Problem A. Adventure on the space

Source file name:	`A.c, A.cpp, A.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI Guadalajara`

Professor JJ has developed the most powerful space travel machine, this machine has the ability it can go from planet A to planet B in just a few miliseconds if B is in the spatial range from A. The spatial range in a planet is given by an integer D, and it is said B is in the spatial range of A if the distance between the planets is at most D.

As some of you are aware JJ prefers always to go in the most secure way, this is why he is developing what he calls the most secure grid of planets. The most secure grid of planets is created selecting pairs of planets where you could travel between those planets using JJ's space travel machine and also it follows these restrictions:

- All planets in the universe should be in the grid of planets.

- For any pair of planets A and B there is only one way to reach B from A using JJ's space travel machine

You always argue with JJ, and now you two are arguing that he should not call his grid "the most secure grid of planets" as there can be more than one way to create it or there may not be a way to create the grid at all. JJ will not believe you until you get some evidence. This is why he just challenged you (as always) given the number N of planets in the universe, the value D that defines the spatial range of a planet and the coordinates where the planets are in the universe, can you determine how many different "secure grids" JJ can create? A "secure grid" A differs from other "secure grid" B if there is at least one pair of planets in the grid A that is not in the grid B.

Input

The input consist of several test cases. Each test case begins with a line containing two numbers N and D. The next N lines contains the values x_i, y_i separated by a space the coordinates of each planet. The end of the test cases is given by a case where N and D equals 0.

- $2 \leq N \leq 10$

- $1 \leq D \leq 10^3$

- $0 \leq x_i, y_i \leq 10^3$

Output

For each test case print in one line the number of different "secure grids" JJ can create. As this number can be very large please print it modulo $10^9 + 7$

Example

Input	Output
3 5	3
1 1	0
2 2	
1 2	
3 1	
1 1	
2 2	
1 2	
0 0	

Explanation

In the first case, there are 3 planets and you can reach any of them from any other. Let the planets be 1,2,3 , you can select the pairs to create 3 different "secure grids" : 1,2,2,3, 1,2,1,3, 1,3,2,3. In the second test case there is no way to select pairs in such a way that all the requirements for a "secure grid" are met.

Problem B. Balloons

Source file name:	B.c, B.cpp, B.java
Input:	Standard
Output:	Standard
Author(s):	Félix Arreola - CUCEI Guadalajara

This year CUCEI's programming contest is so easy that Félix believes every team will solve all the problems, these are good news for you but not for Félix. As you may know each problem has assigned a color, once a team solves that problem a balloon with that color is put in the place where the team who solved the problem is working, since all the teams will be solving all the problems Félix needs to get each balloon color as much times as teams registered in the contest.

The budget for the contest is very low and Félix will feel so bad if a team can't have all the balloons of the problems the team solved. This left Félix with no choice but to limit the number of teams that will be allowed in the contest with a simple rule. If Félix can buy only X balloons of each color with the budget Félix has then only X teams will be allowed to register on the contest.

Given the budget B Felix has, the number N of problems in the set and the cost of each color of balloon, can you determine what is the maximum number of teams that Félix can allow in the contest?

Input

The input consist of several test cases. Each test case begins with a line containing two numbers N and B. The next line contains N numbers separated by a space, the cost of each of the balloon colors. The end of the test cases is given by a case where N and B equals 0.

- $1 \le N \le 100$

- $1 \le B \le 5000$

- The cost of each balloon p_i is in the range $1 \le p_i \le 500$

Output

For each test case print in one line the maximum number X of teams that Félix can allow in the contest.

Example

Input	Output
3 100 1 2 3 2 1000 1 2 0 0	16 333

Problem C. Cable Way

Source file name:	`C.c, C.cpp, C.java`
Input:	Standard
Output:	Standard
Author(s):	Gilberto Vargas - CUCEI Guadalajara

Over the mountains there is a system of basements for a secret research facility. There are many researchers that like to do all their work because it is a quiet place and also the weather is very nice since the mountains create a shadow that covers the sunlight.

A big problem is that there are no roads and helicopters can not get into the mountains, this is why they are planning to build a net of cableways to travel faster between the facility.

The terrain on the facility consists of a series of mountains that vary on height and distance. All the facilities are aligned on a line and there are also one on the edges, those two are the tallest always. Each facility is connected to at least another two facilities. The plan is to connect each one to the next facility that is taller to the left and the next taller to the right. Also, since the two at the edges are the tallest, those don't have a facility which needs to be connected directly (just have connections from lower facilities).

Given the list of facilities, the distance in meters to the leftmost and height of each facility, can you calculate the number of facilities to which each facility is connected and the distance that each cable will have?

Input

The input begins with a line with a single number N that tells how many facilities are there excluding the two at the edges. Each of the next $N + 2$ lines contains two numbers x_i and h_i, where x_i is the distance to the lsftmost facility and h_i is the height of the facility. The first facility is always located at $x = 0$ and the first and last are the same height and also the tallest. There are no two facilities at the same distance and facilities are given in increasing order of x_i in the input.

- $1 \le N \le 5 * 10^5$

- $1 \le B \le 10^8$

Output

For each test case you must print N lines with 4 numbers each, l_i, r_i, d_l, d_r, where l_i and r_i are the indexes of the facilities to which the facility i is connected to the left and to the right respectively (the first facility has index 0). d_l and d_r are the lengths of the cables that connect the facilites rounded to 4 decimas.

Example

Input	Output
4	0 5 2.8284 8.2462
0 10	1 4 6.3246 7.2111
2 8	1 4 7.2111 6.3246
4 2	0 5 8.2462 2.8284
6 2	
8 8	

Problem D. Dios Primes

Source file name:	D.c, D.cpp, D.java
Input:	Standard
Output:	Standard
Author(s):	Mario Diaz - Queen's University & Juan Pablo Marín

In number theory, a left-truncatable prime is a primer number which, in a given base, contains no 0, and if the leading ("left") digit is successively removed, then all resulting numbers are prime. For example, 9137 is a left-truncatable prime, since $9137, 137, 37, 7$ are all prime. For this problem, decimal representation is assumed.

A Dios prime is a left-truncatable prime such that the number obtained by reversing its digits is also left-truncatable. The number 3467 is a Dios prime, since 3467 is left-truncatable and 7643 is also left-truncatable. Given two numbers a and b, your task is to find the number of Dios primes in the range $[a, b]$.

Input

The first input line contains a number T, the number of test cases. Each of the following T lines contains two numbers a and b separated by a space.

- $1 \leq T \leq 10^5$

- $1 \leq a \leq b \leq 10^9$

Output

For each test case you must print the number of Dios primes in the range $[a, b]$.

Example

Input	Output
4	4
1 10	2
11 100	1
3467 3467	1
7643 7643	

Explanation

Side Remark. In order to dial *DIOS in a telephone, one has to dial *3467. That's how the Dios primes got their name.

Problem E. Endless Sum

Source file name:	E.c, E.cpp, E.java
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marín - CUCEI Guadalajara

You are given a list of N positive integer numbers and then proceed to sum each of them in the order the numbers were given, once you reach the end of the list you will go to the first number and proceed to sum the numbers again.

At each step after you sum each number you will have a total sum S, can you determine if at any point the sum S will be a multiple of the number K ?

Input

The first line of input contains a number T, the number of test cases. Followed by T test cases, each test case contains in the first line the numbers N and K separated by a space, followed by a line with N numbers separated by a space, the list you will take to sum the numbers.

- $2 \le T \le 100$

- $1 \le N < 10^5$

- $1 \le K \le 10^9$

- Each number A_i will be in the range : $1 \le A_i \le 100$

Output

For each test case you must print a line with the string "Yes." if in some point the sum will be a multiple of K, print "No." otherwise.

Example

Input	Output
3	Yes.
1 5	Yes.
1	
3 11	
1 5 10	

Explanation

In the first case, after the first number is summed 5 times you will get a multiple of 5.

For the second case, Once you summed all the numbers your total sum is 16, you sum the first number again and the total sum is 17, next you sum the second number and the total sum is 22 which is a multiple of 11.

Problem F. Funny String

Source file name:	`F.c, F.cpp, F.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Juan Pablo Marín - CUCEI Guadalajara`

A string is traditionally seen as a sequence of characters. Based on this definition of string, different types of strings can be defined, sorted strings for example are strings where for any consecutive pair of indexes i and j that you take in the string such that $i+1 = j$ when comparing the numeric values of the characters on those positions S_i and S_j then $S_i \leq S_j$.

Let's now define a funny String. Funny strings are similar to a sorted string, just that they have a Funny factor F such that for any consecutive pair of indexes i and j that you take in the string such that $i+1 = j$ when comparing the numeric values of the characters on those positions S_i and S_j then $S_i + F \leq S_j$.

Given a string S of lowercase characters. Can you determine what is the longest funny string that you can get by removing some characters from S ? For the input characters you can assume that the next character after "z" is "a", in such way if $S_i = z$ then $S_i + 1 = a$, $S_i + 2 = b$ and so on.

Input

The first line of input contains a number T, the number of test cases. Followed by T test cases, each test case contains a single line with the string S and the Funny factor F separated by a space.

- $2 \leq T \leq 100$

- $1 \leq F \leq 26$

- $1 \leq |S| \leq 10^3$

Output

For each test case you must print a line with a single integer. The length of the longest funny string that can be created removing some characters from S.

Example

Input	Output
2	26
abcdefghijklmnopqrstuvwxyz 1	3
zabd 2	

Explanation

In the first case the given string is a funny string itself, therefore no characters should be removed.

In the second case you can remove the second character and the new string is a funny string with size $|S| - 1$.

Problem G. General pump system

Source file name:	`G.c, G.cpp, G.java`
Input:	`Standard`
Output:	`Standard`
Author(s):	`Félix Arreola - CUCEI Guadalajara`

The mexican company ACM (Agua Contaminada con Microorganismos) uses a complex water pump system to clean water for the city. The company needs to replace a lot of old water pumps that have failures, they have removed the failing pumps from the map and left only the pumps that did not require a replacement.

Once the replacements were ready ACM found they did not know the place for each of the replacements as they missed to save the original map, that's why they need your help to find the way the pumps should be placed in the map.

The map is a square of size N, each cell can have maximum 1 pump, the pump system should follow some rules that should be followed carefully in order to avoid having problems with the resultant pump system:

There are 4 types of pumps (numbered from 1 to 4). A pump with number 1 needs to be connected with exactly 1 pump. A pump with number 3, needs to be connected with exactly 3 pumps, in general a pump with number n needs to be connected with exactly n pumps. Two pumps are connected if they are adjacent (share one side) in the map. If a pump does not meet this condition then the water will leak and won't be properly cleaned. For example, the following are valid configurations.

The pumps connections should be done carefully, if two connected pumps have the same number, they will explode. The following is an example of invalid connected pumps:

Your task is to complete the pump system, using the minimum water pumps and connecting every pump as described above. You should not move any of the working pumps (i.e the pumps given in the map).

Input

The input file contains several test cases, the first line contains an integer T the number of test casses. Following T test cases for each test case the first line contains an integer M indicating the size of the room system. The room is always a square. There are M lines following. Each line contains exactly M characters, a dot "." represents an empty space and a number (from 1 to 4) represents an already set pump. The input finishes on the end of file.

- $1 \le T \le 15$

- $5 \le M \le 8$

Output

For every test case you should output the full room, with every pump connected according to the rules. It is guaranteed that the solution exists and is unique.

Example

Input	Output
2 5 .1... ....3 .4.4. 2.42. ..1.. 6 1...2. ...4.1 3.4... ...2.1 1.2... .1...2	.1..1 23.23 34342 2342. ..1.. 1.232. 2.3431 3243.. 2.32.1 1.2.23 .13232

Problem H. Hidden card trick

Source file name:	H.c, H.cpp, H.java
Input:	Standard
Output:	Standard
Author(s):	Gilberto Vargas - CUCEI Guadalajara

There is a guy that comes to CUCEI and do some magick tricks, this is his way to make money and keep studying. A lot of people have seen his magic tricks but nobody know exactly if he really studies or not. His favorite tricks are always with a deck of cards and nobody have ever found how does he do all these sorceries.

One of the tricks that I can remember is that one day he asked a friend to choose a card without letting him see it, then he shuffles the deck and tries to guess what card my friend picked, He guessed incorrectly three times with different cards and when we thought he had failed he asked us to move one step back to give him some space, when we moved back the card was under my friend's foot. Amazing!

Another trick that he does is that he asks you to take one card and to remove it from the deck, then he takes the card and places it in some random position in the deck. After the card is placed he will take the first card and put it at the bottom of the deck, and discard the next card, he continues alternating until only one card is left. As you may guess the last card is the one that was chosen at the beginning of the trick.

Can you guess how did he do this magic trick?

Given a number N representing the size of the deck determine in which position you should place the card so it will be the last card after when doing the trick.

Input

The input consists of several test cases, one case per line. Each test case is described by a single line with a single number N.

- $1 \le N \le 10^{15}$

Output

One line for each test case telling the position in which the card should be placed. The input finishes with a line containing a zer, this should not be processed.

Example

Input	Output
1	1
7	7
14	13
8	1
0	

Explanation

For the case for 7 cards:

- 1 goes to bottom

- 2 is discarded

- 3 goes to bottom

- 4 is discarded

- 5 goes to bottom

- 6 is discarded

- 7 goes to bootm

- 1 is discarded

- 3 goes to bottom

- 5 is discarded

- 7 goes to bottom

- 1 is discarded

- 7 is the last card, so 7 should be the position to place the card.

Problem I. Intersecting edges

Source file name:	`I.c, I.cpp, I.java`
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marin - CUCEI Guadalajara

Last data structures lecture was about graphs, now you can't stop seeing a graph on everything. You say everything is a graph and maybe there is some of true on it, however your obsession with graphs has taken you to imagine some scenarios, one of those scenarios you are interesting about is the number of edges that intersect when drawing a specific type of graph.

You take a graph with N nodes and draw all the nodes on a line numbered from 1 to N. Next you will draw M edges selecting two nodes from the graph a and b and draw the edge starting in a and finishing in b drawing it above all nodes between a and b. The following is an example of the graph where $N = 10$.
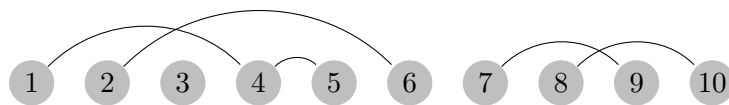
As you can see this graph has two intersections.

- The edge that goes from 1 to 4 intersects with the edge that goes from 2 to 6.

- The edge that goes from 8 to 10 intersects with the edge that goes from 7 to 9.

Given a specific graph, can you determine how many intersections will be in the drawing?

Input

The first line of input contains a number T, the number of test cases. Each test case will begin with two numbers separated by a space N and M the following M lines of the test case will contain two integers separated by a space a and b representing the edges that will be drawn. No pair of edges will start or end on the same node.

- $1 \leq N \leq 10^9$

- $1 \leq M \leq 10^5$

- $1 \leq a < b \leq N$

Output

For each test case your program must output a line with a single number, the number of intersections that the drawing will have.

Example

Input	Output
2	2
10 5	0
4 5	
1 4	
2 6	
7 9	
8 10	
10 5	
1 2	
2 3	
3 4	
4 5	
5 6	

Explanation

The input contains two test cases. The first test case is the case for the image shown above. In the second test case even when some pairs share a node no edges intersect.

Problem J. Jumping Frog

Source file name:	`J.c, J.cpp, J.java`
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marin - CUCEI Guadalajara

On Chapala lake there are N stones numbered from 1 to N, there is a male frog resting on stone X and a female frog resting on stone Y. Some research has shown that the frogs in Chapala are smarter than an average frog, and these studies have shown they have knowledge on some of the math areas. A female frog can only be impressed by a male frog that show knowledge on the most math areas in the lake, this is why male frogs in Chapala lake do a strange ritual to impress females, this ritual is as follows:

- To show numbers knowledge the male frog sits on a stone and selects a prime number p and jump p stones at a time until it reaches the female frog, then the frog will return to the stone where it started jumping and select another prime number that allows it to reach the female,if such number does not exist the male stops.

- To show sets theory knowledge, the male frog will take a subset of the stones that are in the range that goes from X to $Y - 1$ and show his number knowledge from each of the stones in the subset.

- To show combinatorics knowledge, the male frog will do the previous for all the different subsets it can take in the range X to $Y - 1$.

As you can see the male frog will reach the female several times during the ritual.

Can you help researchers to determine the number of times the male frog will reach the female knowing the stones where both frogs are resting before the male starts the ritual?

Input

The input contains several test cases, each test case contains three numbers separated by a space N, X, Y. The end of input is given in a case where all values are equal to 0, and should not be processed

- $2 \le N \le 10^6$

- $1 \le X < Y \le N$

Output

For each test case your program must output a line with a single number, the number of times the male frog will reach the female frog. As this number can be large print it modulo $10^9 + 7$

Example

Input	Output
5 1 3	2
0 0 0	

Explanation

There is only one test case in the input. There are 5 stones in the lake, the male frog rests at stone 1 and the female frog rests at stone 3. There are a total of 3 different subsets the male frog will select to impress the female frog : $\{\{1\}, \{2\}, \{1, 2\}\}$. In the first set the male frog can reach the female frog jumping from stone 1 to 3 using the prime number 2, in the second subset the male frog can not select a prime to reach the female frog, in the third subset the male frog can reach the female frog only from stone 1 using the prime number 2. In total the male frog will reach only 2 times the female frog.

Problem K. King of music

Source file name:	`K.c, K.cpp, K.java`
Input:	Standard
Output:	Standard
Author(s):	Juan Pablo Marin - CUCEI Guadalajara

The king of music has lost his composition validation device. The composition validation device shows the king of music if a given composition will be a huge hit or not, he has explained to us how a composition is known to be a hit. A composition can be seen as an arrangement of the intensity of the tones on the time, there are some "crescendos" and "decrescendos". A "crescendo" is a gradual increase in the sound intensity while a "decrescendo" is a gradual decrease in the sound intensity over some time lapse in the composition.

The secret of the king of music is that he always composes his music in such a way that if you take the intensity at any given time t then none of the intensities in the lapse from time $t - W$ to $t - 1$ are greater or lower to K units when compared to the intensity in t. This method is why he earned the "King" title, however, since his composition validation device is lost he is unable to determine if a random composition will be a hit or not, that's why he asks you to write a program to replace his device, your program should validate a given composition and print if it will be a hit or not, i.e it complies with the rule mentioned above.

Input

The input contains several test cases, each test case starts with three numbers separated by a space N, W, K. Representing the number of intensities the composition have, and the values of W and K as mentioned above. The next line will contain the N intensities separated by a space. The end of input is given in a case where the values for N, W and K are equal to 0, and should not be processed

- $1 \leq N \leq 10^6$

- $1 \leq W \leq N$

- $1 \leq K \leq 10^6$

- Intensity in each time C_i will be in the range : $1 \leq C_i \leq 10^6$

Output

For each test case your program must output a line with the string "Yes." if the given composition complies with the king of music rules, you must print "No." otherwise.

Example

Input	Output
5 1 5	Yes.
1 6 7 6 7	No.
5 2 5	
1 6 7 6 7	
0 0 0	

Explanation

In the first test case you must look only to the previous intensity and none of the intensities in the lapse for all possible lapses should be less than or more than 5 units. All time lapses comply.

In the second test case you must look to the two previous intensity for any intensity and none of the intensities in the lapse for all posible lapses should be less than or more than 5 units. The lapse 1 6 7 does not comply as 1 is more than 5 units less than 7.

Problem L. Lowest Terms Fractions

Source file name:	`L.c, L.cpp, L.java`
Input:	Standard
Output:	Standard
Author(s):	Gilberto Vargas - CUCEI Guadalajara & Juan Pablo Marín

Mr. Homft's today lecture is about fractions. Fractions are the numbers that describes the *ratio* between values. For example, the *Golden Ratio* is described by the fraction $\frac{fib(n+1)}{fib(n)}$, which tends to be 1.61803. Note that $fib(n)$ is the nth term of fibonnaci's sequence.

Mr. Homft explained also the types of fraction there are. The proper fractions are the ones of the form $\frac{a}{b}$ where $a < b$, for example $\frac{3}{4}$, $\frac{5}{7}$ and $\frac{1}{8}$ are proper fractions. Improper fractions are the ones that are not proper fractions or the ones where $a \geq b$, like $\frac{6}{5}$ or $\frac{8}{7}$.

After explaining all that, he mentioned that a good mathematician always simplifies the fractions he writes, for example, $\frac{4}{8}$ should never be written, because it can be simplified into $\frac{1}{2}$, these fractions are known as irreducible fractions or lowest terms fractions.

After all that was explained he wrote the homework:

Given a number n, how many fractions of the form $\frac{x}{n}$ are there which cannot be simplified, and what is their sum?

Input

The input starts with a line containing a single integer T the number of test cases. Each of the next t lines contains a single number n.

- $1 \leq T \leq 10^4$

- $2 \leq N \leq 10^9$

Output

Your program should output a line per test case with two numbers separated by a space, the first one being the number of lowest terms fractions of the form $\frac{x}{n}$, the second one the sum of these fractions rounded to 4 digits.

Example

Input	Output
4	1 0.5000
2	2 1.0000
3	2 1.0000
4	4 2.0000
5	

Explanation

In the first test case there is only one lowest term fraction on the form $\frac{x}{n}$ being $\frac{1}{2}$, the sum is then $\frac{1}{2} = 0.5000$. For the last test case there are 4 lowest terms fractions: $\frac{1}{5}, \frac{2}{5}, \frac{3}{5} \frac{4}{5}$ and their sum is $\frac{1}{5} + \frac{2}{5} + \frac{3}{5} + \frac{4}{5} = \frac{10}{5} = 2$