Codeforces Round 869 (Div. 2)

A. Politics

1 second, 256 megabytes

In a debate club with n members, including yourself (member 1), there are k opinions to be discussed in sequence. During each discussion, members express their agreement or disagreement with the opinion. Let's define Y as the number of members who agree and N as the number of members who disagree. After each discussion, members leave the club based on the following criteria:

- If more members agree than disagree ($Y > N$), all members who disagreed leave the club.
- If more members disagree than agree ($Y < N$), all members who agreed leave the club.
- If there is a tie ($Y = N$), all members leave the club.

As the club president, your goal is to stay in the club and maximize the number of members remaining after the meeting. You have access to each member's stance on all k opinions before the meeting starts, and you can expel any number of members (excluding yourself) before the meeting begins.

Determine the maximum number of members, including yourself, who can remain in the club after the meeting. You don't need to provide the specific expulsion strategy but only the maximum number of members that can stay. Ensure that you remain in the club after the meeting as well.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). Description of the test cases follows.

The first line of each test case contains two positive integers n and k ($1 \le n, k \le 100$) — the number of members and the number of discussions.

The i-th of the following n lines contains a string t_i of length k. The j-th character in the string t_i indicates whether the i-th member agrees or disagrees with the j-th opinion if they are present during that discussion. A "$+$" symbol means the member agrees, while a "$-$" symbol means the member disagrees.

It is guaranteed that the sum of $n \cdot k$ over all test cases does not exceed $5 \cdot 10^4$.

Output

For each test case, output the maximum number of members, including yourself, who can remain in the club after the meeting.

input
```
5
2 2
++
+-
1 3
+-+
4 1
+
-
-
+
5 4
++++
+--+
++-+
+-++
++++
4 2
++
--
--
-+
```

output
```
1
1
2
2
1
```

For convenience, we will analyze the examples based on who actually attended the meeting (i. e. was *not* expelled) rather than who was expelled.

Example 1:

Only the first member could have attended the meeting, otherwise both members would have left after the second opinion is discussed.

Example 2:

There is only a single member that attends the meeting and stays till the end.

Example 3:

The club has 4 members and only one opinion will be discussed during the meeting. Let's analyze the possible outcomes based on the participants in the meeting:

- If only the first member attends, they'll be the only one left after the meeting.
- If the first member attends with the second or third member, they will be a tie in the discussion, making them both leave.
- If the first member attends with the second and third members, the first member will be in the minority and will leave after the discussion, which contradicts the statement.
- If the first and fourth members attend, they will agree during the discussion and both remain till the end.
- If the first, second, and fourth members attend, the second member will be in the minority during the discussion, and only the first and fourth members will remain at the end. The same happens if the second member is replaced by the third member.
- If all four members attend, there will be a tie during the discussion, making everyone leave.

The maximum number of members remaining after the meeting is 2.

Example 4:

The club has 5 members and 4 opinions will be discussed during the meeting.

One way to achieve the maximum number of members is if only the first, third, and fifth members attend the meeting. In this case, they all agree during the first two discussions, after which the third member is in the minority during the third discussion. Then, the first and fifth members agree in the last discussion, and those two members stay till the end of the meeting.

Example 5:

The club has 4 members and 2 opinions will be discussed.

If the first three members attend the meeting, the first member will be in the minority during the first discussion and will leave the club. After that, the second and third members will both disagree with the second opinion, and they both will stay till the end of the meeting. In this way, there will be 2 members left after the meeting, but it is an invalid outcome, as it forces the first member to leave. Therefore, the maximum number of 1 member is achieved if only the first member attends the meeting.

B. Indivisible

1 second, 256 megabytes

You're given a positive integer n.

Find a permutation $a_1, a_2, \dots, a_n$ such that for any $1 \le l < r \le n$, the sum $a_l + a_{l+1} + \dots + a_r$ is not divisible by $r - l + 1$.

A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). Description of the test cases follows.

The first line of each test case contain a single integer n ($1 \le n \le 100$) — the size of the desired permutation.

Output

For each test case, if there is no such permutation print -1.

Otherwise, print n distinct integers $p_1, p_2, \dots, p_n$ ($1 \le p_i \le n$) — a permutation satisfying the condition described in the statement.

If there are multiple solutions, print any.

input
```
3
1
2
3
```
output
```
1
1 2
-1
```

In the first example, there are no valid pairs of $l < r$, meaning that the condition is true for all such pairs.

In the second example, the only valid pair is $l = 1$ and $r = 2$, for which $a_1 + a_2 = 1 + 2 = 3$ is not divisible by $r - l + 1 = 2$.

in the third example, for $l = 1$ and $r = 3$ the sum $a_1 + a_2 + a_3$ is always 6, which is divisible by 3.

C. Almost Increasing Subsequence

2 seconds, 256 megabytes

A sequence is *almost-increasing* if it does not contain three **consecutive** elements x, y, z such that $x \ge y \ge z$.

You are given an array $a_1, a_2, \dots, a_n$ and q queries.

Each query consists of two integers $1 \le l \le r \le n$. For each query, find the length of the longest *almost-increasing* subsequence of the subarray $a_l, a_{l+1}, \dots, a_r$.

A subsequence is a sequence that can be derived from the given sequence by deleting zero or more elements without changing the order of the remaining elements.

Input

The first line of input contains two integers, n and q ($1 \le n, q \le 200\,000$) — the length of the array a and the number of queries.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$) — the values of the array a.

Each of the next q lines contains the description of a query. Each line contains two integers l and r ($1 \le l \le r \le n$) — the query is about the subarray $a_l, a_{l+1}, \dots, a_r$.

Output

For each of the q queries, print a line containing the length of the longest almost-increasing subsequence of the subarray $a_l, a_{l+1}, \dots, a_r$.

input
```
9 8
1 2 4 3 3 5 6 2 1
1 3
1 4
2 5
6 6
3 7
7 8
1 8
8 8
```
output
```
3
4
3
1
4
2
7
1
```

In the first query, the subarray is $a_1, a_2, a_3 = [1, 2, 4]$. The whole subarray is almost-increasing, so the answer is 3.

In the second query, the subarray is $a_1, a_2, a_3, a_4 = [1, 2, 4, 3]$. The whole subarray is a almost-increasing, because there are no three consecutive elements such that $x \ge y \ge z$. So the answer is 4.

In the third query, the subarray is $a_2, a_3, a_4, a_5 = [2, 4, 3, 3]$. The whole subarray is not almost-increasing, because the last three elements satisfy $4 \ge 3 \ge 3$. An almost-increasing subsequence of length 3 can be found (for example taking $a_2, a_3, a_5 = [2, 4, 3]$). So the answer is 3.

D. Fish Graph

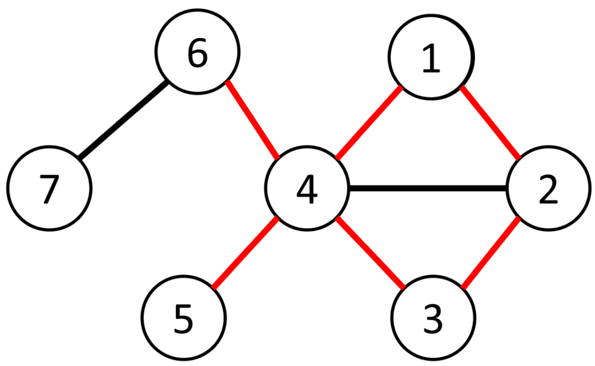
1 second, 256 megabytes

You are given a simple undirected graph with n nodes and m edges. Note that the graph is not necessarily connected. The nodes are labeled from 1 to n.

We define a graph to be a *Fish Graph* if it contains a simple cycle with a special node u belonging to the cycle. Apart from the edges in the cycle, the graph should have exactly 2 extra edges. Both edges should connect to node u, but they should not be connected to any other node of the cycle.

Determine if the graph contains a subgraph that is a Fish Graph, and if so, find any such subgraph.

In this problem, we define a subgraph as a graph obtained by taking any subset of the edges of the original graph.



Visualization of example 1. The red edges form one possible subgraph that is a Fish Graph.

Input

The first line of input contains the integer t ($1 \le t \le 1000$), the number of test cases. The description of test cases follows.

The first line of each test case contains two integers, n and m ($1 \le n, m \le 2\,000$) — the number of nodes and the number of edges.

Each of the next m lines contains the description of an edge. Each line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) — an edge connects node u_i to node v_i.

It is guaranteed that no two edges connect the same unordered pair of nodes.

Furthermore, it is guaranteed that the sum of n and the sum of m over all test cases both do not exceed $2\,000$.

Output

For each testcase, output "YES" if the graph contains a subgraph that is a Fish Graph, otherwise print "NO". If the answer is "YES", on the following lines output a description of the subgraph.

The first line of the description contains one integer k — the number of edges of the subgraph.

On the next k lines, output the edges of the chosen subgraph. Each of the k lines should contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the edge between u and v belongs to the subgraph. The order in which u and v are printed does not matter, as long as the two nodes are connected by an edge in the original graph. The order in which you print the edges does not matter, as long as the resulting subgraph is a fish graph.

If there are multiple solutions, print any.

input

```
3
7 8
1 2
2 3
3 4
4 1
4 5
4 6
4 2
6 7
7 7
6 7
1 2
2 3
3 4
4 1
1 3
3 5
4 4
1 3
3 4
4 1
1 2
```

output

```
YES
6
5 4
6 4
4 3
1 4
2 1
3 2
YES
5
5 3
2 3
3 1
4 3
1 4
NO
```

In the first example, a possible valid subgraph contains the cycle $1 \to 2 \to 3 \to 4 \to 1$. The special node of this cycle is node 4. The two extra edges $4 - 5$ and $4 - 6$ are both connected to 4, completing the Fish Graph.

In the second example, a possible valid subgraph contains the cycle $1 \to 3 \to 4 \to 1$. The special node of this cycle is node 3. The two extra edges $3 - 2$ and $3 - 5$ are both connected to 3, completing the Fish Graph.

In the last example, it can be proven that there is no valid subgraph.

E. Similar Polynomials

4 seconds, 256 megabytes

A polynomial $A(x)$ of degree d is an expression of the form $A(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_d x^d$, where a_i are integers, and $a_d \ne 0$. Two polynomials $A(x)$ and $B(x)$ are called similar if there is an integer s such that for any integer x it holds that

$$B(x) \equiv A(x+s) \pmod{10^9+7}.$$

For two similar polynomials $A(x)$ and $B(x)$ of degree d, you're given their values in the points $x = 0, 1, \dots, d$ modulo $10^9 + 7$.

Find a value s such that $B(x) \equiv A(x+s) \pmod{10^9+7}$ for all integers x.

Input

The first line contains a single integer d ($1 \le d \le 2\,500\,000$).

The second line contains $d+1$ integers $A(0), A(1), \ldots, A(d)$ ($0 \le A(i) < 10^9+7$) — the values of the polynomial $A(x)$.

The third line contains $d+1$ integers $B(0), B(1), \ldots, B(d)$ ($0 \le B(i) < 10^9+7$) — the values of the polynomial $B(x)$.

It is guaranteed that $A(x)$ and $B(x)$ are similar and that the leading coefficients (i.e., the coefficients in front of x^d) of $A(x)$ and $B(x)$ are not divisible by 10^9+7.

Output

Print a single integer s ($0 \le s < 10^9+7$) such that $B(x) \equiv A(x+s) \pmod{10^9+7}$ for all integers x.

If there are multiple solutions, print any.

input

```
1
1000000006 0
2 3
```

output

```
3
```

input

```
2
1 4 9
100 121 144
```

output

```
9
```

In the first example, $A(x) \equiv x - 1 \pmod{10^9+7}$ and $B(x) \equiv x + 2 \pmod{10^9+7}$. They're similar because

$$B(x) \equiv A(x+3) \pmod{10^9+7}.$$

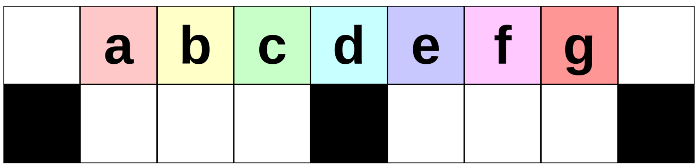
In the second example, $A(x) \equiv (x+1)^2 \pmod{10^9+7}$ and $B(x) \equiv (x+10)^2 \pmod{10^9+7}$, hence

$$B(x) \equiv A(x+9) \pmod{10^9+7}.$$

F. Toy Machine

1 second, 256 megabytes

There is a toy machine with toys arranged in two rows of n cells each (n is odd).



Initial state for $n = 9$.

Initially, $n-2$ toys are placed in the non-corner cells of the top row. The bottom row is initially empty, and its leftmost, rightmost, and central cells are blocked. There are 4 buttons to control the toy machine: left, right, up, and down marked by the letters `L`, `R`, `U`, and `D` correspondingly.

When pressing `L`, `R`, `U`, or `D`, all the toys will be moved simultaneously in the corresponding direction and will only stop if they push into another toy, the wall or a blocked cell. Your goal is to move the k-th toy into the leftmost cell of the top row. The toys are numbered from 1 to $n-2$ from left to right. Given n and k, find a solution that uses at most $1\,000\,000$ button presses.

To test out the toy machine, a **web page** is available that lets you play the game in real time.

Input

The first and only line contains two integers, n and k ($5 \le n \le 100\,000$, n is odd, $1 \le k \le n-2$) — the number of cells in a row, and the index of the toy that has to be moved to the leftmost cell of the top row.

Output

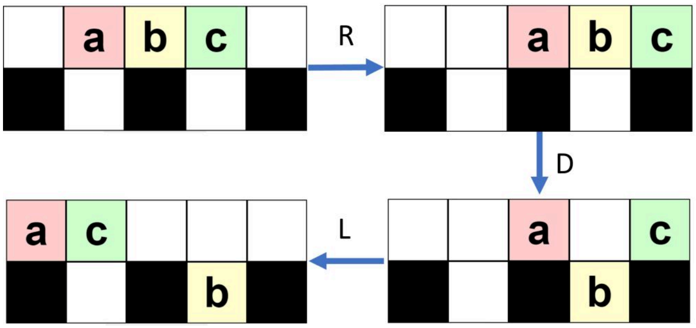
On a single line, output a description of the button presses as a string of at most $1\,000\,000$ characters. The string should only contain the characters `L`, `R`, `U`, and `D`. The i-th character in the string is the i-th button that is pressed. After all the button presses are performed, the k-th toy should be in the leftmost cell of the top row.

If there are multiple solutions, print any. The number of button presses does not have to be minimized.

input
5 1
output
RDL

input
7 2
output
RDL

In the first example, there will be $5-2=3$ toys. The first toy needs to end up in the leftmost cell of the top row. The moves `RDL` will achieve this, see the picture for a better understanding. Another possible solution would be to do one button press `L`.



Visualization of the moves for the first example.