2021 KAIST RUN Spring Contest

Sponsored By:

Jane Street

MOLOCO

NAVER D2

Rules

- This contest is 2021 KAIST RUN Spring Contest.
- This contest is sponsored by Jane Street, MOLOCO, and Naver D2.
- You can only participate individually.
- Use of the network is prohibited during the competition, except for submitting source codes and accessing language reference sites, and using translation. Here are the allowed reference and translation sites.
  - `C/C++           : https://en.cppreference.com/w/`
  - `Java            : https://docs.oracle.com/javase/8/docs/api/`
  - `Python          : https://docs.python.org/`
  - `Kotlin          : https://kotlinlang.org/docs/reference/`
  - `Naver Papago    : https://papago.naver.com/`
  - `Google Translate: https://translate.google.com/`
- The contest lasts 4 hours.
- The contest consists of 8 problems.
- You are allowed to submit solutions every 10 seconds for each problem. This limit may be increased due to heavy load.
- You are allowed to submit at most 100 submissions for each problem.
- Problems are ordered roughly by the difficulty of getting full points. You are strongly encouraged to read all the problems.
- Each problem consists of 1 or more subtasks, and each subtask is worth a certain number of points. Subtasks are easier versions of the problems with more restrictive constraints.
- Each subtask consists of several test cases, and points for the subtask are awarded if every test case is passed. The total number of points for each problem is 100 points.
- Each problem has a time limit and memory limit. This means your problem should run in the given time and memory limit for each test case.
- Every problem is guaranteed to be solvable using C++17. This is not guaranteed for any other language.
- The memory limit for every problem is 1024MiB.
- Each participant's ranking is determined in the following way.
  - Penalty time = (Duration from contest start to the last submission that increased points)
  - Ranking = (# of participants with higher points) + (# of participants with same points and lower penalty time) + 1

Language Guide

- You can choose your programming language from among the following:
  - C11: gcc (Ubuntu 10.2.0-5ubuntu1 20.04) 10.2.0
  - C++17: g++ (Ubuntu 10.2.0-5ubuntu1 20.04) 10.2.0
  - Java: OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
  - Kotlin: kotlinc-jvm 1.5.0 (JRE 11.0.11+9-Ubuntu-0ubuntu2.20.04)
  - PyPy 3: Python 3.7.10 (7.3.4+dfsg-1 ppa1 ubuntu20.04, Apr 12 2021, 21:32:29) [PyPy 7.3.4 with GCC 9.3.0]
- In Java, your class name which includes the `main` method should be `Main`.
- Compilation commands are available at the contest system.
- Following is sample code which reads space-separated integers from standard input, and prints their sum to standard output.
  - C++17

```
#include <iostream>
using namespace std;

int main() {
    auto a=0, b=0;
    cin >> a >> b;
    cout << a+b << endl;
    return 0;
}
```

  - Java

```
import java.util.*;

public class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int a, b;
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println(a + b);
    }
}
```

  - Kotlin

```
import java.util.Scanner

fun main(args: Array<String>) {
    val sc: Scanner = Scanner(System.‘in‘)
    var a = sc.nextInt()
    var b = sc.nextInt()
    println(a+b)
}
```

– PyPy 3

```
import sys

def main():
    a, b = map(int, sys.stdin.readline().split())
    print(a+b)

if __name__ == '__main__':
    main()
```

Problem list

#	Problem Name	Time limit (All languages)	Score	Subtask score				
			Full score	1	2	3	4	5
A	Bingo	1 second	100	19	35	46		
B	Histogram Sequence 2	1 second	100	10	5	10	15	60
C	Opinion Pool	1 second	100	10	15	75		
D	Three Slices	1 second	100	10	20	70		
E	Factory Balls	1 second	100	16	84			
F	AND PLUS OR	3 seconds	100	14	17	69		
G	Endless Road	5 seconds	100	7	9	17	13	54
H	Fake Plastic Trees 2	3 seconds	100	9	24	10	57	

Problem A. Bingo

Time limit: 1 second

Bingo is a game on a square grid. Each player gets an $n \times n$ grid and writes a unique number in each cell. The game host then draws a random number, and each player looks for that number on their grid and, if the number is present on their grid, fills in the corresponding cell. This repeats until someone finds n filled cells on a single line, which we will call a **bingo line**.

There are $2n + 2$ possible bingo lines: n horizontal lines, n vertical lines, and 2 diagonal lines.

```
---  ...  ...  |..  .|.  ..|  \..  ../
...  ---  ...  |..  .|.  ..|  .\.  ./.
...  ...  ---  |..  .|.  ..|  ..\  /..
```

For example, the following grid has four bingo lines: two horizontal lines, one vertical line, and one diagonal line.

```
#..#.
#####
..###
#####
..###
```

Exactly when is a bingo line formed? That is completely random: you can get a line quite early if you are lucky, but on the other hand, you can fill most of the grid without getting any bingo lines. In this problem, we investigate the unfortunate case of filling k cells without making any bingo lines.

Given two integers n and k, determine whether it is possible to fill exactly k cells in an $n \times n$ grid, without making any bingo lines. If it's possible, demonstrate how to do it.

Input

The first and only line of input contains two integers, n and k.

Output

On the first line, output `YES` if it's possible to fill exactly k cells of an $n \times n$ grid without making any bingo lines. Otherwise, output `NO`.

If the answer is `YES`, then output each row of the grid starting from the next line. Each row should be represented by a string of n characters. The i-th character is `#` (ASCII 35) if the i-th cell of the row is filled, and `.` (ASCII 46) if it's not filled. Exactly k cells must be filled, and there cannot be any bingo lines.

If there are multiple ways to fill the grid, then output any one of them.

Constraints

- $1 \le n \le 100$
- $0 \le k \le n^2$

Subtask 1 (19 points)

This subtask has an additional constraint:

- $k \le n - 1$

Subtask 2 (35 points)

This subtask has an additional constraint:

- $n \leq 4$

Subtask 3 (46 points)

This subtask has no additional constraints.

Examples

standard input	standard output
4 2	YES ##.. ....
4 16	NO

Notes

Note that the second example is only valid for subtasks 2 and 3.

Problem B. Histogram Sequence 2

Time limit: 1 second

A 2-dimensional grid can be represented as a set of pairs of positive integers. Each cell can be numbered as in the figure below:

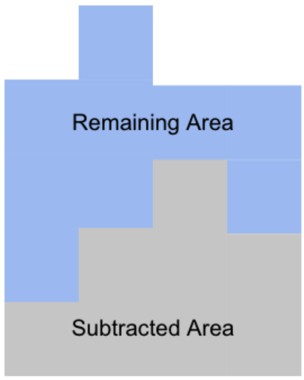
(1,3)	(2,3)	(3,3)	(4,3)	...
(1,2)	(2,2)	(3,2)	(4,2)	...
(1,1)	(2,1)	(3,1)	(4,1)	...

Suppose you are given a histogram where the height of each column is $h_1, h_2, \cdots, h_N$. Its area can be represented as a set containing cells $(i, 1), (i, 2), \cdots, (i, h_i)$ for $i = 1, 2, \cdots, N$. (If $h_i = 0$, it means the histogram's area does not contain any cell from the i-th column.)

You may choose N integers $x_1, x_2, \cdots, x_N$ with $0 \leq x_i < h_i$ and subtract a sub-histogram with heights $x_1, x_2, \cdots, x_N$. After removing such a sub-histogram, the remaining area can be written as:

$$\bigcup_{i=1}^{N} \{(i, j) : x_i < j \leq h_i\}.$$

For example, the following figure shows an example of $h_1 = 4$, $h_2 = 5$, $h_3 = 4$, $h_4 = 4$ and $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, $x_4 = 2$.



The remaining area is called *connected* if for every pair of cells $((r_1, c_1), (r_2, c_2))$ in the remaining area, one can move from (r_1, c_1) to (r_2, c_2) using only the following moves without leaving the remaining area:

- $(r, c) \to (r + 1, c)$
- $(r, c) \to (r - 1, c)$
- $(r, c) \to (r, c + 1)$
- $(r, c) \to (r, c - 1)$

Compute the number of ways to choose $(x_1, x_2, \cdots, x_N)$ such that the remaining area is connected.

Input

The first line contains a single integer, N. The second line contains N integers - $h_1, h_2, \cdots, h_N$ in order.

Output

Output the number of ways to choose $(x_1, x_2, \cdots, x_N)$ such that the remaining area is connected, modulo $10^9 + 7$.

Constraints

- $1 \le N \le 250\,000$
- $1 \le h_i \le 10^9$

Subtask 1 (10 points)

This subtask has additional constraints:

- $N \le 3$
- $h_i \le 100$ for $i = 1, 2, \cdots, N$

Subtask 2 (5 points)

This subtask has an additional constraint:

- $N \le 3$

Subtask 3 (10 points)

This subtask has an additional constraint:

- $h_1 = h_2 = \cdots = h_N$

Subtask 4 (15 points)

This subtask has an additional constraint:

- $h_1 \le h_2 \le \cdots \le h_N$

Subtask 5 (60 points)

This subtask has no additional constraints.

Examples

standard input	standard output
1 100	100
3 2 3 4	12
3 100 100 100	1000000
2 100000 100000	999999937
2 3 42	9
3 1 1 1	1

Problem C. Opinion Pool

Time limit: 1 second

MOLOCO, a global company with an unbeatable global reach, is developing a new survey platform to increase user engagement.

There are N people who want to vote on an issue. Each person is either in support of the issue or against the issue.

There are M not necessarily disjoint sets of people $S_1, S_2, \cdots, S_M$. For these M sets and a constant p ($0 \le p \le 1$), the propositions below are established.

- For every set S_i, at least $p \cdot |S_i|$ people belonging to S_i is in support of the issue.

If $p = 0$, no information can be obtained from this proposition. $p = 1$ shows that everyone is in support of the issue. That is, as p grows, it becomes easier to ascertain who is in support of the issue.

Thus, if a proposition is established for a sufficiently large p, we can know that everyone is in support of the issue. Find the maximum value of p such that you can not be certain everyone is in support.

Input

The first line contains two integers N and M, where N denotes the number of people and M denotes the number of sets.

The next M lines describe the information of each of the M sets.

The ith line starts with an integer $|S_i|$, denoting the number of elements in the set S_i, followed by $|S_i|$ distinct integers $S_{i,j}$, denoting the elements in the set S_i.

Output

Output the maximum value of p such that you cannot be certain everyone is in support of the issue.

Your answer will be considered correct if it has an absolute or relative error less than 10^{-6}.

Constraints

- $1 \le N, M \le 200\,000$
- $S_i \subseteq \{1, 2, \cdots, N\}$ $(1 \le i \le M)$
- $\sum_{i=1}^{M} |S_i| \le 1\,000\,000$
- Everyone appears in at least one set.

Subtask 1 (10 points)

This subtask has additional constraints:

- $N \le 10$
- $M \le 500$

Subtask 2 (15 points)

This subtask has an additional constraint:

- $N, M \le 500$

Subtask 3 (75 points)

This subtask has no additional constraints.

Examples

standard input	standard output
3 3 1 1 1 2 1 3	0
4 2 2 1 2 2 3 4	0.5
10 7 4 8 6 10 5 4 9 5 6 1 4 4 8 1 10 4 1 5 9 3 4 6 10 5 1 4 8 3 1 10 6 5 7 6 8 1 2	0.833333333

Note

In example 2, the proposition can be established for $p = 0.5$, if people 1, 3 are for and people 2, 4 are against.

However, if the proposition is established for $p > 0.5$, it is a contradiction to the proposition if there exists a person against an issue.

Problem D. Three Slices

Time limit: 1 second

You are given an array A_0, A_1, A_2, ... , A_{N-1} of N positive integers. Also, you are given an positive integer K. Your task is to find the largest positive integer M such that the following condition is satisfied:

- There exists an integer $0 \le i \le N - 3M$ such that
  1. $\sum_{j=i}^{i+M-1} A_j \le K$
  2. $\sum_{j=i+M}^{i+2M-1} A_j \le K$
  3. $\sum_{j=i+2M}^{i+3M-1} A_j \le K$

Input

The first line contains two integers, N and K. The second line contains N integers, the array A given in order.

Output

Output a single positive integer denoting the largest possible M. If there is no such M, output 0.

Constraints

- $1 \le N \le 5 \times 10^5$
- $1 \le K \le 10^9$
- $1 \le A_i \le 10^9$

Subtask 1 (10 points)

This subtask has an additional constraint:

- $N \le 100$

Subtask 2 (20 points)

This subtask has an additional constraint:

- $N \le 10000$

Subtask 3 (70 points)

This subtask has no additional constraints.

Examples

standard input	standard output
10 10 3 7 4 3 1 3 5 2 5 1	2

Problem E. Factory Balls

Time limit: 1 second

The year 2021 marks the death of *Adobe Flash*, a multimedia software for animations, applications, games, and so on. Together with it, the era of Flash games (especially from 2005 to 2012) has started to fade away into history. Some good news on this front is that people have already been working on a way to preserve Flash content. Major efforts include *Ruffle*: an open source Flash player written in Rust, and *Flashpoint*: an archive of over 78,000 Flash applications.

In memory of the golden era of Flash games, we present a simplified variant of a famous Flash game called *Factory Balls*. You are tasked with painting a ball in a given pattern. The surface of a ball can be divided into N distinct regions, enumerated with indices from 1 to N. You have K paint cans full of paint, where the i-th paint can has the color i. You are also given M pieces of equipment. Each piece of equipment is specified by a set of regions, and it precisely covers that subset of the regions of the ball.

In the beginning, all regions have color 1 and all pieces of equipment are unequipped. You may perform one of the following actions any number of times:

1. Immerse the ball into the i-th paint can. Every region not covered by any of the pieces of equipment on the ball will be painted with color i.
2. Pick one piece of equipment currently not equipped, and equip it. You can equip multiple pieces of equipment.
3. Pick one piece of equipment currently equipped, and unequip it.

In the end, each region of the ball should have a specific color, and all pieces of equipment should be unequipped. Find the minimum number of actions required to paint the ball, or report that it is impossible.

Input

The first line contains three integers: N, K, and M.

The next line contains N integers. The i-th integer c_i is the desired color of the i-th region.

Each of the next M lines describes a piece of equipment. The first number of each line, r_j, denotes the number of regions the piece of equipment covers, followed by r_j distinct positive integers, denoting the indices of the regions the piece of equipment covers.

Output

If it's not possible to paint each region of the ball to a given color, output `-1`. Otherwise, output the minimum number of actions required.

Constraints

- $1 \le N, K \le 10$
- $0 \le M \le 10$
- $1 \le c_i \le K$
- $1 \le r_j \le N$, and for each piece of equipment, all indices of the regions are distinct.

Subtask 1 (16 points)

This subtask has additional constraints:

- $N = M$
- For each $j = 1, 2, \cdots, N$, the j-th piece of equipment covers only the j-th region.

Subtask 2 (84 points)

This subtask has no additional constraints.

Examples

standard input	standard output
3 5 3 1 3 5 1 1 1 2 1 3	6
4 3 2 3 3 2 1 2 1 2 2 2 3	7
4 2 2 1 2 2 1 2 1 2 2 3 4	-1
2 10 0 1 1	0

Notes

In the second example, this is the fastest method:

- Immerse the ball into the third paint can.
- Equip the first piece of equipment.
- Immerse it into the second paint can.
- Equip the second piece of equipment.
- Immerse it into the first paint can.
- Unequip both pieces of equipment.

Problem F. AND PLUS OR

Time limit: 3 seconds

For two nonnegative integers a, b, let $a \wedge b$ be their bitwise AND, and $a \vee b$ be their bitwise OR.

You are given an array $A_0, A_1, \ldots, A_{2^N-1}$ of length 2^N consisting of nonnegative integers. Please find a pair of indices $0 \le i, j \le 2^N - 1$ such that $A_i + A_j < A_{i \wedge j} + A_{i \vee j}$, or state that no such pair exists. If there is more than one such pair, print any.

Input

The first line contains an integer N.

The second line contains 2^N integers, the array A given in order.

Output

If there is an answer, output two integers i, j denoting the answer, separated by spaces. i, j should be in the range $[0, 2^N - 1]$. Otherwise, output `-1`.

Constraints

- $0 \le N \le 20$
- $0 \le A_i \le 10^7$

Subtask 1 (14 points)

This subtask has an additional constraint.

- $N \le 14$

Subtask 2 (17 points)

This subtask has an additional constraint.

- $N \le 17$

Subtask 3 (69 points)

This subtask has no additional constraints.

Examples

standard input	standard output
2 0 1 1 2	-1
2 0 1 1 3	1 2

Problem G. Endless Road

Time limit: 5 seconds

Whenever students of KAIST go outside of campus, they usually follow a straight road named the *Endless Road*. The name comes from the fact that most students feel like the road is *endless*. One factor for this is due to the boring scenery of the road. Here, we model the Endless Road as a straight line of length 10^9. The position over the straight line can be denoted as a single real number $x \in [0, 10^9)$.

To cope with this, the members of RUN@KAIST decided to plant different kinds of colorful flowers on this road. N members of RUN@KAIST are numbered from $1 \ldots N$. In the last regular meeting, each member is assigned a nonempty segment over the straight line $[L_i, R_i)$, containing all positions $L_i \le x < R_i$.

The member with a higher index bears more responsibility for the tasks on the club. Thus, **the length of the segment is nondecreasing** as the indices of the members increases. In other words, $R_i - L_i \le R_j - L_j$ for $1 \le i < j \le N$.

The planting is done in N turns. In each turn, we select one previously unselected member to plant flowers. The selected member plants the flowers in their assigned segment $[L_i, R_i)$, except where other members have already planted flowers before.

To make the work distribution more equitable, the selection is done in the following way:

- Pick the member who will plant the minimum amount of flowers. Here, the amount of flowers are determined by the length of the interval where a member planted new flowers, which could be zero.
- If there is more than one such member, pick the member with minimum index.

Jaeung should now announce the planting schedule. For this, he has to find an order in which the N members will plant the flowers, according to the above selection rules.

Input

The first line contains an integer N.

The next N lines each contain two integers L_i and R_i.

Output

Output N space-separated integers denoting the order of planting. The i-th integer denotes the index of member who will plant the flower in the i-th turn.

Constraints

- $1 \le N \le 250\,000$
- $0 \le L_i < R_i \le 10^9$
- $R_i - L_i \le R_j - L_j$ for all $1 \le i < j \le N$.
- All intervals are distinct. In other words, $(L_i, R_i) \ne (L_j, R_j)$ for all $1 \le i < j \le N$.

Subtask 1 (7 points)

This subtask has an additional constraint.

- $N \le 100$
- $R_i \le 100$ for all $1 \le i \le N$

Subtask 2 (9 points)

This subtask has an additional constraint.

- $N \leq 10\,000$

Subtask 3 (17 points)

This subtask has an additional constraints.

- There is **no pair** of members $1 \leq i, j \leq N$ such that $i \neq j$, $L_i < L_j$, $R_j < R_i$.

Subtask 4 (13 points)

This subtask has an additional constraints.

- There is **no pair** of members $1 \leq i, j \leq N$ such that $i \neq j$, $L_i < L_j < R_i < R_j$.

Subtask 5 (54 points)

This subtask has no additional constraints.

Examples

standard input	standard output
6 1 2 2 3 3 4 4 5 1 3 3 5	1 2 5 3 4 6
4 3 7 10 14 1 6 6 11	1 3 2 4

Problem H. Fake Plastic Trees 2

Time limit: 3 seconds

You are given a tree with N vertices. Vertices are indexed from 1 to N. The tree is vertex-weighted. In other words, each vertex of the tree is assigned a nonnegative integer weight.

We will delete some edges from the tree. After the deletion, for each connected component, the sum of vertex weights should be in the range $[L, R]$.

For all integers $0 \le i \le K$, determine if we can achieve this goal by deleting **exactly** i edges.

Input

The first line contains a single integer T, the number of test cases. T test cases follow, each following the given specification:

The first line contains four integers N, K, L, and R.

The next line contains N integers $A_1, A_2, \ldots, A_N$, where A_i denotes the weight of vertex i.

The next $N - 1$ lines contain two integers x, y, denoting the pair of vertices connected by an edge.

Output

For each test case, output a binary string of length $K + 1$. The i-th character should be `1` if it is possible to achieve the desired goal by deleting exactly $i - 1$ edges. Otherwise, the i-th character should be `0`.

Constraints

- $1 \le N \le 1\,000$
- $0 \le K \le \min(50, N - 1)$
- $0 \le L \le R \le 10^{18}$
- $0 \le A_i \le 10^{18}$
- $1 \le x, y \le N$
- $x \ne y$
- The given graph is a tree.
- For all test cases, the sum of N is at most $1\,000$.

Subtask 1 (9 points)

This subtask has an additional constraint:

- $N \le 10$

Subtask 2 (24 points)

This subtask has an additional constraint:

- $R \le 50$

Subtask 3 (10 points)

This subtask has an additional constraint:

- $y = x + 1$ for all edges in the tree.

Subtask 4 (57 points)

This subtask has no additional constraints.

Examples

standard input	standard output
3 4 3 1 2 1 1 1 1 1 2 2 3 3 4 4 3 1 2 1 1 1 1 1 2 1 3 1 4 4 3 0 0 1 1 1 1 1 2 1 3 1 4	0111 0011 0000