The Fancy New Hard Disk!

Time limit: 1s

Zafar brought a new and huge (not only in memory but also in size!) hard disk to connect to the super computer in the server room of NCS. The hard disk is spherical in shape. The room has enough space to store it. But now we are not sure if he can fit this through the server room's door! Given the length and width of the door and the radius of the hard disk, can you tell us if it will fit through the door?

Input Format

The first line of the input contains an integer T ($1 \le T \le 50$) denoting the number of test cases. The description of T test cases follows.

Each test case contains three integers length of the door L ($1 \le L \le 100$), width of the door W ($1 \le W \le 100$) and radius of the hard disk R ($1 \le R \le 50$). Note that it is not necessary that $W \le L$.

Constraints

Mentioned in the Input section

Output Format

For each test case, output a single line containing "YES" if the hard disk can fit through the door and "NO" otherwise.

Sample Input 0

```
3
2 2 1
10 5 3
5 10 2
```

Sample Output 0

```
YES
NO
YES
```

Explanation 0

Illustration: In first sample input the diameter of the hard disk is 2, which is not larger than the smaller side of the door and so the output is YES. In second sample input the diameter is 6, which is larger than 5, so the output is NO.

Weird Patterns by Roth

Time limit: 1s

Allen was walking by the Roth Pond one day solving a math problem in his head. Suddenly he finds a piece of paper. On one side, he finds this function

$$F(n) = \begin{cases} n, & if\ 0 \le n \le 1 \\ F(n-1) + F(n-2), & If\ n > 1 \end{cases}$$

On the other side, he finds these:

```
For 2, the drawing should be
#..#
#..#
.##.
.##.


For 3, the drawing should be
#......#
#......#
.#....#.
.#....#.
..####..
..####..


For 4, the drawing should be
#..........#
#..........#
.#........#.
.#........#.
..##....##..
..##....##..
...######...
```

```
...######...


For 5, the drawing should be
#................#
#................#
.#..............#.
.#..............#.
..##..........##..
..##..........##..
...###......###...
...###......###...
....##########....
....##########....
```

Now Allen immediately realized the pattern in the drawing for n is related to $F(n)$ and started drawing patterns for other values of n, Do you see how they are related? Can you find out the patterns for other values of n?

Input Format

Input file contains at most 10 lines of input. Each line contains an integer n (2 ≤ n ≤ 20). Input is terminated by a line containing -1. You should not process this line and terminate your program after this.

Constraints

Mentioned in the Input section

Output Format

For each input n, draw the pattern corresponding to it. Print a blank line after the output for each pattern. Sometimes the outputs can be pretty big, so please make sure that you produce output in an optimized way to run within time limit. See the sample output section and follow that format strictly.

Sample Input 0

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

Sample Output 0

```
#..#
#..#
.##.
.##.

#......#
#......#
.#....#.
.#....#.
..####..
..####..

#..........#
#..........#
.#........#.
.#........#.
```

```
..##....##..
..##....##..
...######...
...######...

#................#
#................#
.#..............#.
.#..............#.
..##..........##..
..##..........##..
...###......###...
...###......###...
....##########....
....##########....
```

Walking Around Roth Just Creates Problems!

Time limit: 2s

You already probably know Allen likes walking around Roth while solving problems. This time he again finds a piece of paper and it has again an equation on one side.

F(N) = The product of digits of N

So F(254) = 2x5x4 = 40. F(105) = 1x0x5 = 0.

That is not so interesting. So he decides to create a problem using it.

Given two integers **A** and **B** find the sum: **F(A) + F(A+1) + ... + F(B)**

Input Format

First line of the input contains **T (1<=T<=10000)** which is the number of test cases. Each of the following T lines contain two space separated integers **A** and **B(1<=A<=B<=10^18)**

Constraints

Mentioned in the input section

Output Format

Output the case number, followed by the required quantity. Output the result modulo **1000000007**.

Sample Input 0

```
3
8 12
35 105
1324567086 6545650456
```

Sample Output 0

```
Case 1: 20
Case 2: 1860
Case 3: 105564501
```

Upstate-NYC-Stony Brook

HackerRank

Time limit: 4s

Allen and Shawn decided to meet in New York city. Right now, Allen is in upstate New York and Shawn is in Stony Brook. They cannot communicate with each other for the sake of this problem. They did not decide the time when they should meet beforehand. They both have a list of Amtrak trains from upstate to the city and also a list of LIRR trains from Stony Brook to the city. For every train, the arrival time in the city is given as a range. Formally, there are n trains departing from upstate. For the i-th train, two numbers x_i and y_i are given. It means that the i-th train will need at least x_i hours from now and at most y_i hours from now to reach the city. There are m trains starting from Stony Brook for the city and similar information is given for them too. Now Allen starts thinking how many pairs of trains are there where the first train is from upstate to city and the second is from Stony Brook to city and the ranges of the arrival time of the pair have an intersection.

While calculating this, Allen suddenly notices that the electronic board in the station is showing Q updates. In each update, three numbers $source$, x and y are shown. A value of 1 for the $source$ means that a new train from upstate to the city is added to the schedule whose arrival time is in between x and y. A value of 2 for the $source$ means that a new train from Stony Brook to the city is added. As expected, Allen starts calculating the number of pairs of trains with intersecting range of the arrival times after every update. Can you write a code to help him?

Input Format

The first line contains an integer **T** indicating the number of test cases. For each test case, in the first line an integer **n** will be given indicating the number of available trains from the upstate to the city initially. Then n lines will follow, each for one train. In the i-th line, two integers x_i and y_i are given. It means that the i-th train has an arrival time in between x_i and y_i. After these n lines are given, the next line will contain an integer, **m** indicating the number of available trains from Stony Brook to the city initially.

Then m lines will follow in exactly the same format as the trains above. Then a line will contain an integer **Q** indicating the number of updates. Then Q lines will follow. In the i-th line, three integers $source, x_i$ **and** y_i will be given. The meaning of these three integers are given in the problem description.

Please use faster I/O as dataset is huge.

Constraints

- $1 \leq T \leq 40$
- $0 \leq n, m \leq 10^5$
- $0 \leq Q \leq 10^5$
- $1 \leq x_i \leq y_i \leq 10^9$

$\Sigma\,(n+m+q)$ over all test cases $\leq 1.5 * 10^6$

Output Format

For each case, in the first line print the case number and the number of pairs of trains with an intersecting arrival time range for the initial list of trains. Then print Q lines. The i-th line should contain the answer after the i-th update.

Sample Input 0

```
2
2
1 10
4 12
3
5 13
1 10
90 100
0
1
1 10
2
5 13
90 100
2
1 4 12
2 1 10
```

Sample Output 0

```
Case 1: 4
Case 2: 1
2
4
```

Can Disco Lights Distract Contestants Enough?

Time limit: 3s

Kenny is in charge of decorations for the ICPC contest this year.He decides to hang some disco lights in the rooms and of course as a contestant, writes a code for changing the color of lights every minute using some rules.

The rooms are connected through some passages. You need to determine the final color of the lights in each room following Kenny's rules. Kenny wants you to go through few definitions and terminology before going into the rules

- color(u,t) – this denotes the color of room u at minute t.
- adjacent(u,k) – this denotes a set of rooms. Say v is an element of this set. It must be true that, the shortest path between u and v is requires exactly k passages to be used.
- reach – this is an integer value. This denotes how far a a room can reach to get its color.

Now, the rule for coloring. Suppose, current time is t. Then,

$$\text{color}(u, t+1) = \text{color}(u, t) + \sum_{v \in \text{adjacent}(u, (t \bmod \text{reach})+1)} \text{color}(v, t)$$

Now given initial color(u,0) of the nodes and P, you need to find the color of every room after P minutes.

Input Format

First line of input contains a single integer, T (T ≤ 15), the number of test cases. T test cases follows. Each consists of an integer N (1 ≤ N ≤ 100), number of rooms, M (0 ≤ M ≤ N*(N-1)/2), number of passages, R (1 ≤ R ≤ 10), reach of the rooms and P (0 ≤ P ≤ 10^9), the time in minutes. Next line contains N integers, the value of color(u,0) for every room in the order 0 to N-1, 0 ≤ color(u,0) < 2^31. Next M lines each consists of two integers U and V (0 ≤ U, V< N), which denotes there is an undirected passage between U and V.

Constraints

Mentioned in the Input section

Output Format

For each case print one line "Case X: " (without the quotes), where X is the case number. Next print N numbers denoting color(u,P) where u is from 0 to N-1 in the same line. As color(u,P) can be large, print modulo 1000000007.

Sample Input 0

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

Sample Output 0

```
Case 1: 90 102 129 102 90
```

Explanation 0

Here,
adjacent(0,1) ={1}, adjacent(0,2)={2}, adjacent(0,3)={3}
adjacent(1,1) ={0,2}, adjacent(1,2)={3}, adjacent(1,3)={4}
adjacent(2,1) ={1,3}, adjacent(2,2)={0,4}, adjacent(2,3)={}
adjacent(3,1) ={2,4}, adjacent(3,2)={1}, adjacent(3,3)={0}
adjacent(4,1) ={3}, adjacent(4,2)={2}, adjacent(4,3)={1}

So, color of each room after each minute is given below:

- 0 min => 1 2 2 1 3
- 1 min => 3 5 5 6 4 (colors come from adjacent(u,1))
- 2 min => 8 11 12 11 9 (colors come from adjacent(u,2))
- 3 min => 19 20 12 19 20 (colors come from adjacent(u,3))
- 4 min => 39 51 51 51 39 (colors come from adjacent(u,1))
- 5 min => 90 102 129 102 90 (colors come from adjacent(u,2))

Mandatory Problem about Cats

Time limit: 1s

Astra has N baskets, each having X kittens in them. The i-th basket has A_i orange kittens and $X - A_i$ brown kittens. Astra opens each box and picks up one kitten randomly from each box. She can open boxes in any order and the probability of any kitten being chosen is equal. What is the probability that all the kittens she picks are brown?

Input Format

The first line of the input contains an integer T (T ≤ 10) denoting the number of test cases. Each of the following T cases starts with a pair of space separated positive integers N and X where N is the number of baskets and X is the number of kittens in each of the N baskets. The following line has N space separated positive integers, where the i-th of them A_i denotes the number of orange kittens in the i-th box. Note that, 1 ≤ X, N ≤ 100000. Number of orange kittens in any of the boxes doesn't exceed X.

Constraints

Mentioned in the Input section

Output Format

For each test case, print the output in the format, "Case C: (A % M)/(B % M)" (quote for clarity), where A/B is the answer for the test case, A, B are relative prime, C is the case number and M = 1000000007. If A/B is 0 or 1, simply print the integer. Here % means the modulus or modulo operator.
For exact output format please check sample input/output section. Please note that A/B is the actual answer without considering the modulus operation.

Sample Input 0

```
3
2 2
0 1
3 10
```

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

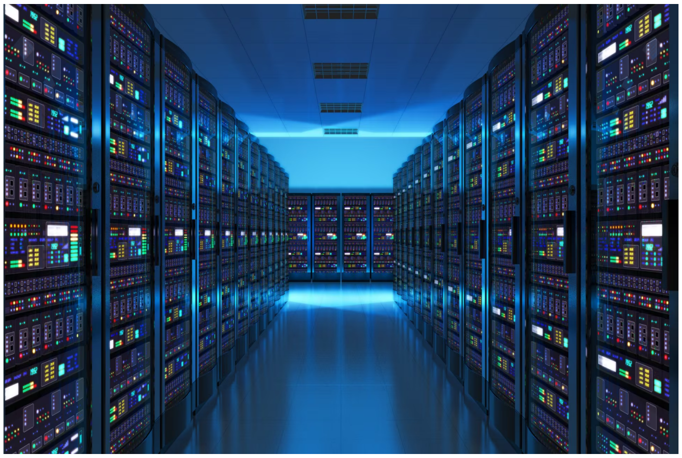
Sample Output 0

```
Case 1: 1/2
Case 2: 63/125
Case 3: 0
```

SBU Campus Network

locked

Time limit: 2s



Mike is in charge of SBU's new campus network project. He decides to deploy servers on some nodes in the network to serve all other nodes. A node can access any server connected to it in the network. However, the nodes are prone to failure or corruption. Therefore, Mike wants to know the minimum number of servers needed to serve the entire network when at most one node in the network is unavailable. Mike can decide where to deploy the servers. Since there may be many ways for deploying the minimum number of servers, Mike also needs to find the number of ways. Can you solve the problem like Mike did? The number of input data guarantees the number of ways is less than 2^64.

Input Format

The input includes multiple test cases.

The first line of each test case consists of the number of edges m. As a termination signal, if the number is 0, it means there is no more test case.

The 2nd line to $m+1$-th line indicates an edge of the network. Each line includes two integers, which indicates the nodes connected by the edge.

Constraints

$2 <= m <= 500$.

The indices of nodes are between 1 and 500.

The number of test cases is less than 10.

Output Format

For the i-th test case, output one line started by "Case i: ". Then output the minimum number of servers and the total ways to deploy that number of servers.

Sample Input 0

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

Sample Output 0

```
Case 1: 2 4
Case 2: 4 1
```

Explanation 0

For case one, the servers can be deployed at (2, 4), (3, 4), (4, 5), (4, 6).

For case two, the servers can be deployed at (4, 5, 6, 7).

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

Peace Maker

HackerRank

Time limit: 2s

One day, Yimin and Tanzir engage in a heated argument over the phone. Later, they both decide to meet and confront each other face to face. Jiarui, who is at Tanzir's place, does not have a phone and wants to stop the fight. He takes a car and starts driving towards Yimin's place. At the same time, Yimin and Tanzir start walking angrily towards each other's place. Jiarui meets Yimin and gives her Tanzir's message, takes her message, then turns back and starts driving towards Tanzir's place. He meets Tanzir on the way and gives Tanzir Yimin's message. After this, he parks there and walks Tanzir back. Thanks to Jiarui's effort, Yimin and Tanzir reconcile as friends. You can assume no time is needed for exchanging messages and Jiarui turning his car around. Suppose everyone is always at constant speed and walks/drives in a straight line. Can you calculate the distance Jiarui drives in total?

Input Format

A single line will contain four non-negative single-space separated floating point numbers representing the distance *d* between Yimin's and Tanzir's house, Yimin's walking speed *v_y*, Tanzir's walking speed *v_t*, and Jiarui's driving speed *v_j*.

Constraints

0 < *d* <= 100, 0 < *v_y*, *v_t* < *v_j* <= 60

Output Format

Output a single number (rounded to the third decimal) specifying the length Jiarui drives in total.

Sample Input 0

```
10 2 2 3
```

Sample Output 0

```
7.200
```

Sample Input 1

```
1 1 1 4
```

Sample Output 1

```
1.280
```

Sample Input 2

```
92 11 2 32
```

Sample Output 2

```
128.876
```