Educational Codeforces Round 49 (Rated for Div. 2)

A. Palindromic Twist

2 seconds, 256 megabytes

You are given a string s consisting of n lowercase Latin letters. n is even.

For each position i $(1 \le i \le n)$ in string s you are required to change the letter on this position either to the previous letter in alphabetic order or to the next one (letters '`a`' and '`z`' have only one of these options). Letter in every position must be changed **exactly once**.

For example, letter '`p`' should be changed either to '`o`' or to '`q`', letter '`a`' should be changed to '`b`' and letter '`z`' should be changed to '`y`'.

That way string "`codeforces`", for example, can be changed to "`dpedepqbft`" ('`c`' → '`d`', '`o`' → '`p`', '`d`' → '`e`', '`e`' → '`d`', '`f`' → '`e`', '`o`' → '`p`', '`r`' → '`q`', '`c`' → '`b`', '`e`' → '`f`', '`s`' → '`t`').

String s is called a palindrome if it reads the same from left to right and from right to left. For example, strings "`abba`" and "`zz`" are palindromes and strings "`abca`" and "`zy`" are not.

Your goal is to check if it's possible to make string s a palindrome by applying the aforementioned changes to every position. Print "`YES`" if string s can be transformed to a palindrome and "`NO`" otherwise.

Each testcase contains several strings, for each of them you are required to solve the problem separately.

Input

The first line contains a single integer T $(1 \le T \le 50)$ — the number of strings in a testcase.

Then $2T$ lines follow — lines $(2i - 1)$ and $2i$ of them describe the i-th string. The first line of the pair contains a single integer n $(2 \le n \le 100$, n is even$)$ — the length of the corresponding string. The second line of the pair contains a string s, consisting of n lowercase Latin letters.

Output

Print T lines. The i-th line should contain the answer to the i-th string of the input. Print "`YES`" if it's possible to make the i-th string a palindrome by applying the aforementioned changes to every position. Print "`NO`" otherwise.

input

```
5
6
abccba
2
cf
4
adfa
8
abaazaba
2
ml
```

output

```
YES
NO
YES
NO
NO
```

The first string of the example can be changed to "`bcbbcb`", two leftmost letters and two rightmost letters got changed to the next letters, two middle letters got changed to the previous letters.

The second string can be changed to "`be`", "`bg`", "`de`", "`dg`", but none of these resulting strings are palindromes.

The third string can be changed to "`beeb`" which is a palindrome.

The fifth string can be changed to "`lk`", "`lm`", "`nk`", "`nm`", but none of these resulting strings are palindromes. Also note that no letter can remain the same, so you can't obtain strings "`ll`" or "`mm`".

B. Numbers on the Chessboard

1 second, 256 megabytes

You are given a chessboard of size $n \times n$. It is filled with numbers from 1 to n^2 in the following way: the first $\lceil \frac{n^2}{2} \rceil$ numbers from 1 to $\lceil \frac{n^2}{2} \rceil$ are written in the cells with even sum of coordinates from left to right from top to bottom. The rest $n^2 - \lceil \frac{n^2}{2} \rceil$ numbers from $\lceil \frac{n^2}{2} \rceil + 1$ to n^2 are written in the cells with odd sum of coordinates from left to right from top to bottom. The operation $\lceil \frac{x}{y} \rceil$ means division x by y rounded up.

For example, the left board on the following picture is the chessboard which is given for $n = 4$ and the right board is the chessboard which is given for $n = 5$.

1	9	2	10
11	3	12	4
5	13	6	14
15	7	16	8

1	14	2	15	3
16	4	17	5	18
6	19	7	20	8
21	9	22	10	23
11	24	12	25	13

You are given q queries. The i-th query is described as a pair x_i, y_i. The answer to the i-th query is the number written in the cell x_i, y_i (x_i is the row, y_i is the column). Rows and columns are numbered from 1 to n.

Input

The first line contains two integers n and q $(1 \le n \le 10^9, 1 \le q \le 10^5)$ — the size of the board and the number of queries.

The next q lines contain two integers each. The i-th line contains two integers x_i, y_i $(1 \le x_i, y_i \le n)$ — description of the i-th query.

Output

For each query from 1 to q print the answer to this query. The answer to the i-th query is the number written in the cell x_i, y_i (x_i is the row, y_i is the column). Rows and columns are numbered from 1 to n. Queries are numbered from 1 to q in order of the input.

input

```
4 5
1 1
4 4
4 3
3 2
2 4
```

output

```
1
8
16
13
4
```

input

```
5 4
2 1
4 2
3 3
3 4
```

output

```
16
9
7
20
```

Answers to the queries from examples are on the board in the picture from the problem statement.

C. Minimum Value Rectangle

2 seconds, 256 megabytes

You have n sticks of the given lengths.

Your task is to choose exactly four of them in such a way that they can form a rectangle. No sticks can be cut to pieces, each side of the rectangle must be formed by a single stick. No stick can be chosen multiple times. It is guaranteed that it is always possible to choose such sticks.

Let S be the area of the rectangle and P be the perimeter of the rectangle.

The chosen rectangle should have the value $\frac{P^2}{S}$ minimal possible. The value is taken without any rounding.

If there are multiple answers, print any of them.

Each testcase contains several lists of sticks, for each of them you are required to solve the problem separately.

Input

The first line contains a single integer T ($T \geq 1$) — the number of lists of sticks in the testcase.

Then $2T$ lines follow — lines $(2i - 1)$ and $2i$ of them describe the i-th list. The first line of the pair contains a single integer n ($4 \leq n \leq 10^6$) — the number of sticks in the i-th list. The second line of the pair contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_j \leq 10^4$) — lengths of the sticks in the i-th list.

It is guaranteed that for each list there exists a way to choose four sticks so that they form a rectangle.

The total number of sticks in all T lists doesn't exceed 10^6 in each testcase.

Output

Print T lines. The i-th line should contain the answer to the i-th list of the input. That is the lengths of the four sticks you choose from the i-th list, so that they form a rectangle and the value $\frac{P^2}{S}$ of this rectangle is minimal possible. You can print these four lengths in arbitrary order.

If there are multiple answers, print any of them.

input

```
3
4
7 2 2 7
8
2 8 1 4 8 2 1 5
5
5 5 5 5 5
```

output

```
2 7 7 2
2 2 1 1
5 5 5 5
```

There is only one way to choose four sticks in the first list, they form a rectangle with sides 2 and 7, its area is $2 \cdot 7 = 14$, perimeter is $2(2 + 7) = 18$. $\frac{18^2}{14} \approx 23.143$.

The second list contains subsets of four sticks that can form rectangles with sides $(1, 2)$, $(2, 8)$ and $(1, 8)$. Their values are $\frac{6^2}{2} = 18$, $\frac{20^2}{16} = 25$ and $\frac{18^2}{8} = 40.5$, respectively. The minimal one of them is the rectangle $(1, 2)$.

You can choose any four of the 5 given sticks from the third list, they will form a square with side 5, which is still a rectangle with sides $(5, 5)$.

D. Mouse Hunt

2 seconds, 256 megabytes

Medicine faculty of Berland State University has just finished their admission campaign. As usual, about 80% of applicants are girls and majority of them are going to live in the university dormitory for the next 4 (hopefully) years.

The dormitory consists of n rooms and a single mouse! Girls decided to set mouse traps in some rooms to get rid of the horrible monster. Setting a trap in room number i costs c_i burles. Rooms are numbered from 1 to n.

Mouse doesn't sit in place all the time, it constantly runs. If it is in room i in second t then it will run to room a_i in second $t + 1$ without visiting any other rooms inbetween ($i = a_i$ means that mouse won't leave room i). It's second 0 in the start. If the mouse is in some room with a mouse trap in it, then the mouse get caught into this trap.

That would have been so easy if the girls actually knew where the mouse at. Unfortunately, that's not the case, mouse can be in any room from 1 to n at second 0.

What it the minimal total amount of burles girls can spend to set the traps in order to guarantee that the mouse will eventually be caught no matter the room it started from?

Input

The first line contains as single integers n ($1 \leq n \leq 2 \cdot 10^5$) — the number of rooms in the dormitory.

The second line contains n integers $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq 10^4$) — c_i is the cost of setting the trap in room number i.

The third line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) — a_i is the room the mouse will run to the next second after being in room i.

Output

Print a single integer — the minimal total amount of burles girls can spend to set the traps in order to guarantee that the mouse will eventually be caught no matter the room it started from.

input

```
5
1 2 3 2 10
1 3 4 3 3
```

output

```
3
```

input

```
4
1 10 2 10
2 4 2 2
```

output

```
10
```

input

```
7
1 1 1 1 1 1 1
2 2 2 3 6 7 6
```

output

```
2
```

In the first example it is enough to set mouse trap in rooms 1 and 4. If mouse starts in room 1 then it gets caught immideately. If mouse starts in any other room then it eventually comes to room 4.

In the second example it is enough to set mouse trap in room 2. If mouse starts in room 2 then it gets caught immideately. If mouse starts in any other room then it runs to room 2 in second 1.

Here are the paths of the mouse from different starts from the third example:

- $1 \to 2 \to 2 \to \ldots$;
- $2 \to 2 \to \ldots$;
- $3 \to 2 \to 2 \to \ldots$;
- $4 \to 3 \to 2 \to 2 \to \ldots$;
- $5 \to 6 \to 7 \to 6 \to \ldots$;
- $6 \to 7 \to 6 \to \ldots$;
- $7 \to 6 \to 7 \to \ldots$;

So it's enough to set traps in rooms 2 and 6.

E. Inverse Coloring

2 seconds, 256 megabytes

You are given a square board, consisting of n rows and n columns. Each tile in it should be colored either white or black.

Let's call some coloring *beautiful* if each pair of adjacent rows are either the same or different in every position. The same condition should be held for the columns as well.

Let's call some coloring *suitable* if it is *beautiful* and there is no **rectangle** of the single color, consisting of at least k tiles.

Your task is to count the number of *suitable* colorings of the board of the given size.

Since the answer can be very large, print it modulo 998244353.

Input

A single line contains two integers n and k ($1 \le n \le 500$, $1 \le k \le n^2$) — the number of rows and columns of the board and the maximum number of tiles inside the rectangle of the single color, respectively.

Output

Print a single integer — the number of *suitable* colorings of the board of the given size modulo 998244353.

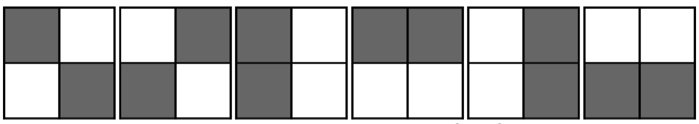
input
```
1 1
```
output
```
0
```

input
```
2 3
```
output
```
6
```

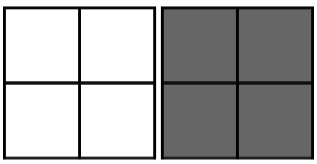
input
```
49 1808
```
output
```
359087121
```

Board of size 1×1 is either a single black tile or a single white tile. Both of them include a rectangle of a single color, consisting of 1 tile.

Here are the *beautiful* colorings of a board of size 2×2 that don't include rectangles of a single color, consisting of at least 3 tiles:



The rest of *beautiful* colorings of a board of size 2×2 are the following:



F. Session in BSU

4 seconds, 256 megabytes

Polycarp studies in Berland State University. Soon he will have to take his exam. He has to pass exactly n exams.

For the each exam i there are known two days: a_i — day of the first opportunity to pass the exam, b_i — day of the second opportunity to pass the exam ($a_i < b_i$). Polycarp *can pass at most one exam* during each day. For each exam Polycarp chooses by himself which day he will pass this exam. He has to pass all the n exams.

Polycarp wants to pass all the exams as soon as possible. Print the minimum index of day by which Polycarp can pass all the n exams, or print `-1` if he cannot pass all the exams at all.

Input

The first line of the input contains one integer n ($1 \le n \le 10^6$) — the number of exams.

The next n lines contain two integers each: a_i and b_i ($1 \le a_i < b_i \le 10^9$), where a_i is the number of day of the first passing the i-th exam and b_i is the number of day of the second passing the i-th exam.

Output

If Polycarp cannot pass all the n exams, print `-1`. Otherwise print the minimum index of day by which Polycarp can do that.

input
```
2
1 5
1 7
```
output
```
5
```

input
```
3
5 13
1 5
1 7
```
output
```
7
```

input
```
3
10 40
40 80
10 80
```
output
```
80
```

input
```
3
99 100
99 100
99 100
```
output
```
-1
```

G. X-mouse in the Campus

1.5 seconds, 256 megabytes

The campus has m rooms numbered from 0 to $m-1$. Also the x-mouse lives in the campus. The x-mouse is not just a mouse: each second x-mouse moves from room i to the room $i \cdot x \mod m$ (in fact, it teleports from one room to another since it doesn't visit any intermediate room). Starting position of the x-mouse is unknown.

You are responsible to catch the x-mouse in the campus, so you are guessing about minimum possible number of traps (one trap in one room) you need to place. You are sure that if the x-mouse enters a trapped room, it immediately gets caught.

And the only observation you made is $\text{GCD}(x, m) = 1$.

Input

The only line contains two integers m and x ($2 \le m \le 10^{14}$, $1 \le x < m$, $\text{GCD}(x, m) = 1$) — the number of rooms and the parameter of x-mouse.

Output

Print the only integer — minimum number of traps you need to install to catch the x-mouse.

input
4 3
output
3

input
5 2
output
2

In the first example you can, for example, put traps in rooms 0, 2, 3. If the x-mouse starts in one of this rooms it will be caught immediately. If x-mouse starts in the 1-st rooms then it will move to the room 3, where it will be caught.

In the second example you can put one trap in room 0 and one trap in any other room since x-mouse will visit all rooms $1..m-1$ if it will start in any of these rooms.