote for their most favorite pizza, and give the pizza a number of deliciousness. Now, the king wants to march from one island to another island, and stay on some of the islands on the path to have a taste of their favorite pizzas. The only constraint is that the deliciousness of these pizzas must be strictly increasing, so that the king will become happier and happier along the march. Your job is to find the most suitable plan, including the islands to start with, to end with, and to stop by, such that the king will have the most number of pizzas.

Input

The first line of the input file contains an integer T indicating the number of test cases.

For each test case, the first line contains one integer n. The second line contains n integers, and the i-th number indicates the deliciousness number of the most favorite pizza of the i-th island. These n numbers are a permutation from 1 to n. Each of the next $n-1$ lines contains two integers x_i, y_i denoting a bridge connecting island x_i and island y_i.

- $1 \leq T \leq 50$
- $2 \leq n \leq 60000$

Output

For each test case please output the number of pizzas that the king can eat.

Sample Input

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

Sample Output

```
3
5
```