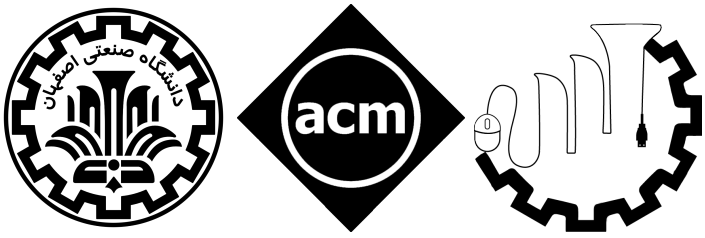
ACM Local Collegiate Programming Contest

Isfahan University of Technology

March 2018

Organizers:



Sponsors:

A. Young Physicist

A guy named Alireza attends the final semester of a university. One day Alireza decided to compete in a programming contest. And, as the boy loves programming contests very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got very angry at Alireza and decided to teach him a lesson. He gave the lazy student a seemingly easy task: You are given an idle body in space and the forces that affect it. The body can be considered as a material point with coordinates (0; 0; 0). Alireza had only to answer whether it is in equilibrium.

"Piece of cake" - thought Alireza, we need only to check if the sum of all vectors is equal to 0. So, Alireza began to solve the problem. But later it turned out that there can be lots and lots of these forces, and Alireza can not cope without your help. Help him. Write a program that determines whether a body is idle or is moving by the given vectors of forces.

Input

There are several test cases.

The first line contains a positive integer T, the number of test cases. The first line of each test, contains a positive integer n ($1 \leq n \leq 100$), then follow n lines containing three integers each: the x_i coordinate, the y_i coordinate and the z_i coordinate of the force vector, applied to the body ($-100 \leq x_i, y_i, z_i \leq 100$).

Output

For each test case print a line containing the word "`YES`" if the body is in equilibrium, or the word "`NO`" if it is not.

Sample Input

```
2
3
4 1 7
-2 4 -1
1 -5 -3
3
3 -1 7
-5 2 -4
2 -1 -3
```

Sample Output

```
NO
YES
```

B. Restoring Password

Mehran always used to trust his favorite Antivirus. That is why he didn't hesitate to download the link one of his groupmates sent him via Telegram. The link was said to contain "some real funny stuff about ACM contestants". The antivirus had no objections and Mehran run the flash application he had downloaded. Immediately his Telegram application said: "invalid login/password".

Mehran entered the information from his additional account and looked at the info of his main one. His first name and family name changed to "MEH" and "Infected" correspondingly, and the "Bio" field contained a strange-looking binary code 80 characters in length, consisting of zeroes and ones. "I've been hacked" - thought Mehran and run the Internet Exploiter browser to quickly type his favorite search engine's address. Soon he learned that it really was a virus that changed Telegram users' passwords. Fortunately, he soon found out that the binary code was actually the encrypted password where each group of 10 characters stood for one decimal digit. Accordingly, the original password consisted of 8 decimal digits.

Help Mehran restore his Telegram account by the encrypted password and encryption specification.

Input

The Input data contains multiple test cases. The first line contains a positive integer T, the number of test cases.

Each test case contains 11 lines. The first line represents the binary code 80 characters in length. That is the code written in Mehran's Telegram account's info. Next 10 lines contain pairwise distinct binary codes 10 characters in length, corresponding to numbers 0, 1, ..., 9.

Output

Print one line per test case containing 8 characters - The password to Mehran's Telegram account. It is guaranteed that the solution exists.

Sample Input

```
1
01001100100101100000010110001001011001000101100110010110100001011010100101101100
0100110000
0100110010
0101100000
0101100010
0101100100
0101100110
0101101000
0101101010
0101101100
0101101110
```

Sample Output

```
12345678
```

C. Lucky Mask

Mahya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits 4 and 7. For example, numbers 47, 744, 4 are lucky and 5, 17, 467 are not.

Mahya calls a mask of a positive integer n the number that is obtained after successive writing of all lucky digits of number n from the left to the right. For example, the mask of number 72174994 is number 7744, the mask of 7 is 7, the mask of 9999047 is 47. Obviously, mask of any number is always a lucky number.

Mahya has two numbers - an arbitrary integer a and a lucky number b. Help her find the minimum number c ($c > a$) such that the mask of number c equals b.

Input

The Input data contains multiple test cases. The first line contains a positive integer T, the number of test cases. Each test case contains two integers a and b ($1 \leq a, b \leq 10^5$). It is guaranteed that number b is lucky.

Output

For each test case print a line containing a single number - the number c that is sought by Mahya.

Sample Input

```
2
1 7
100 47
```

Sample Output

```
7
147
```

D. cd and pwd commands

Izak is writing an operating system shell, and it should have commands for working with directories. To begin with, he decided to go with just two commands: `cd` (change the current directory) and `pwd` (display the current directory).

Directories in Izak's operating system form a traditional hierarchical tree structure. There is a single root directory, denoted by the slash character "`/`". Every other directory has a name - a non-empty string consisting of lowercase Latin letters. Each directory (except for the root) has a parent directory - the one that contains the given directory. It is denoted as "`..`".

The command `cd` takes a single parameter, which is a path in the file system. The command changes the current directory to the directory specified by the path. The path consists of the names of directories separated by slashes. The name of the directory can be "`..`", which means a step up to the parent directory. "`..`" can be used in any place of the path, maybe several times. If the path begins with a slash, it is considered to be an absolute path, that is, the directory changes to the specified one, starting from the root. If the parameter begins with a directory name (or "`..`"), it is considered to be a relative path, that is, the directory changes to the specified directory, starting from the current one.

The command `pwd` should display the absolute path to the current directory. This path must not contain "`..`".

Initially, the current directory is the root. All directories mentioned explicitly or passed indirectly within any command `cd` are considered to exist. It is guaranteed that there is no attempt of transition to the parent directory of the root directory.

Input

The Input data contains multiple test cases. The first line contains a positive integer T, the number of test cases.

The first line of each test case contains the single integer n ($1 \le n \le 50$) - the number of commands. Then follow n lines, each contains one command. Each of these lines contains either command `pwd`, or command `cd`, followed by a space-separated non-empty parameter. The command parameter `cd` only contains lower case Latin letters, slashes and dots, two slashes cannot go consecutively, dots occur only as the name of a parent pseudo-directory. The command parameter `cd` does not end with a slash, except when it is the only symbol that points to the root directory. The command parameter has a length from 1 to 200 characters, inclusive.

Directories in the file system can have the same names.

Output

Print the desired output for each test case. Print a black line after each test case.

For each command `pwd` you should print the full absolute path of the given directory, ending with a slash. It should start with a slash and contain the list of slash-separated directories in the order of being nested from the root to the current folder. It should contain no dots.

Sample Input

```
2
7
pwd
cd /home/izak
pwd
cd ..
pwd
cd izak/../mamad
pwd
4
cd /a/b
pwd
cd ../a/b
pwd
```

Sample Output

```
/
/home/izak/
/home/
/home/mamad/

/a/b/
/a/a/b/
```

E. Movie Critics

A film festival is coming up in the city N. The festival will last for exactly n days and each day will have a premiere of exactly one film. Each film has a genre - an integer from 1 to k.

On the i-th day the festival will show a movie of genre a_i. We know that a movie of each of k genres occurs in the festival programme at least once. In other words, each integer from 1 to k occurs in the sequence $a_1, a_2, ..., a_n$ at least once.

Amirhosein is a movie critic. He wants to watch some movies of the festival and then describe his impressions on his site.

As any creative person, Amirhosein is very susceptive. After he watched the movie of a certain genre, Amirhosein forms the mood he preserves until he watches the next movie. If the genre of the next movie is the same, it does not change Amirhosein's mood. If the genres are different, Amirhosein's mood changes according to the new genre and Amirhosein has a stress.

Amirhosein can't watch all n movies, so he decided to exclude from his to-watch list movies of one of the genres. In other words, Amirhosein is going to choose exactly one of the k genres and will skip all the movies of this genre. He is sure to visit other movies.

Amirhosein wants to choose such genre x ($1 \leq x \leq k$), that the total number of after-movie stresses (after all movies of genre x are excluded) were minimum.

Input

The Input data contains multiple test cases. The first line contains a positive integer T, the number of test cases.

The first line of each test case contains two integers n and k ($2 \leq k \leq n \leq 10^5$), where n is the number of movies and k is the number of genres.

The second line of each test case contains a sequence of n positive integers $a_1, a_2, ..., a_n$ ($1 \leq a_i \leq k$), where a_i is the genre of the i-th movie. It is guaranteed that each number from 1 to k occurs at least once in this sequence.

Output

For each test case print a single number - the number of the genre (from 1 to k) of the excluded films. If there are multiple answers, print the genre with the minimum number.

Sample Input

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

Sample Output

```
3
```

F. Zahra, Maryam and Chocolate

Zahra and Maryam like games. And now they are ready to start a new game. They have placed n chocolate bars in a line (The chocolate bars were for Tahere actually!). Zahra starts to eat chocolate bars one by one from left to right, and Maryam - from right to left. For each chocolate bar the time, needed for the player to consume it, is known (Zahra and Maryam eat them with equal speed). When the player consumes a chocolate bar, he immediately starts with another. It is not allowed to eat two chocolate bars at the same time, to leave the bar unfinished and to make pauses. If both players start to eat the same bar simultaneously, Maryam leaves it to Zahra because Zahra may get angry (and scary)!

How many bars each of the players will consume?

Input

The Input data contains multiple test cases. The first line contains a positive integer T, the number of test cases.

The first line of each test case contains one integer n ($1 \leq n \leq 10^5$) - the amount of bars on the table. The second line contains a sequence $t_1, t_2, ..., t_n$ ($1 \leq t_i \leq 1000$), where t_i is the time (in seconds) needed to consume the i-th bar (in the order from left to right).

Output

For each test case print two numbers a and b, where a is the amount of bars consumed by Zahra, and b is the amount of bars consumed by Maryam.

Sample Input

```
1
5
2 9 8 2 7
```

Sample Output

```
2 3
```

G. Multiplication Game

Alice and Bob are in their class doing drills on multiplication and division. They quickly get bored and instead decide to play a game they invented.

The game starts with a target integer $N \geq 2$, and an integer $M = 1$. Alice and Bob take alternate turns. At each turn, the player chooses a prime divisor p of N, and multiply M by p. If the player's move makes the value of M equal to the target N, the player wins. If $M > N$, the game is a tie.

Assuming that both players play optimally, who (if any) is going to win?

Input

The first line of input contains T ($1 \leq T \leq 10\,000$), the number of cases to follow. Each of the next T lines describe a case. Each case is specified by N ($2 \leq N \leq 2^{31}-1$) followed by the name of the player making the first turn. The name is either `Alice` or `Bob`.

Output

For each case, print the name of the winner (`Alice` or `Bob`) assuming optimal play, or `tie` if there is no winner.

Sample Input

```
10
10 Alice
20 Bob
30 Alice
40 Bob
50 Alice
60 Bob
70 Alice
80 Bob
90 Alice
100 Bob
```

Sample Output

```
Bob
Bob
tie
tie
Alice
tie
tie
tie
tie
Alice
```

H. Winning ICPC

There are N teams (numbered from 1 to N) and M problems (numbered from 1 to M) in this year's ICPC. The j-th problem has T_j test cases. Surprisingly, every team submitted exactly one solution to every problem. The i-th team managed to solve $S_{i,j}$ test cases on the j-th problem.

A team solved a problem only if the team managed to solve all test cases on that problem. The winning team is the team with the most number of problems solved. If there are more than one team with the most number of problems solved, then the winning team is the team with the smallest index among those teams.

Determine the index of the winning team.

Input

The first line contains two integers: N, M $(1 \leq N, M \leq 100)$ in a line denoting the number of teams and the number of problems. The second line contains M integers: $T_1, T_2, ...T_M$ $(0 \leq T_i \leq 100)$ in a line denoting the number of test cases. The next N following lines, each contains M integers; the j-th integer on the i-th line is $S_{i,j}$ $(0 \leq S_{i,j} \leq T_j)$ denoting the number of solved test cases by the i-th team for the j-th problem.

Output

The output contains the index of the winning team, in a line.

Sample Input

```
3 2
10 20
0 19
10 0
9 19
```

Sample Output

```
2
```

I. Cloning Toys

Zeinab likes her plush toy a lot.

Recently, she found a machine that can clone plush toys. Zeinab knows that if she applies the machine to an original toy, she additionally gets one more original toy and one copy, and if she applies the machine to a copied toy, she gets two additional copies.

Initially, Zeinab has only one original toy. She wants to know if it is possible to use machine to get exactly x copied toys and y original toys? She can't throw toys away, and she can't apply the machine to a copy if she doesn't currently have any copies.

Input

The only line contains two integers x and y ($0 \leq x, y \leq 10^9$) - the number of copies and the number of original toys Zeinab wants to get (including the initial one).

Output

Print "`Yes`", if the desired configuration is possible, and "`No`" otherwise.

Sample Input 1

```
6 3
```

Sample Output 1

```
Yes
```

Sample Input 2

```
4 2
```

Sample Output 2

```
No
```

Sample Input 3

```
1000 1001
```

Sample Output 3

```
Yes
```

J. Permute Digits

You are given two positive integer numbers a and b. Permute (change order) of the digits of a to construct maximal number not exceeding b. No number in input and/or output can start with the digit 0.

It is allowed to leave a as it is.

Input

The first line contains integer a ($1 \leq a \leq 10^{18}$). The second line contains integer b ($1 \leq b \leq 10^{18}$). Numbers don't have leading zeroes. It is guaranteed that answer exists.

Output

Print the maximum possible number that is a permutation of digits of a and is not greater than b. The answer can't have any leading zeroes.

The number in the output should have exactly the same length as number a. It should be a permutation of digits of a.

Sample Input 1

```
123
222
```

Sample Output 1

```
213
```

Sample Input 2

```
3921
10000
```

Sample Output 2

```
9321
```

Sample Input 3

```
4940
5000
```

Sample Output 3

```
4940
```

Sample Input 4

```
9999999999999999
1000000000000000000
```

Sample Output 4

```
9999999999999999
```

K. Planet Hunting

In a fictitious solar system consisting of star S, a planet P, and its moon M, planet P orbits in a perfect circle around star S with a revolution period of exactly T Earth days, and moon M orbits in a perfect circle around planet P with an unknown revolution period. Given the position of moon M relative to star S at three different time points, your goal is to compute the distance of planet P from star S.

To do this, consider a two-dimensional Cartesian coordinate system with its origin centered at star S. You may assume that P's counterclockwise orbit around S and M's counterclockwise orbit around P both lie completely within the x - y coordinate plane. Let (x_1, y_1) denote the position of the moon M on the first observation, let (x_2, y_2) denote its position k_1 Earth days later, and let (x_3, y_3) denote its position k_2 Earth days after the second observation.

Input

The input test file will contain multiple test cases. Each test case consists of two lines. The first line contains the integers T, k_1, and k_2, where $1 \leq T, k_1, k_2 \leq 1000$. The second line contains six floating-point values x_1, y_1, x_2, y_2, x_3, and y_3. Input points have been selected so as to guarantee a unique solution; the final distance from planet P to star S will always be within 0.1 of the nearest integer. The end-of-file is denoted with a single line containing "`0 0 0`".

Output

For each input case, the program should print the distance from planet P to star S, rounded to the nearest integer.

Sample Input

```
360 90 90
5.0 1.0 0.0 6.0 -5.0 1.0
0 0 0
```

Sample Output

```
5
```