Problem A. Problem Select

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

HOJ is an online judge system for students in HIT to practice. As the administrator, you would always setup contests by selecting problems from the problem set. Problem IDs are integers that can be phased by url. For example, the corresponding problem ID of url `http://acm.hit.edu.cn/problemset/1001` is 1001.

Now the task is, given n urls, print the smallest k problem IDs in increasing order.

Input

The first line of the input contains one integer T $(1 \leq T \leq 10)$, indicating the number of test cases.

For each test case, the first line contains two integers, n $(1 \leq n \leq 1000)$ and k $(1 \leq k \leq n)$.

For the next n lines, each line contains a string indicating the problem's url. The problem ID is guaranteed to be in range $[1, 10000]$ and unique in each case.

Output

Output T lines.

For each test case, you should output k numbers in a line, and there shouldn't be any spaces at the end of the line.

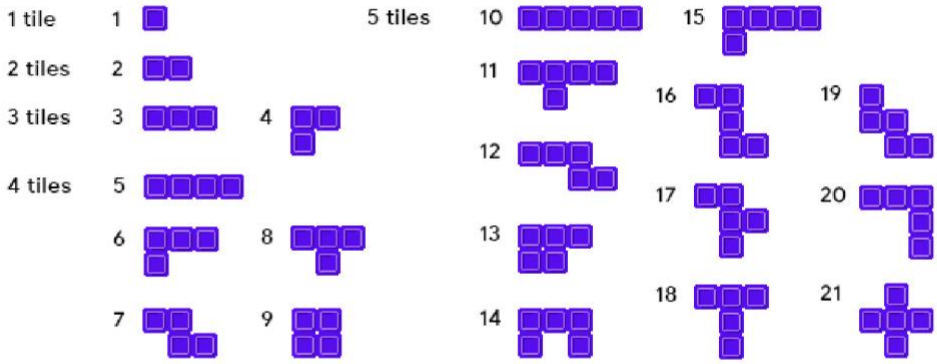
Example

standard input	standard output
2 3 2 http://acm.hit.edu.cn/problemset/1003 http://acm.hit.edu.cn/problemset/1002 http://acm.hit.edu.cn/problemset/1001 4 1 http://acm.hit.edu.cn/problemset/1001 http://acm.hit.edu.cn/problemset/2001 http://acm.hit.edu.cn/problemset/3001 http://acm.hit.edu.cn/problemset/501	1001 1002 501

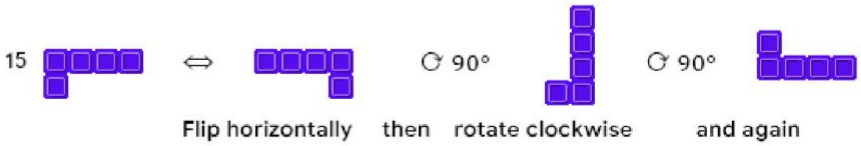
B. Blokus Duo

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Blokus Duo is a 2-player version of the Blokus board game. The game is played on a 14×14 grid. The rows are numbered from 1 to 14 from top to bottom. The columns are numbered from 1 to 14 from left to right. The game uses two (purple and orange) sets of 21 different pieces.



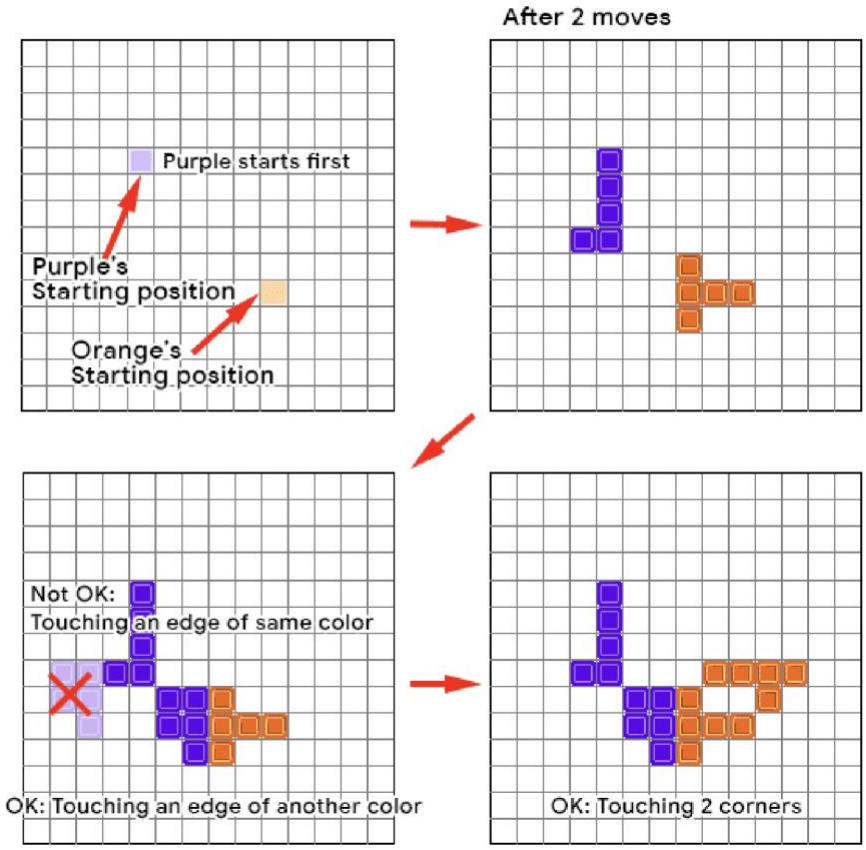
Pieces can be flipped and/or rotated to produce different shapes, as illustrated below:



The two players take turn to place pieces. The player that has the purple pieces goes first. First, he/she must place a piece such that it covers the cell $(5, 5)$. The other (orange) player goes next. He/she must place a piece on empty cells such that it covers the cell $(10, 10)$.

After that, each piece must be placed such that:

- It is placed on entirely empty cells.
- It touches at least 1 corner of 1 or more existing piece(s) of the same color.
- It does not touch any edge of any existing piece of the same color.



If a player cannot make any valid move, he/she must surrender so that the other player may continue placing his/her remaining pieces until he/she cannot do so. The game ends when both players are out of pieces or surrenders. Note that once a player surrenders, even if by mistake (when there existed valid move(s) but the player thought that there wasn't any), he/she cannot make any more moves.

You are doing a research about the strategies so you collected some end game states (the state of the grid). Can you write a program to generate a valid sequence of moves to recreate the end game state?

Input

The input consists of a 14×14 grid that consists of `.` (for empty cell), `P` for a cell that is covered by a Purple piece, or `O` for a cell that is covered by an Orange piece. It is guaranteed that it is a valid end game state and both players placed at least 1 piece. Which means that all rules described above are strictly followed, but it's not guaranteed that no player can make no more moves, because both players could have surrendered early by mistake.

Output

Output any sequence of moves that recreates the given end game state, starting from the empty grid. On the first line output an integer N – the number of moves. The next N lines describe the moves in order. A move should consist of the following the information: `color num_cells row_1 column_1 [row_2 column_2]...`

- `color` is the color of the piece. It should be either `P` (for Purple) or `O` (for Orange)
- `num_cells` is an integer between 1 and 5 inclusive that indicates the size of the piece. (number of cells that it covers)
- After that, output the row and column number of the covered cells in any order using `num_cells` pairs of integers between 1 and 14.

Example

input Copy

```
..............
..............
......PPP.....
.....P.....O..
..PPPP..OO.OO.
......PO....O.
.....PPO..OO..
.....POOO.OO..
.....P...O..OO
......PPPO..O.
........PO....
........P..P..
.........PPP..
..............
```

output Copy

```
11
P 5 5 5 5 6 4 6 5 4 5 3
O 3 10 10 11 10 9 10
P 3 3 7 3 8 3 9
O 4 8 11 7 11 7 12 8 12
P 5 6 7 7 7 7 6 8 6 9 6
O 3 9 13 10 13 9 14
P 5 10 7 10 8 10 9 11 9 12 9
O 4 6 13 5 13 5 12 4 12
P 4 13 10 13 11 13 12 12 12
O 5 8 9 8 8 7 8 6 8 8 7
O 2 5 9 5 10
```

C. camelCaseCounting

time limit per test: 1 second
memory limit per test: 1024 megabytes
input: standard input
output: standard output

Camel case is one of the common practices for naming variables in programming, so that phrases, clauses (or even sentences) can be represented as a string with no spaces or punctuation. In camel case, the first letter of each word (except the first word) is written in capital letter. All other letters, including the first letter of the first word, should be written in lower case. For instance, the phrase "camel case" in camel case is `camelCase`, and "to do list" in camel case is `toDoList`. Here are some other examples of valid camel case writing:

- `laSalleCollege` ("la salle college")
- `puiChing` ("pui ching")
- `challenge` ("challenge")
- `oMG` ("o m g")

And here are some examples of invalid camel case writing:

- `ProgrammingChallenge`
- `ILoveCoding`
- `THISproblemISeasy`
- `TLDR`

Given a valid camel case writing S, please find the number of unique substrings that is a valid camel case writing. Here, substring is a contiguous sequence of characters within a string.

Input

The only line contains a string S, which consists of only lowercase letters and uppercase letters. It is guaranteed that S is a valid camel case writing, and $1 \leq |S| \leq 10^6$ (where $|S|$ is the length of S).

Output

Output a single integer, the number of unique substrings that is a valid camel case writing.

Examples

input	Copy
ease	
output	Copy
9	

input	Copy
camelCaseCounting	
output	Copy
130	

input	Copy
oMG	
output	Copy
3	

Note

In Sample 1, the 9 unique substrings that are valid camel case writings: `a`, `e`, `s`, `as`, `ea`, `se`, `ase`, `eas`, `ease`.

In Sample 2, several substrings of valid camel case writings: `elCaseCount`, `aseCou`, `cam`, `e`.

Problem D. Slalom

Input filename:	`slalom.in`
Output filename:	`slalom.out`
Time limit:	2 seconds
Memory limit:	256 Mb

One of the popular ski resorts in Italy hosts a slalom competition. Every participant will go down the mountain on the skis. On every path of the path, the participant will get some number of points. As the participant goes down, we add these points together. The participant with the maximum number of points wins. The mountain is a triangle of integers, representing points for parts of the path. Participants may choose whether to go left or right on every layer as they go down to the next layer. The participants start in the highest number and may finish in any of the lowest.

For example, the mountain can look like this:

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

Find the maximum number of points that one can score.

Input file format

The first line of the input file contains one positive integer $1 \leq n \leq 100$, representing the number of layers. The following n lines contain the description of layers, with i numbers $-100 \leq a_{i,1}, a_{i,2}, \ldots, a_{i,i} \leq 100$ in the i-th row.

Output file format

On the first and only line print the maximum score possible.

Sample tests

slalom.in	slalom.out
4 1 4 3 5 6 7 8 9 0 9	20

E. Hacking

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Largest Security Professional Challenge (LSPC) is a team contest. Each team consists of 3 hackers, and team shall hack as many websites as possible in 4 hours.

This year, LSPC's problemset consists of three websites to be hacked, each of different hacking difficulties. The hacking difficulties of the three websites can be represented as integers: d_1, d_2 and d_3, where $d_1 < d_2 < d_3$. This means that website 1 is the easiest to hack, with difficulty of d_1, followed by website 2 with difficulty d_2. Website 3 is the hardest to hack, with difficulty of d_3.

The team from Hackerland consists of three members: Alice, Bob and Charles, each of different hacking skills. The hacking skills of the three members can also be represented as integers: s_A, s_B and s_C, where $s_A < s_B < s_C$. This means that Alice with skill level s_A is the weakest hacker in the team, followed by Bob with skill level s_B. Charles, as the best hacker in the team, has skill level s_C.

A Hacker i is skillful enough to hack website j if and only if the hacker's skill level is **strictly greater than** the hacking difficulty of the website, i.e. $s_i > d_j$.

As the contest time is limited, each hacker can hack **NO MORE THAN ONE** website during the whole contest time.

Given the hacking difficulties of the three websites, and the skill levels of the three hackers, what is the maximum number of websites that can be hacked?

Input

The first line consists of three integers d_1, d_2, d_3 $(1 \le d_1 < d_2 < d_3 \le 100)$.

The second line consists of three integers s_A, s_B, s_C $(1 \le s_A < s_B < s_C \le 100)$.

Output

A single integer, indicating the maximum number of websites that can be hacked by the team from Hackerland.

Examples

input
```
10 30 50
20 45 100
```
output
```
3
```

input
```
56 78 90
12 34 56
```
output
```
0
```

input
```
31 41 59
26 53 58
```
output
```
2
```

input
```
2 71 82
8 18 28
```
output
```
1
```

input
```
12 22 32
1 10 100
```
output
```
1
```

Note

Notice that in Sample 5, although Charles is skillful enough to hack any of the websites, he can only hack one of them during the whole contest time.

F. Charm Is Not Always Enough

Score: 100

CPU: 0.5s

Memory: 1024MB

One day Munta was sitting under a tree when a fairy appeared before him. Being impressed by his charm, the fairy gave him a machine which can pour some mysterious liquid called “Felix Felicis”. Every time Munta pressed a button on the machine to pour down some of the liquid, exactly M litre of liquid would come down. But he couldn’t find bottles of same sizes! Munta placed each bottle in the machine and kept on pressing the button until the bottle was filled to its complete capacity. But he couldn’t change the bottle when the liquid was pouring down. As a result, some of the liquid overflowed and it got wasted. Since the liquid is very valuable, the fairy didn’t like that at all, and wanted to punish Munta for wasting so much liquid. Given the number of bottles N, amount of liquid coming down at a time M and the capacity of N different bottles C1, C2, C3,..... Cn your job is to inform the fairy how much liquid Munta wasted so that the fairy can punish Munta accordingly.

Input

The first line will contain an integer T (1<=T<=100) representing the number of test case.

For each test case, there will be two integers N and M. (1 <= N<=10^5 and 1<= M<=10^5).

Then there will be N integers representing the capacity of the bottles (1 <= Ci <= 10^5).

Output

For each test case, output the total amount of liquid wasted by Munta in separate lines.

Samples

Input	Output
1 5 10 11 12 13 15 17	32

Input	Output
2 2 100 100 55 3 100 100 150 250	45 100

G. Just A $10 Note

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Jane is working as a cashier at a grocery store in Hackerland. This grocery store is now offering a special discount, all the items are on sale, each with a positive integral price of no more than $10.

In the early morning, N customers are queuing outside the store waiting for the store to open, but they have not decided what to purchase yet. When Jane arrives, people in the queue all ask the same question: "I have just a $10 note, do you have enough coins for changes?". What a coincidence! Unfortunately, Jane knows that there are no coins in the store, so she has to ask for some from the bank.

In Hackerland, there are only three types of coins, $5, $2 and $1, and they are of equal weight. Jane wants to minimize the number of coins to carry as the coins are too heavy! Jane predicts that no more customers will be coming. Therefore, she should prepare the least number of coins, so that no matter which item(s) eventually each of the N customers choose to purchase (of course, not more than $10 per person), she has enough amount of coins for changes.

You may assume that when Jane gets back to the store again, every customer has decided what to purchase so she will be able to know the amount of changes needed for every customer, then decide how to distribute the coins for giving the changes. Please note that she must give the exact amount of changes to every customer.

Input

A single integer N, denoting the number of customers queuing outside the store ($1 \leq N \leq 50$).

Output

A single integer, denoting the minimum number of coins Jane carries, such that she has enough amount of coins for changes no matter what item(s) the N customers purchase.

Example

input	Copy
1	

output	Copy
4	

Note

There is only one customer, one of the optimal ways is: Jane can prepare four coins: one $5 coin, one $2 coin, two $1 coins. There are ten possible situations:

- The customer purchase items that worth $10 in total, no changes needed!
- The customer purchase items that worth $9 in total, Jane can give one $1 coin back.
- The customer purchase items that worth $8 in total, Jane can give one $2 coin back.
- The customer purchase items that worth $7 in total, Jane can give one $2 coin and one $1 coin back.
- The customer purchase items that worth $6 in total, Jane can give one $2 coin and two $1 coins back.
- The customer purchase items that worth $5 in total, Jane can give one $5 coin back.
- The customer purchase items that worth $4 in total, Jane can give one $5 coin and one $1 coin back.
- The customer purchase items that worth $3 in total, Jane can give one $5 coin and one $2 coin back.
- The customer purchase items that worth $2 in total, Jane can give one $5 coin, one $2 coin and one $1 coin back.
- The customer purchase items that worth $1 in total, Jane can give one $5 coin, one $2 coin and two $1 coins back.

H. Kth number in Byteland

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are thousands of written languages in the world. Some of them are read from left to right, such as English. Some of them are read from right to left, such as Arabic.

In Byteland, programmers come from many different countries. To avoid ambiguity, the mayor of Byteland decided to publish a new written language and also a new number system that remains the same regardless if it is read from left to right or right to left. Moreover, since the mayor loves odd number, he wants the numbers in the new number system consist of odd number of digits. Therefore, the mayor decided that the k^{th} number in the new number system shall be the k^{th} smallest non-negative number in the decimal number system that consists of an odd number of digits and remains unchanged when it is read from either direction.

For example, the 1^{st} number in the new number system is 0, and the 24^{th} number is 232.

As you are a newcomer in Byteland and you are trying to get familiar with the number system, you want to know the k^{th} number in the Byteland!

Input

The input contains only one integer, k ($1 \leq k \leq 10^9$).

Output

Output one integer, the k^{th} number in the new number system in Byteland.

Examples

input	Copy
1	

output	Copy
0	

input	Copy
24	

output	Copy
232	

I. Just One Swap

Score: 100

CPU: 1s

Memory: 1024MB

You are given an array of size N. How many distinct arrays can you generate by swapping two numbers for exactly once? The two selected numbers can be equal but their positions in the array must be different.

Input Specification

The first line of the input contains a single integer T, denoting the number of test cases. Every test case starts with an integer N. The next line contains N integers, the numbers of the array.

Output Specification

For each tescase output the answer in a single line.

Constraints

Subtask 1: 2 <= N <= 50 (15 points)

Subtask 2: 2 <= N <= 100000 (85 points)

In all subtasks:

1 <= T <= 5

1 <= Value of a number in the array <= 100000

Sample

Input	Output
1 5 2 3 2 3 3	7

You can generate the following arrays:

2 3 2 3 3

2 2 3 3 3

2 3 3 2 3

2 3 3 3 2

3 2 2 3 3

3 3 2 2 3

3 3 2 3 2

J. Shaat Chara

Score: 100

CPU: 1s

Memory: 1024MB

Meena and Razu have been playing Shaat Chara for many years. But they think the game is too violent. So, they came up with a new game. In this game, instead of just 1 pile, they start with n piles of stones. The game is for two players. Each player takes turn alternatively. In each turn a player can select any of the piles that has at least 1 stone left, and then remove some stone (at least one) from it. The game goes on like this until there are no stones left. The player who can't make a move loses.

At first, Razu was very happy because there was no chance of injury in this game. But he quickly grew frustrated, because Meena was winning all the games. Meena is very clever. She always plays optimally. That means, if there is a chance to win, she will always win. So, Razu came to you with a particular state of the game. He wants to know how many possible moves he may take such that Meena's win does not become guaranteed.

Input

In a single line, you will be given T, the number of test cases.

T test cases follow. In each test case, you will be given n, the number of piles. Then in a single line, you will be given n numbers ai, the number of stones in the ith (1 <=i <= n) pile.

Subtask 1 - 40 Points

T <= 100

n = 3

1 <= ai <= 50

Subtask 2 - 60 Points

T <= 100

1<=n<=10000

1 <= ai <= 100000

Output

You have to print the case number first in the form “Case 1: ”.

For each case you have to print only the number of winning moves from that particular state of the game. See sample for clarification.

Sample

Input	Output
3 1 1 2 1 1 3 1 1 1	Case 1: 1 Case 2: 0 Case 3: 3

Note that, winning moves mean the moves that will lead Meena to a losing state. For example, if the current state is {1, 3}, Razu can remove 2 stone from the second pile, which is a winning state. Because this move will lead to the state {1, 1}. Then Meena will remove 1 stone from any pile, and Razu will remove the other in the next turn. And Meena will lose the game, because there will be no remaining stone. But if Razu removed only 1 stone from the second pile, that will lead to the state {1, 2}, which is a winning state for Meena. Thus Meena will win.

Also note that, this is a normal nim game of n piles. In a normal Nim game, the player making the first move has a winning strategy if and only if the nim-sum of the sizes of the piles is nonzero. Otherwise, the second player has a winning strategy. Nim sum of n piles is defined as follows.

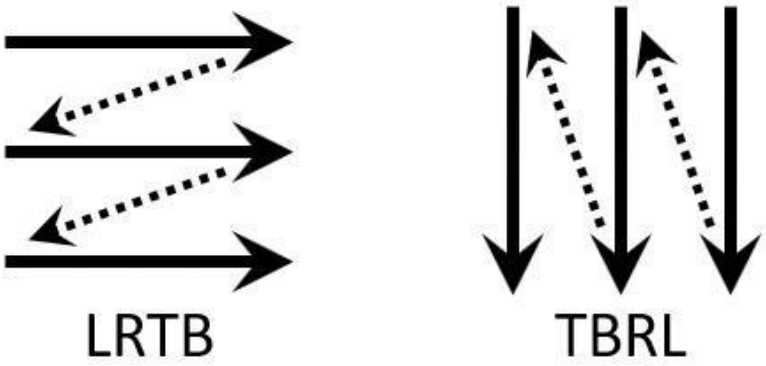
Nim_sum = a1 ^ a2 ^ a3 ^…… ^ an , where ^ is the Bitwise X-OR operation.

Now you might be wondering how it would even work. To put it simply, think of it this way. If the theory is correct, then the first player wins when the nim sum of the current piles are non-zero, and loses otherwise. Now, consider a case where he is in a zero nim-sum position. He must make a change to the piles, which will lead the nim-sum to non-zero, which is a winning state for the opponent. On the other

K. LRTB and TBRL

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Chinese is one of the several languages that can be written and read in two directions: "Left-to-right, top-to-bottom" (LRTB), and "Top-to-bottom, right-to-left" (TBRL).



When reading in LRTB, texts are written row by row, and from left to right in each row. Reading starts from the topmost row to the bottommost row.

When reading in TBRL, texts are written column by column, and from top to bottom in each column. Reading starts from the rightmost column to the leftmost column.

For example, when reading the text above, in LRTB, it is "培正喇沙編程挑戰賽". In TBRL, it is "喇程賽正編戰培沙挑".

Lee Sin has learnt N Chinese characters. He has a grid paper with R rows and C columns, and he wants to write one of the N Chinese characters he knows in each of the $R \times C$ cells. To prevent the two reading directions LRTB and TBRL from bringing readers into confusion, Lee Sin plans to write in a way that readers can read the same content no matter they read in LRTB or TBRL.

Please find out the number of ways Lee Sin can fill the grid paper so that readers can read the same content no matter they read in LRTB or TBRL.

Input

The only line contains three integers N, R and C ($1 \le N \le 4762$, $1 \le R, C \le 1000$).

Output

A single integer, the number of ways Lee Sin can fill the grid paper so that readers can read the same content no matter they read in LRTB or TBRL. As the number may be large, please output it modulo $10^9 + 7$.

Examples

input

```
3 3 3
```

output

```
27
```

input

```
2 3 2
```

output

```
4
```

input

```
4 4 5
```

output

```
256
```

input

```
11 19 31
```

output

```
300916151
```

Note

In Sample 2, assume the $N = 2$ Chinese characters Lee Sin knows are "李" and "星". The four ways he can fill the grid paper are: ("李李李李李李", "李李星李星星" , "星星李星李李"and "星星星星星星")

In Sample 4, the answer is $611590904484145462 91 \mod (10^9 + 7) = 300916151$

L. XOR 'em all!

Score: 100

CPU: 5s

Memory: 1024MB

You are given an array a of n integers, all integers ranging in between 0 and 2^20-1 (inclusive). You are asked to perform m operations on the array in order. All operations are either of the following two types:

1 l r v: You are asked to find a position i, such that i is in between l and r (inclusive) and the absolute difference in parity between a[i] and v is minimum, for all a[l], a[l+1], a[l+2],...., a[r]. Parity of a number is defined as the number of ones in its binary representation. If there are multiple such positions i, output the minimum among them.

2 l r: Update all elements in the range a[l], a[l+1], a[l+2], ... , a[r] by XOR-ing them with the value 2^20-1.

Input

The first line contains the number of test cases: T.

Each test case starts with n, m.

The following line contains n space-separated integers, denoting a[1], a[2], a[3], ... a[n], all within the range 0 to 2^20-1 (inclusive).

The following m lines contain the descriptions of the operations. The i th such line describes the i th operation, and is of either of the two formats:

1 l r v (1<=l r<=n, 0<=v<=2^20-1)

2 l r (1<=l, r<=n)

Output

For each test case, output the case number in a single line as shown in the sample. For each operation of type 1, print an integer in a single line, representing the answer to the query.

Samples

Input	Output
1 6 2 2 5 3 15 6 8 1 1 6 7 1 1 3 1024	Case 1: 2 1

Input	Output
2 4 3 1 5 8 2 2 3 4 1 1 4 0 1 1 2 1 5 1 21 26 16 19 71 1 2 4 71	Case 1: 1 1 Case 2: 2

N. B.

1. I/O files are huge, use faster input-output methods.
2. The second sample is valid only for task 3.

M. Max and Alexis Plan to Conquer the World

Score: 100

CPU: 1.25s

Memory: 1024MB

There is this kitten Alexis. His favorite game is playing football. Then there is another kitten, Max. His favorite game is biting "hoomans". So the first time they met, naturally they couldn't agree on the game they should play together. So after two minutes of arguing (as kittens cannot focus on anything for more than two minutes), Max proposed, "Let's conquer the world of hoomans. They have been there for too long. We are the real kings and queens who should rule the kittenland." Alexis agreed with Max and immediately they started formulating their masterplan. Now they are facing a problem. There are N kittens in kittenland currently. Each year the number of kittens increase by R%. Max has decided that they wouldn't start their conquest before they have at least P kittens in their army. So now they are struggling a bit to find out how many years they have to wait till they can start their conquest. For example, let's say, there are currently 10 kittens in kittenland. If their number increases by 15% each year, then there will be 11.5 kittens after one year. So, if they need 11 kittens to be present in their army, they would need to wait for one year. [Yes, it is very much possible that there can be 11.5 kittens. You know, it's mathematics and stuff.] Now if you can solve the problem for queen Max and king Alexis, they might let you go after they conquer the world.

Input

The first line contains a positive integer T, number of test cases. In each of the following T lines, there will be 3 integers, N, R, and P. (The number of kittens currently in the kittenland, the percentage of growth of population each year and the minimum number of kittens needed for the conquest.)

Output

For each case print the case number and then a single integer which is the number of years, that Max and Alexis have to wait before starting the conquest.

Input Constraints

Subtask 1: (20 points)

1 <= T <= 1000

Subtask 2: (80 points)

1 <= T <= 250000

For both subtasks:

1 <= N <= 10^9

1 <= R <= 100

1 <= P <= 10^19

Sample

Input	Output
3 10 15 11 10 15 12 446390799 24 6505977804605754 59	Case 1: 1 Case 2: 2 Case 3: 99

Input file is very large. Please use faster methods for taking inputs and giving outputs. Avoid using cin/cout.

N. Simple Path

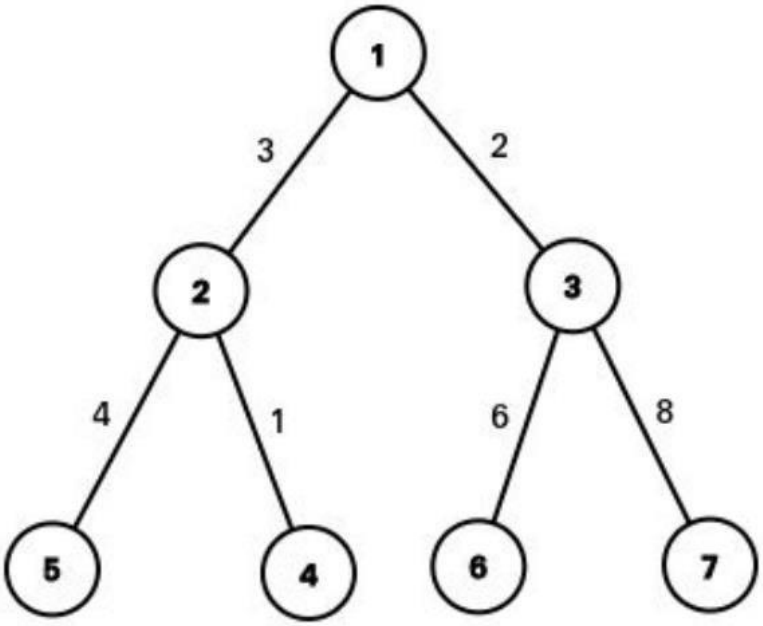
Score: 100

CPU: 1s

Memory: 1024MB

You will be given a weighted rooted tree, where root is node 1. For each subtree of that tree, you will have to compute the summation of lengths of all possible simple paths present in that subtree. Eventually you need to print the summation of those values modulo 1000000007.

Image of the tree in the first sample is given below.



Input Specification:

First line of input will contain an integer T (1 <= T <= 10) denoting the number of test cases. Each test case will begin with an integer N, indicating number of nodes. The following N-1 lines will have three integers each, U (1 <= U <= N), V (1 <= V <= N), W (1 <= W <= 1000000000), which denotes that there is an edge in the tree from node U to node V having W weight.

Input File is large. Use fast I/O methods.

Output Specification

For each test case, print the case number, and then print the answer to the problem modulo 1000000007 in separate lines.

Sample

Input	Output
2 7 1 2 3 1 3 2 2 4 1 2 5 4 3 6 6 3 7 8 6 1 2 3 1 3 2 1 4 4 3 5 7 3 6 1	Case 1: 212 Case 2: 109

Explanation of the 1st sample:

Summation of lengths of all possible sample paths for each subtree is given below:

1: 174

2: 10

3: 28

4: 0

5: 0

6: 0

7: 0

Explanation of the 2nd sample:

Summation of lengths of all possible sample paths for each subtree is given below:

1: 93

2: 0

3: 16

4: 0

5: 0

6: 0