

北京理工大学第十六届“连山科技”程序设计大赛

The 16-th BIT Campus Programming Contest

Sponsored by

连山科技
LSSEC TECH

网络选拔赛
Online Round



题目列表
Problem List

| | |
|---|------------------------|
| A | 15 Puzzle |
| B | SVM |
| C | Easy Nim |
| D | Hard Nim |
| E | Power tower |
| F | lzd is a scout |
| G | Environmental friendly |
| H | Observe |
| I | Broken routers |
| J | League schedule |
| K | Simple Math |
| L | Carrot Fantasy |
| M | Debug it! |

请勿在比赛开始前翻阅试题！

Do not open before the contest has started.

2021 年 11 月 7 日

Problem A. 15 Puzzle

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

The 15 puzzle (also called Gem Puzzle, Boss Puzzle, Game of Fifteen, Mystic Square and many others) is a sliding puzzle that consists of a frame of numbered square tiles in random order with one tile missing. The puzzle also exists in other sizes, particularly the smaller 8 puzzle. If the size is 3×3 tiles, the puzzle is called the 8 puzzle or 9 puzzle, and if 4×4 tiles, the puzzle is called the 15 puzzle or 16 puzzle named, respectively, for the number of tiles and the number of spaces. The goal of the puzzle is to place the tiles in order by making sliding moves that use the empty space.

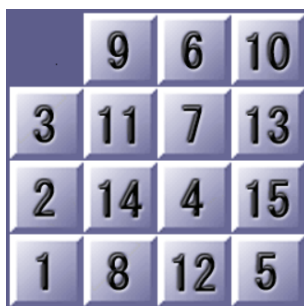
Here is some pictures for better understanding.



This picture is the final state of the game.



This picture is a condition that you can not solve it.



This picture is a shuffle of game.

So your target is to solve a shuffled 15 puzzle. You can use U, D, L, R to move tiles.

U means move the tile under the empty space up to the empty space.

D means move the tile above the empty space down to the empty space.

L means move the tile on the right side of the empty space left to the empty space.

R means move the tile on the left side of the empty space right to the empty space.

Now you can use these letters to represent your solution. But it may not have a solution, just print No.

Input

4 lines, each line has 4 numbers, represent the 15 puzzle game.

The numbers between 1 to 15 means the tiles with number and 0 means the empty space.

Output

Print No if there is no solution.

Print Yes in the first line if there are solutions. Then output the length of your solution and your solution in the second and the third line. Meanwhile, the length of your solution should not exceed 2000.

Examples

| standard input | standard output |
|--|-----------------|
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 0 15 | Yes 1 L |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 15 14 0 | No |

Problem B. SVM

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **128 megabytes**

SVM loss is a useful loss function in machine learning. Assume there are n training examples and m classes in a image classification problem. Each training example contains a training image and a label. The label is an integer in range $[1, m]$.

Our machine learning model takes n training images as input and outputs n score vectors of m dimensions. SVM loss of i -th score vector L_i is defined as the following formula.

$$L_i = \sum_{j=1, j \neq \text{label}_i}^m \max(0, S_{i,j} - S_{i, \text{label}_i} + \text{Hinge}), \text{ where } \text{Hinge} \text{ is a constant.}$$

We want to accelerate the model training process by calculating the SVM loss function quickly. You need to answer Q queries asking the sum of SVM loss from l -th score vector to r -th score vector when Hinge equals h . Formulaly, you need to answer $\sum_{i=l}^r L_i(\text{Hinge} = h)$.

Input

The first line contains two integers n, m ($1 \leq n \times m \leq 10^6$).

The second line contains n integers, the i -th integer represents label_i ($1 \leq \text{label}_i \leq m$).

The next n lines each contains m integers representing n score vectors of m dimensions. The j -th integer of i -th line represents $S_{i,j}$ ($-10^9 \leq S_{i,j} \leq 10^9$) which means the j -th dimension of i -th score vector.

The next line contains an integer Q ($1 \leq Q \leq 2 * 10^5$) representing the number of queries.

The next Q lines each contains three integers l, r, h ($1 \leq l, r \leq n, 0 \leq h \leq 10^9$) having the same meaning in description.

Output

There are Q lines in the output, i -th line contains an integer representing the answer of i -th query.

Example

| standard input | standard output |
|----------------|-----------------|
| 5 3 | 9 |
| 3 2 3 3 1 | 6 |
| 6 5 4 | 14 |
| 8 6 3 | 11 |
| 4 2 1 | 23 |
| 3 3 3 | 12 |
| 5 2 2 | 30 |
| 8 | 16 |
| 1 3 0 | |
| 2 5 0 | |
| 1 3 1 | |
| 2 5 1 | |
| 1 4 2 | |
| 3 5 2 | |
| 1 4 3 | |
| 3 5 3 | |

Problem C. Easy Nim

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

lyw and his friend zyw are playing an boring game called *Easy Nim*. In this game, there is a pile of n stones on the table. Two players take turns choosing some stones and removing them from the table. The game always starts from lyw and ends when there is no stone left on the table. The player who removes the last stone wins this game.

But they all felt that the rules of the game were a bit boring, so they added some additional constraints. First, the player can only remove at least 1 stone and at most m stones from the table each round. The second, zyw will have k extra magic cards as a blessing from God. Each card has a number written on it. When zyw uses this magic card, he can change the upper limit of the stone that can be removed in the rest round of the game to it, but he can't remove any stones during the round when he uses the magic card. Note that each card could be used only once.

Assume that both of them are witty enough to win the *Easy Nim* game, please tell Jury who can win this game?

Input

The input contain two lines.

The first line contains three integers n , m , k ($1 \leq n \leq 10^9$, $1 \leq m \leq 10^6$, $1 \leq k \leq 1000$) representing when the game start there are n stones on the table and most m stones can be removed in each round and zyw will have k magic cards.

The second line contains k intergers, the i -th interger c_i ($1 \leq c_i \leq 10^6$) representing the number written on the i -th card.

Output

Print lyw if lyw will win, otherwise zyw.

Examples

| standard input | standard output |
|----------------|-----------------|
| 10 5 1 2 | zyw |
| 10 5 1 3 | lyw |

Problem D. Hard Nim

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

lyw and his friend zyw are playing an exciting game called *Hard Nim*. In this game, there is a pile of N stones on the desk. Two players take turns selecting some stones and remove them from the desk. lyw will always go first. The game ends when there is no stone left on the desk. The player who removes the last stone wins the game.

There are also two additional constraints in the game. First, in each turn, the player must remove at least 1 and at most M stones from the desk. Second, the player will lose the game if the number of stones remaining (represents in binary) has odd number of ones after his removing.

Assume both of them playing optimally, can you judge who will win the *Hard Nim* game?

Input

Single line contains two intergers N , M ($1 \leq N \leq 10^{12}$, $1 \leq M \leq 10^6$). It guarantees that N will have an even number of ones in its binary representation.

Output

Output lyw if lyw will win, otherwise ouput zyw

Examples

| standard input | standard output |
|----------------|-----------------|
| 3 1 | zyw |
| 3 2 | zyw |
| 3 3 | lyw |

Problem E. Power tower

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

Consider an array $\{a_n\}$, where $a_1 = 2$ and $a_n = 2^{a_{n-1}}$, for all $n \geq 2$. As you can imagine, while n is huge enough, a_n looks like a tower, and we call this tower “Power tower”.

But as we all know, computer has difficulty in dealing with big integers, so he defines another array $b_n = a_n \bmod p$, where p is a given positive integer and \bmod represents modular operation.

It is easy to prove that $\{b_n\}$ is a convergent array(In other words, $\lim_{n \rightarrow +\infty} b_n$ exists). Your task is to calculate the $\lim_{n \rightarrow +\infty} b_n$.

Input

The input has many lines.

The first line contains a positive integer T ($1 \leq T \leq 10^3$), means the number of limit you need to calculate.

The next T lines, Each line contains a positive integer p ($1 \leq p \leq 10^7$).

Output

The output contains T lines, for each line, you need to output a positive integer p , means the value of $\lim_{n \rightarrow +\infty} b_n$.

Example

| standard input | standard output |
|----------------|-----------------|
| 3 | 0 |
| 2 | 1 |
| 3 | 4 |
| 6 | |

Problem F. lzd is a scout

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

Recently, lzd love play *Battlefield 1*, he especially likes to play scouts. In the game he is a shooter and can snipe any opponent he sees. But there is an interval feature called **damage drop** in the game, which means that if the distance between enemy and him is in this interval, lzd only needs to shoot it once to kill it, otherwise he needs an extra shot to make up for the damage. For example, now lzd has a Gewehr 98, and its damage range is $[80, 125]$, so all enemies whose distance is within this range can be killed with just one shot.

Today, lzd is trapped on the top of the mountain by n enemies. But he only has a homemade rifle with m bullets while the rifle's damage drop is $[l, r]$. You have known all the distance between he and enemies. Please tell him whether he can finish all the kill or not.

Input

The input contain two lines.

The first line contains four intergers n, m, l, r ($1 \leq n \leq 10^6, 1 \leq m \leq 10^9, 1 \leq l \leq r \leq 10^9$), representing there are n enemies and m bullets with the damage dorp $[l, r]$ of lzd's rifle.

The second line contains m intergers, the i -th interger d_i ($1 \leq d_i \leq 10^9$) representing the distance between the i -th enemy and him.

Output

Print a line with **lzdak!** if lzd can finish the all kill, otherwise print a integer representing the number of enemies he can kill

Example

| standard input | standard output |
|------------------|-----------------|
| 2 3 5 10 3 10 | lzdak! |

Problem G. Environmental friendly

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 256 megabytes

In response to the national call for energy conservation and emission reduction, Master Yi, the director of the Transportation Bureau in Pigeon province, decided to demolish some unnecessary roads to reduce waste of resources. There are n cities numbered from 1 to n in his province. Also there are m roads connecting the cities. One can go from city u_i to v_i (and vise versa) using i -th road, the length of this road is w_i .

Master Yi doesn't want to waste the money of the province, so he is going to close some of the roads in response to the country's call to save resources. Please tell Master Yi the maximum number of the roads which can be closed such that the distance between any two cities in Pigeon province does not change.

Input

The first line contains an single integer T ($1 \leq T \leq 10$), which is the number of the text cases.

Then T cases follow, each case starts with two integers n, m ($2 \leq n \leq 200, 1 \leq m \leq 10^5$).

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$). Note that there may be more than one road between two cities.

Output

For each case, output a single integer representing the maximum number of the roads which can be closed.

Example

| standard input | standard output |
|----------------|-----------------|
| 2 | 2 |
| 2 3 | 1 |
| 1 2 1 | |
| 1 2 1 | |
| 1 2 2 | |
| 4 4 | |
| 1 2 1 | |
| 1 3 3 | |
| 2 3 2 | |
| 3 4 4 | |

Problem H. Observe

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

In order to collect geographical information on Mount Wolwz, a team of geographical observers needed to establish four observation points on the mountain. In order to obtain accurate data, the four observation points need to be exactly the 4 **vertices of a square** when viewed from the air. The observation algorithm used by the team required at least three of the observation points to be at the same height. And for the last observation point, which is at a different height to the other three, it needs to be at a **higher** position than the other three.

The top view of Mount Wolwz can be seen as a square with $n * n$ small square platforms, where each piece has an integer height. The group discussed that the higher the three observation points of the same height, the better the result. So they wanted to find out what the highest of the three observation points at the same height was.

Input

The input contains a total of $n + 1$ lines

The first line contains an integer n ($2 \leq n \leq 300$), representing the length of Mount Wolwz's side.

Next n lines, each contains n integers. The j -th integer of the i -th line $h_{i,j}$ ($1 \leq h_{i,j} \leq 10^9$), representing the height of each of the small platforms of Mount Wolwz.

Output

One line contains an interger, representing the best height. If suitable observation points cannot be established in Mount Wolwz, output -1.

Examples

| standard input | standard output |
|------------------------------|-----------------|
| 3 1 1 3 1 2 2 3 2 3 | 2 |
| 3 1 2 1 2 2 2 1 2 1 | -1 |

Note

In the first example, $[(1, 1), (1, 2), (2, 1), (2, 2)]$ and $[(2, 2), (3, 2), (2, 3), (3, 3)]$ are all suitable observation point combinations.

Problem I. Broken routers

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

There are n routers in a network each of which has some packets stored in it and a routing table. Routers look up ip address of the packet in its routing table and send the packet to corresponding router neighbor. If the packet's ip address is not in the routing table then the packet will be send to its default router neighbor.

Unfortunately, all of the n routers have broken down and their routing table become empty. So they can only send packets to their default router neighbor.

There are a_i packets stored in i -th router at time slot 0. Each of the n routers send all the packets stored in it to default router neighbor and remove these packets from itself in every time slot. Routers also store the packets it recieved in each time slot.

As a administrator of the network, you want to know how many packets are there in router x at time t .

Input

The first line contains an integer n ($1 \leq n \leq 10^5$), meaning the number of routers.

The second line contains n intergers a_1, \dots, a_n ($1 \leq a_i \leq 10^9$).

The third line contains n intergers neb_1, \dots, neb_n ($1 \leq neb_i \leq n$) where neb_i represents the default router neighbor of i -th router.

The next line contains an integer Q ($1 \leq Q \leq 10^5$), meaning the number of queries.

The next Q lines each contains two integer x, t ($1 \leq x \leq n, 0 \leq t \leq 10^9$), having the same meaning in description.

Output

There are Q lines in the output, the i -th line contains one interger representing the answer of i -th query.

Example

| standard input | standard output |
|----------------|-----------------|
| 7 | 7 |
| 1 2 3 4 5 6 7 | 5 |
| 2 3 1 1 4 2 2 | 5 |
| 5 | 7 |
| 1 2 | 0 |
| 4 1 | |
| 5 0 | |
| 3 3 | |
| 6 1 | |

Problem J. League schedule

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 256 megabytes

People in Kingdom Z loves football. Zyw the king of Kingdom Z wants to establish a football league in his country called *Z-league*.

There are n teams in the *Z-league* and each of them has an **unique** team name which consists of three uppercase letters. For example, PSG, MCI, MUN are legal team names but FcB, S04, OL are not.

Now, giving you n (n is even) team names. You need to arrange a schedule for the first season of *Z-league*. Your schedule should meet the following conditions.

1. Each pair of teams, for example team AAA and team BBB should have exactly two matches in the season. One is “AAA vs BBB” in which team AAA plays as the home team, the other is “BBB vs AAA” in which team BBB plays as the home team.
2. Each team has at most one match per week.
3. Number of weeks the season takes should be as little as possible.

Input

There are $n + 1$ lines in the input.

The first line contains an even integer n ($1 \leq n \leq 500$) represents the number of teams in *Z-league*.

The next n lines each contains a legal team name.

Output

There are $n \times (n - 1)$ lines in the output.

Each line should be the format of *home_team_name vs visiting_team_name on week w*, where maximum value of w should be as little as possible.

Examples

| standard input | standard output |
|-------------------------------|--|
| 2 MCI MUN | MCI vs MUN on week 1 MUN vs MCI on week 2 |
| 4 LIV TOT ARS CHE | TOT vs ARS on week 1 ARS vs TOT on week 2 LIV vs CHE on week 1 CHE vs LIV on week 2 ARS vs LIV on week 4 LIV vs ARS on week 3 TOT vs CHE on week 3 CHE vs TOT on week 4 LIV vs TOT on week 5 TOT vs LIV on week 6 ARS vs CHE on week 5 CHE vs ARS on week 6 |

Problem K. Simple Math

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

Recently, zyw is very interested in number theory. He find a very hard question:

Give a integer n , try to find 3 integers x, y, z satisfied $1 \leq x \leq y \leq z$ and $x + y + z = n$ makes the value of $\gcd(x, y) + \gcd(y, z) + \gcd(z, x)$ as big as possible.

Zyw is a math genius and solved it in just $10^{-114514}$ second and it costs the judger 1000ms to give answer. Now he wants to know whether there is a faster way to solve it.

Input

First line gives a integer T , represents the number of problems you should solve.

Next T ($1 \leq T \leq 10^4$) lines, each line gives a integer n_i ($3 \leq n_i \leq 10^9$), represent the n in the legend for each problem.

Output

T lines, each line has three integers, represents x, y, z .

If there are more than 1 solutions, output any of it.

Example

| standard input | standard output |
|----------------|-----------------|
| 1 | 1 1 1 |
| 3 | |

Problem L. Carrot Fantasy

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

Master Yi is a farmer. He has a large farm with many radishes planted in it. Recently, he lost some due to the stealing of wild rabbits. He thus decided to place some fences to protect all his carrots.

The farm is a rectangle consisting of $R \times C$ cells. Each cell is either empty, contains a carrot, a rabbit or a fence. Rabbits can roam freely around the farm, by repeatedly moving to the left, right, up or down to a neighboring cell. When a rabbit enters a cell with a carrot, it consumes it. However, no rabbit can enter a cell with a fence.

Initially there are no fences. Place fences onto the farm in such a way that no rabbit can reach any carrot, or determine that it is impossible. Note that since you have many fences, you do **not** need to minimize their number.

Input

First line contains two integers R ($1 \leq R \leq 500$) and C ($1 \leq C \leq 500$), denoting the number of rows and the numbers of columns respectively.

Each of the following R lines is a string consisting of exactly C characters, representing one row of the pasture. Here, C means a carrot, R is a rabbit and . an empty cell.

Output

If it is impossible to protect all carrots, output a single line with the word No.

Otherwise, output a line with the word Yes. Then print R lines, representing the farm after placing fences. Again, C means a carrot, R is a rabbit, # is a fence and . an empty cell. You are not allowed to move, remove or add a carrot or a rabbit.

Example

| standard input | standard output |
|--|--|
| 5 5R C..R. CC..RR.CC | Yes ..#.R C.#R. CC#.R ##### .R#CC |

Problem M. Debug it!

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

Miamiao writes a program and submits it to the judger. The judger tells her the result of each test case by a string containing P or F. P means to pass the test and F means to fail the test. Now Huihui wants to help her debug the program. He can remove the first or the last letter in 1 minute or remove any other letter in 2 minutes. After removing a letter from the middle the two remaining parts are joined.

Now, Huihui wants to know, how fast can he finish debugging the problem, which means he can obtain a string without any letter F?

Input

The first line of input contains an integer n ($1 \leq n \leq 10^6$), which means the length of the string.

The second line of input contains a string s ($|s| = n$, $s_i \in \{F, P\}$), which means Miamiao's result.

Output

Output a number, which means the minimum time needed to debug the problem by Huihui.

Example

| standard input | standard output |
|----------------|-----------------|
| 4 FPPF | 2 |