A. Make it White

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

You have a horizontal strip of n cells. Each cell is either white or black.

You can choose a **continuous** segment of cells once and paint them all white. After this action, all the black cells in this segment will become white, and the white ones will remain white.

What is the minimum length of the segment that needs to be painted white in order for all n cells to become white?

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The descriptions of the test cases follow.

The first line of each test case contains a single integer n ($1 \le n \le 10$) — the length of the strip.

The second line of each test case contains a string s, consisting of n characters, each of which is either `'W'` or `'B'`. The symbol `'W'` denotes a white cell, and `'B'` — a black one. It is guaranteed that at least one cell of the given strip is black.

Output

For each test case, output a single number — the minimum length of a **continuous** segment of cells that needs to be painted white in order for the **entire** strip to become white.

Standard Input	Standard Output
8 6 WBBWBW 1 B 2 WB 3 BBW 4 BWWB 6 BWBWWB 6 WWBBWB 9 WBWBWWWBW	4 1 1 2 4 6 4 7

Note

In the first test case of the example for the strip "WBBWBW", the minimum length of the segment to be repainted white is 4. It is necessary to repaint to white the segment from the 2-nd to the 5-th cell (the cells are numbered

from 1 from left to right).

,

B. Following the String

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

Polycarp lost the string s of length n consisting of lowercase Latin letters, but he still has its *trace*.

The *trace* of the string s is an array a of n integers, where a_i is the number of such indices j ($j < i$) that $s_i = s_j$. For example, the *trace* of the string `abracadabra` is the array $[0, 0, 0, 1, 0, 2, 0, 3, 1, 1, 4]$.

Given a *trace* of a string, find **any** string s from which it could have been obtained. The string s should consist only of lowercase Latin letters `a`-`z`.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then the descriptions of the test cases follow.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the lost string.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < n$) — the *trace* of the string. It is guaranteed that for the given *trace*, there exists a suitable string s.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a string s that corresponds to the given *trace*. If there are multiple such strings s, then output any of them.

The string s should consist of lowercase Latin letters `a`-`z`.

It is guaranteed that for each test case, a valid answer exists.

Standard Input	Standard Output
5 11 0 0 0 1 0 2 0 3 1 1 4 10 0 0 0 0 0 1 0 1 1 0 1 0 8 0 1 2 3 4 5 6 7 8 0 0 0 0 0 0 0 0	abracadabra codeforces a aaaaaaaa dijkstra

,

C. Choose the Different Ones!

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

Given an array a of n integers, an array b of m integers, and an even number k.

Your task is to determine whether it is possible to choose **exactly** $\frac{k}{2}$ elements from both arrays in such a way that among the chosen elements, every integer from 1 to k is included.

For example:

- If $a = [2, 3, 8, 5, 6, 5]$, $b = [1, 3, 4, 10, 5]$, $k = 6$, then it is possible to choose elements with values $2, 3, 6$ from array a and elements with values $1, 4, 5$ from array b. In this case, all numbers from 1 to $k = 6$ will be included among the chosen elements.
- If $a = [2, 3, 4, 5, 6, 5]$, $b = [1, 3, 8, 10, 3]$, $k = 6$, then it is not possible to choose elements in the required way.

Note that you are not required to find a way to choose the elements — your program should only check whether it is possible to choose the elements in the required way.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The descriptions of the test cases follow.

The first line of each test case contains three integers n, m, and k ($1 \le n, m \le 2 \cdot 10^5$, $2 \le k \le 2 \cdot \min(n, m)$, k is even) — the length of array a, the length of array b, and the number of elements to be chosen, respectively.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the elements of array a.

The third line of each test case contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_j \le 10^6$) — the elements of array b.

It is guaranteed that the sum of values n and m over all test cases in a test does not exceed $4 \cdot 10^5$.

Output

Output t lines, each of which is the answer to the corresponding test case. As the answer, output "YES" if it is possible to choose $\frac{k}{2}$ numbers from each array in such a way that among the chosen elements, every integer from 1 to k is included. Otherwise, output "NO".

You can output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes", and "YES" will be accepted as a positive answer.

Standard Input	Standard Output
6 6 5 6 2 3 8 5 6 5 1 3 4 10 5	YES NO YES YES

6 5 6 2 3 4 5 6 5 1 3 8 10 3 3 3 4 1 3 5 2 4 6 2 5 4 1 4 7 3 4 4 2 1 4 2 2 6 4 4 2 1 5 2 3 2 2 1 4 3	NO NO

Note

In the first test case of the example, it is possible to choose elements equal to 2, 3, and 6 from array a and elements equal to 1, 4, and 5 from array b. Thus, all numbers from 1 to $k = 6$ are included among the chosen elements.

In the second test case of the example, it can be shown that it is not possible to choose exactly three elements from each array in the required way.

In the third test case of the example, it is possible to choose elements equal to 1 and 3 from array a and elements equal to 2 and 4 from array b. Thus, all numbers from 1 to $k = 4$ are included among the chosen elements.

,

D. Find the Different Ones!

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 256 megabytes

You are given an array a of n integers, and q queries.

Each query is represented by two integers l and r ($1 \le l \le r \le n$). Your task is to find, for each query, two indices i and j (or determine that they do not exist) such that:

- $l \le i \le r$;
- $l \le j \le r$;
- $a_i \neq a_j$.

In other words, for each query, you need to find a pair of different elements among $a_l, a_{l+1}, \ldots, a_r$, or report that such a pair does not exist.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. The descriptions of the test cases follow.

The first line of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the length of the array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$) — the elements of the array a.

The third line of each test case contains a single integer q ($1 \le q \le 2 \cdot 10^5$) — the number of queries.

The next q lines contain two integers each, l and r ($1 \le l < r \le n$) — the boundaries of the query.

It is guaranteed that the sum of the values of n across all test cases does not exceed $2 \cdot 10^5$. Similarly, it is guaranteed that the sum of the values of q across all test cases does not exceed $2 \cdot 10^5$.

Output

For each query, output two integers separated by space: i and j ($l \le i, j \le r$), for which $a_i \neq a_j$. If such a pair does not exist, output $i = -1$ and $j = -1$.

You may separate the outputs for the test cases with empty lines. This is not a mandatory requirement.

Standard Input	Standard Output
5 5 1 1 2 1 1 3 1 5 1 2 1 3 6 30 20 20 10 10 20 5	2 3 -1 -1 1 3 2 1 -1 -1 4 2 4 6 5 3

1 2 2 3 2 4 2 6 3 5 4 5 2 3 4 4 1 2 1 4 2 3 2 4 5 1 4 3 2 4 5 1 5 2 4 3 4 3 5 4 5 5 2 3 1 4 2 7 1 2 1 4 1 5 2 4 2 5 3 5 4 5	1 2 1 2 2 3 3 2 1 3 2 4 3 4 5 3 5 4 1 2 4 2 1 3 2 3 3 2 5 4 5 4

,

E. Klever Permutation

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

You are given two integers n and k ($k \leq n$), where k is even.

A permutation of length n is an array consisting of n distinct integers from 1 to n in any order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (as 2 appears twice in the array) and $[0,1,2]$ is also not a permutation (as $n=3$, but 3 is not present in the array).

Your task is to construct a k-*level* permutation of length n.

A permutation is called k-*level* if, among all the sums of continuous segments of length k (of which there are exactly $n-k+1$), any two sums differ by no more than 1.

More formally, to determine if the permutation p is k-*level*, first construct an array s of length $n-k+1$, where $s_i = \sum_{j=i}^{i+k-1} p_j$, i.e., the i-th element is equal to the sum of $p_i, p_{i+1}, \ldots, p_{i+k-1}$.

A permutation is called k-*level* if $\max(s) - \min(s) \leq 1$.

Find **any** k-*level* permutation of length n.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. This is followed by the description of the test cases.

The first and only line of each test case contains two integers n and k ($2 \leq k \leq n \leq 2 \cdot 10^5$, k is even), where n is the length of the desired permutation.

It is guaranteed that the sum of n for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output **any** k-*level* permutation of length n.

It is guaranteed that such a permutation always exists given the constraints.

Standard Input	Standard Output
5 2 2 3 2 10 4 13 4 7 4	2 1 1 3 2 1 8 4 10 2 7 5 9 3 6 4 10 1 13 5 9 2 12 6 8 3 11 7 1 6 3 7 2 5 4

Note

In the second test case of the example:

- $p_1 + p_2 = 3 + 1 = 4$;
- $p_2 + p_3 = 1 + 2 = 3$.

The maximum among the sums is 4, and the minimum is 3.

,

F. Microcycle

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

Given an undirected weighted graph with n vertices and m edges. There is at most one edge between each pair of vertices in the graph, and the graph does not contain loops (edges from a vertex to itself). The graph is not necessarily connected.

A cycle in the graph is called simple if it doesn't pass through the same vertex twice and doesn't contain the same edge twice.

Find any simple cycle in this graph in which the weight of the lightest edge is minimal.

Input

The first line of the input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Then follow the descriptions of the test cases.

The first line of each test case contains two integers n and m ($3 \le n \le m \le \min(\frac{n \cdot (n-1)}{2}, 2 \cdot 10^5)$) — the size of the graph and the number of edges.

The next m lines of the test case contain three integers u, v, and w ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^6$) — vertices u and v are connected by an edge of weight w.

It is guaranteed that there is at most one edge between each pair of vertices. Note that under the given constraints, there is always at least one simple cycle in the graph.

It is guaranteed that the sum of the values of m for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a pair of numbers b and k, where:

- b — the minimum weight of the edge in the found cycle,
- k — the number of vertices in the found cycle.

On the next line, output k numbers from 1 to n — the vertices of the cycle in traversal order.

Note that the answer always exists, as under the given constraints, there is always at least one simple cycle in the graph.

Standard Input	Standard Output
5 6 6 1 2 1 2 3 1 3 1 1 4 5 1 5 6 1 6 4 1 6 6	1 3 1 2 3 3 3 6 4 5 1 5 4 2 1 6 3 1 4 1 4 3 2

1 2 10 2 3 8 3 1 5 4 5 100 5 6 40 6 4 3 6 15 1 2 4 5 2 8 6 1 7 6 3 10 6 5 1 3 2 8 4 3 4 5 3 6 2 6 6 5 4 5 4 1 3 6 4 5 4 2 1 3 1 7 1 5 5 4 6 2 3 2 1 3 10 1 4 1 3 4 7 2 4 5 1 2 2 4 5 2 1 10 3 1 3 4 2 6 1 4 7 2 3 3	3 3 2 3 1

,

G. Paint Charges

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

A horizontal grid strip of n cells is given. In the i-th cell, there is a paint charge of size a_i. This charge can be:

- either used to the left — then all cells to the left at a distance less than a_i (from $\max(i - a_i + 1, 1)$ to i inclusive) will be painted,
- or used to the right — then all cells to the right at a distance less than a_i (from i to $\min(i + a_i - 1, n)$ inclusive) will be painted,
- or not used at all.

Note that a charge can be used no more than once (that is, it **cannot** be used simultaneously to the left and to the right). It is allowed for a cell to be painted more than once.

What is the minimum number of times a charge needs to be used to paint all the cells of the strip?

Input

The first line of the input contains an integer t ($1 \le t \le 100$) — the number of test cases in the test. This is followed by descriptions of t test cases.

Each test case is specified by two lines. The first one contains an integer n ($1 \le n \le 100$) — the number of cells in the strip. The second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$), where a_i is the size of the paint charge in the i-th cell from the left of the strip.

It is guaranteed that the sum of the values of n in the test does not exceed 1000.

Output

For each test case, output the minimum number of times the charges need to be used to paint all the cells of the strip.

Standard Input	Standard Output
13 1 1 2 1 1 2 2 1 2 1 2 2 2 2 3 1 1 1 3 3 1 2 3	1 2 1 1 1 3 1 2 3 4 2 3 3

1 3 1 7 1 2 3 1 2 4 2 7 2 1 1 1 2 3 1 10 2 2 5 1 6 1 8 2 8 2 6 2 1 2 1 1 2 6 1 1 4 1 3 2	

Note

In the third test case of the example, it is sufficient to use the charge from the 1-st cell to the right, then it will cover both cells 1 and 2.

In the ninth test case of the example, you need to:

- use the charge from the 3-rd cell to the left, covering cells from the 1-st to the 3-rd;
- use the charge from the 5-th cell to the left, covering cells from the 4-th to the 5-th;
- use the charge from the 7-th cell to the left, covering cells from the 6-th to the 7-th.

In the eleventh test case of the example, you need to:

- use the charge from the 5-th cell to the right, covering cells from the 5-th to the 10-th;
- use the charge from the 7-th cell to the left, covering cells from the 1-st to the 7-th.