Problem A. Patterns

Time limit: 1 second
Memory limit: 512 megabytes

Inspired by the Ulam spiral, which unravels strange patterns of prime numbers' distribution, Petya decided to come up with his own analog.

Petya writes down integers from 1 to n^2 into square table of size $n \times n$ starting from the upper left corner. Numbers from 1 to n go into the first row, numbers from $n+1$ to $2n$ — into the second row, and so on.

Then he colors cells which contain a number with no more than k different divisors. After that, Petya studies the resulting picture in hope of finding some patterns. For example, if $n = 7$, $k = 3$, Petya will get the following picture:

1	**2**	**3**	**4**	**5**	**6**	**7**
8	**9**	**10**	**11**	**12**	**13**	**14**
15	**16**	**17**	**18**	**19**	**20**	**21**
22	**23**	**24**	**25**	**26**	**27**	**28**
29	**30**	**31**	**32**	**33**	**34**	**35**
36	**37**	**38**	**39**	**40**	**41**	**42**
43	**44**	**45**	**46**	**47**	**48**	**49**

Help Petya, print the picture which he will get after coloring, represent colored cells with asterisks «*», and non-colored cells with dots «.».

Input

Input contains two integers n and k ($1 \le n \le 40$, $1 \le k \le n^2$).

Output

Output n lines, each one must contain n characters. If the j-th cell of the i-th row of Petya's table is colored, the j-th character of the i-th line must be equal to «*», it must be equal to «.» otherwise.

Example

standard input	standard output
7 3	*****.* .*.*.*. ..*.*.. .*.*... *.*.... .*...*. *...*.*

Problem B. Interesting Excursion

Time limit:	4 seconds
Memory limit:	512 megabytes

Flatland has n cities, connected by m one-directional roads.

Tourist company plans to develop a scenic cyclic tour along the roads of Flatland. This tour must start and finish at the same city, visiting some intermediate cities and traveling along some of the roads in their direction. The tour can visit some cities multiple times, but it may not use the same road more than once.

Each road is characterized by the type of its landscape, which is the number from 1 to m. To make the tour really magnificent, every two adjacent roads in the tour must have different landscape types. This also should be true for the first and the last road in the tour, so that you start to travel from any city of the tour.

Help the company to find the tour satisfying these conditions, or report that no such tour exists.

Input

Input contains multiple test cases. The first line contains an integer T — the number of test cases ($1 \le T \le 10^5$).

The first line of each test case description contains two integers n and m ($2 \le n, m \le 2 \cdot 10^5$) — the number of cities and the number of roads. Each of the next m lines contains three integers u_i v_i c_i, meaning that the i-th road starts at city u_i, ends at city v_i and has landscape type c_i ($1 \le u_i, v_i \le n$, $1 \le c_i \le m$, $u_i \ne v_i$).

Sum of all n in all test cases does not exceed $2 \cdot 10^5$. Sum of all m in all test cases does not exceed $2 \cdot 10^5$.

Output

Output the answer of each test case.

If the desired tour does not exist, output the only number «-1». Otherwise, print number k, such that $2 \le k \le m$ — the length of the tour. The next line must contain k numbers $e_1, e_2, \ldots, e_k$ — the numbers of roads in the tour. All numbers e_i must be different. If there are multiple possible tours, output any of them.

Example

standard input	standard output
3 5 8 1 4 1 2 4 1 4 5 2 3 2 2 5 3 1 3 2 3 5 2 2 2 1 3 4 5 1 2 2 2 3 1 2 4 4 4 1 2 3 1 2 2 3 1 2 1 1 2 2 2 1 1	4 3 5 6 2 -1 2 2 3

Problem C. Jump and Turn

Time limit: 1 second
Memory limit: 512 megabytes

Vanya is now sleeping and in the dream he's standing on a grid of size $n \times m$. He wants to visit every cell of the grid exactly once. Vanya can initially place himself to the center of any cell and then he can jump from the cell he's standing on now to any other cell. Every time Vanya jumps exactly to the center of cell. Vanya can jump from any cell to any other cell but there's a problem. Every time after a jump he must turn strictly to the left. Specifically any three cells Vanya visit successively must satisfy following: if one stands in the center of the first cell and looks to the center of the second cell, then center of the third cell must belong to the left half-plane, excluding the line connecting the centers of first and second cells. In particular, the centers of the three successively visited cells must not belong to the same straight line.

Columns of the grid are numbered from left to right from 1 to n. Rows of the grid are numbered from bottom to top from 1 to m.

Help Vanya to find the way to visit every cell exactly once, or report that it is impossible.

Input

Input contains two integers n and m — number of columns and number of rows ($1 \le n, m \le 100$).

Output

The first line must contain «`Yes`», if it is possible to visit every cell as described, output «`No`» otherwise.

If the solution exists, output $n \cdot m$ more lines, the i-th of them must contain two integers x_i and y_i — the column and the row of the cell that Vanya should visit on the i-th step ($1 \le x_i \le n$, $1 \le y_i \le m$).

Examples

standard input	standard output
2 2	Yes 1 1 2 2 1 2 2 1
4 2	Yes 4 1 2 2 1 1 4 2 3 2 2 1 3 1 1 2

Note

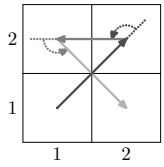

Рис. 1: The first sample case.

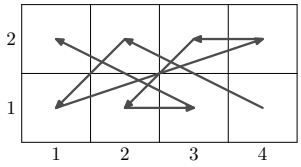

Рис. 2: The second sample case.

Problem D. Counting in the Order

Time limit: 1 second
Memory limit: 512 megabytes

The whole year Vasya didn't attend the university, so he didn't pass his exams and was expelled. That's how he ended up in the army. And the most popular exercise in the army is standing in the row.

There are n soldiers in Vasya's troop. Soldiers are standing in a row, each of them is looking to his left or to his right. The height of the i-th soldier in the row is h_i. Vasya thinks that the soldier with number i sees the soldier with number j if the following conditions are true:

- soldier i looks in the direction of the soldier j;
- all soldiers standing between them are not taller than the soldier j.

For example, if there are 4 soldiers in the row with heights $h_1 = 178$, $h_2 = 180$, $h_3 = 170$, $h_4 = 190$, and all soldiers are looking to the left, then the 2-nd soldier will only see the 1-st one, the 3-rd soldier will only see the 2-nd one (because there is a taller soldier between him and the first soldier), the 4-th one sees the 2-nd and the 3-rd soldier.

There is nothing to do while standing in the row, so Vasya wants to calculate how many other soldiers each of the soldiers in the row sees.

Input

The first line of input contains an integer n — the number of soldiers in the row ($1 \le n \le 10^5$).

The second line contains n integers $h_1, h_2, \ldots, h_n$ — the heights of soldiers in the row ($1 \le h_i \le 10^9$).

The third line contains n characters representing the directions in which the soldiers look: the i-th symbol is equal to «L» if the i-th soldier looks to the left, and can potentially see only soldiers with $1, 2, \ldots, i-1$, or to «R», if the i-th soldier looks to the right, and can potentially see only soldiers $i+1, i+2, \ldots, n$.

Output

Output n integers, the i-th integer must be equal to the number of soldiers in the row that the i-th soldier sees.

Examples

standard input	standard output
4 178 180 170 190 LLLL	0 1 1 2
5 178 180 175 170 190 LLRLL	0 1 2 2 3
5 178 180 170 170 160 LLRLL	0 1 1 2 3

Problem E. Different Digits

Time limit: 1 second
Memory limit: 512 megabytes

Senya likes integers such that any two consecutive decimal digits in them are distinct. You are given an integer n. Help Senya to find the smallest integer he likes, which is strictly greater than n.

Input

Input contains integer n ($1 \le n \le 10^{18}$).

Output

Print the smallest integer strictly greater than n, such that any two consecutive digits in it are distinct.

Example

standard input	standard output
98	101

Problem F. Drawing

Time limit: 1 second
Memory limit: 512 megabytes

Math class is very boring, so Borya tries to find an entertainment. He has a sheet of grid paper that is divided into $4n$ rows and $4m$ columns. After each 4 rows he drew a line so the sheet is divided into n big rows each consisting of 4 small rows. In the same way he split $4m$ columns into m big ones. So he has his sheet divided into nm squares, each consisting of 16 small ones.

Borya suggested Misha to write down a number between 4 and 12 inclusive into each big cell. All those preparations were needed for testing Borya's painting skills. He wants to paint some of small cells in the way that the following conditions are satisfied.

First Borya decided that it is too bad that some cells have four neighbors, but others might have two or three. So Borya considers the first cell of each row to be the neighbor of the last cell of this row. Similarly the first and the last cells of each column are neighbors.

All painted cells should form a connected figure. That means that for each two painted cells there should exist a path of painted cells connecting them, such that each pair of adjacent cells in path are neighbors on the paper (in accordance with Borya's decision that for each row and column the first and the last cells are also neighbors).

Furthermore all not painted cells must also form a connected figure.

Finally if Misha wrote down number X in a big cell, exactly X small cells in it must be painted.

Borya is sure that no matter what numbers Misha writes down, he can paint some cells in such way that all the conditions are satisfied. Proof that you are as good as Borya in painting!

Input

In the first line there are two integers n and m ($1 \le n, m \le 100$). They mean that Borya has a sheet of grid paper of size n by m cells.

Next n lines contains m integers each: a_{ij} ($4 \le a_{ij} \le 12$) — number of small cells which Borya should paint in the j-th big cell of the i-th row.

Output

You must print $4n$ lines each consisting of $4m$ characters. For each small cell you should print «.» if Borya must not paint it, and symbol «*», if Borya must paint it.

If there are exist more than one solution, print any of them.

Example

standard input	standard output
3 6 5 8 8 7 6 6 7 5 5 4 7 7 5 7 5 5 7 4	....********************. .*......................*. .*.****..******..*.....*. .*.*..*..*...*..*.....*. .*.*.....*..**..*.....*. .*.****..*.**...******.*. .*....*..*......*...*.*. .*.*..*..*....*.*...*.*. .*.****..*....*.***...*. .**********************. ....

Problem G. The Final Battle

Time limit: 1 second
Memory limit: 512 megabytes

The final battle between Humans and Martians is coming soon. Spies of Humans have found out that the Martians have n soldiers left. It also turns out that Humans as well as the Martians have exactly n soldiers.

According to the experience of past battles, Humans know that only the Martian number i can defeat the Human number i.

The Human commander has decided to form a line of Humans. After analyzing Martians plans, the commander found out that a Human at the i-th position in the line will fight with the Martian number a_i. People will win only if each of the soldiers will win his battle.

Initially the commander has put the Human number i to the i-th position in the line. After that he realized that he had little time before the battle, and that Humans can lose. Each second he can move a solider from the last position to the beginning of the line, after this operation this soldier is at the first position, and the number of the position of each of the other fighters increases by 1.

Help him to find the minimal time he can rebuild the line so that Humans would win.

Input

The first line contains an integer n — the number of soldiers of each side ($1 \le n \le 2 \cdot 10^5$).

The second line contains n distinct integers $a_1, a_2, \ldots, a_n$, where a_i is the number of the Martian that the Human from the i-th position in the line will fight ($1 \le a_i \le n$, if $i \ne j$, then $a_i \ne a_j$).

Output

Output a single number k — the minimum number of seconds that is enough for the commander to rebuild the line so that Humans would win. If Humans can not defeat the Martians, print the number «-1».

Examples

standard input	standard output
5 1 4 2 3 5	2
5 1 3 5 2 4	-1

Note

In the first example, initially the soldiers stand opposite each other in the following way:

Martians:	1	4	2	3	5
Humans:	1	2	3	4	5

Humans lose, as the Martians number 1 and 5 win their fights. After the first move, the line of the soldiers changes to this line:

Martians:	1	4	2	3	5
Humans:	5	1	2	3	4

Now the Martians 2 and 3 win their fights, so the commander need to move the last soldier to the beginning of the line again. After it, the line of the soldiers becomes such that all Humans win their fights.

Martians:	1	4	2	3	5
Humans:	4	5	1	2	3

Problem H. Schedule

Time limit: 1 second
Memory limit: 512 megabytes

Anton goes to school from Monday to Friday. Lessons in his school last for 45 minutes and breaks are for 15 minutes. The first lesson starts at 7:00 a.m. and the last lesson finishes at 3:45 p.m. It's easy to see that each lesson starts at the beginning of an hour.

Anton wants to stay healthy, so he wants to sleep for 8 hours every day. He doesn't always get it, because lessons can start at 7 a.m. Therefore, in order to sleep 8 hours a day on average he can sleep more than 8 hours, on other days, for example, on Saturday and Sunday, when he doesn't go to school. Anton can't sleep more than 10 hours a day, because otherwise he would a feeling that he has slept too much and the rest of the day he would feel bad. Anton gets up at least an hour before the beginning of the lessons to have a breakfast and get to school. Anton falls asleep at midnight every day.

Anton is a good student, so he wants to be at all lessons in his school. But if one day, at a time when he needs to get up to go to school, he has a lack of sleep of at least k hours, he will oversleep his first lesson. Anton started to go to school on Monday. Initially Anton he has no lack of sleep, because in summer he had enough time to rest and sleep.

Help Anton to find out what day he will oversleep his first lesson.

Input

The first line contains k — the number of lack of sleep hours that Anton needs to oversleep his first lesson ($1 \le k \le 1000$).

The following five lines contain 9 characters each, 0-s and 1-s — the description of Anton's school schedule from Monday to Friday. On Saturday and Sunday Anton doesn't go to school. If on the i-th day of week Anton has a lesson with the number j, then the j-th character in the i-th line is 1, otherwise it is 0.

Output

If Anton never has k hours of lack of sleep, and never oversleeps his lessons, output «-1», otherwise output two integers — the number of the week and day of that week when Anton will skip his first lesson.

Example

standard input	standard output
11 111111100 111111100 111111100 111111100 111111100	2 3

Note

Every day Anton's lessons last from 7:00 a.m. to 1:45 p.m., so every day he will get up at 6 a.m. and accumulate 2 hours of lack of sleep. By the weekend Anton will accumulate 10 hours of lack of sleep, but he will be able to compensate for 4 of them, if he sleeps 10 hours on Saturday and Sunday. Starting the second week with 6 hours of lack of sleep, on Wednesday Anton will overspleep his first lesson.

Problem I. Pizza

Time limit: 2 seconds
Memory limit: 512 megabytes

Vasya is going to bake a pizza for his m friends. There are n additional ingredients at Vasya's disposal, each of them can either be put into pizza or not. Vasya may use all ingredients or even prepare a pizza without additional ingredients at all. Thus, there are 2^n possible pizza recipes.

Not just any pizza will make Vasya's friends happy, though. Every friend has prepared a wish list of the form "ingredient t should be included into the pizza" or "ingredient t shouldn't be included into the pizza". Vasya's friends aren't too choosy: any pizza which has at least one of friend's wishes satisfied will make the friend happy.

Calculate the number of ways Vasya can bake the pizza to make all friends happy. Since this number may be too large, output it modulo 998244353.

Input

The first line of the input contains two integers n and m — the number of ingredients and the number of Vasya's friends, respectively ($1 \le n \le 1000$, $1 \le m \le 20$).

Each of the next m lines corresponds to one of Vasya's friend and contains an integer a_i — the number of wishes on the wish list, followed by a_i integers $b_{i,j}$ — the description of wishes on the list ($1 \le a_i \le 100$, $-n \le b_{i,j} \le n$, $b_{i,j} \ne 0$). If $b_{i,j}$ is positive, the i-th friend has a wish "ingredient $b_{i,j}$ should be included into the pizza", if it's negative, the i-th friend has a wish "ingredient $-b_{i,j}$ shouldn't be included into the pizza".

Every ingredient occurs at most once in every list.

Output

Output the number of different pizzas making all friends happy, modulo 998244353.

Examples

standard input	standard output
4 3 2 1 3 3 2 -4 1 1 -2	5
68 1 1 -42	468704809

Note

In the first example, the following sets of ingredients will make all friends happy: (1), (3), (1, 3), (1, 4), (1, 3, 4).

In the second example, ingredient 42 shouldn't be included into the pizza, while all the other ingredients may be either included or not. The answer is equal to 2^{67} modulo 998244353.

Problem J. Retwinting twinter

Time limit: 1 second
Memory limit: 512 megabytes

Project twinter (from words «the winter») is a social network in which people can write about the winter that is coming. Each message in this network is called a twint. Beforehand the length of each twint was limited to 140 characters, but recently this limit was raised to 280.

If a user needs to post a longer message, she traditionally breaks it up into a series of several twints. In the end of each twint she puts its number and the total number of twints in the chain, for example:

```
Winter!.. The peasant breathes a sigh, (1/2)
```

```
renews his sledge, and makes his way. (2/2)
```

Creators of twinter decided to reformat old twint chains automatically according with the new limit of 280 characters. This limit includes « (i/n)» mark in the end of each twint. Help them reformat the given chain, placing it into as little twints as possible. Words (that is, sequences of non-space characters) may not be broken up between twints.

Input

The first line contains positive integer n — the number of twints in the initial chain ($1 \le n \le 5000$).

Each of the next n lines contains a twint. Its length is at most 140 characters and it ends with the string « (i/n)», where i is the number of this twint. Twints do not start with a space and do not contain two sequential spaces. Twints can only contain spaces, English letters, digits and punctuation marks (with ASCII codes from 33 to 63).

Output

In the first line output the number m — the number of twints of your reformatted shortest chain.

Then output the twints of your chain, one per line. They may not start with a space or contain two sequential spaces. They must end with the string « (i/m)», as in the input. The sequence of words in your chain must be the same as in the initial one.

Examples

standard input
2 Winter!.. The peasant breathes a sigh, (1/2) renews his sledge, and makes his way. (2/2)
standard output
1 Winter!.. The peasant breathes a sigh, renews his sledge, and makes his way. (1/1)

standard input
3 An example with sylla- (1/3) bification and punctuation (2/3) ! (3/3)
standard output
1 An example with sylla- bification and punctuation ! (1/1)

Problem K. Fractification

Time limit: 1 second
Memory limit: 512 megabytes

Andrew is styding fractions. He needs to do a difficult homework.

Let's say that *fractification* of four integers (a, b, c, d) is the sum

$$\frac{a}{b} + \frac{c}{d}.$$

Given four integers, the task is to order them in such way that their fractification is as small as possible.

Help Andrew to order the numbers to minimize their fractification.

Input

The first and only line of input contains four positive integers a, b, c and d ($1 \le a, b, c, d \le 10^9$).

Output

Print four integers which are the permutation of given four, so their fractification is minimal possible. If there are more than one such permutations, print any.

Examples

standard input	standard output
1 2 3 4	1 3 2 4
5 5 5 5	5 5 5 5

Note

In the first example output permutation gives us the value of

$$\frac{1}{3} + \frac{2}{4} = \frac{5}{6},$$

which is minimal.

In the second example only possible output leads to value of

$$\frac{5}{5} + \frac{5}{5} = 2.$$