

Problem A. actGenshinImp

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes

In the recent few years, this game has been so popular worldwide, it has even become a meme in the competitive programming community. Why would it be a bad idea to set problems about it?

You are given a grid G of lowercase latin alphabets. A simple path on this grid is defined as a sequence of $k \geq 1$ **distinct** cells $p_1, p_2, \dots, p_k$, such that p_{i-1} and p_i are adjacent either vertically or horizontally. Also, for some simple path d of m cells, let $f(d)$ be the string of length m such that $(f(d))_i$ is the letter written on the cell d_i of the grid G .

Please find the number of simple paths a of 13 cells, such that $f(a)$ is a cyclic shift of "genshinimpact". As the answer may be very large, you are only required to find the value modulo 998 244 353.

Input

The first line contains two integers r and c — the number of rows and the number of columns of G . ($1 \leq r, c \leq 500$)

Each of the r following lines contains a string of length c consisting of lowercase latin letters. The i -th of them is the i -th row of the grid G .

Output

Output the answer modulo 998 244 353 on one line.

Example

standard input	standard output
3 7 gshimct eninpag ppmpact	8

Note

The grid in the sample input contains 8 simple paths satisfying the condition. The 8 simple paths are as follows.

g s h i m c t e n i n p a g p p m p a c t	g s h i m c t e n i n p a g p p m p a c t
g s h i m c t e n i n p a g p p m p a c t	g s h i m c t e n i n p a g p p m p a c t
g s h i m c t e n i n p a g p p m p a c t	g s h i m c t e n i n p a g p p m p a c t
g s h i m c t e n i n p a g p p m p a c t	g s h i m c t e n i n p a g p p m p a c t

Problem B. Bracket Problem Yet Again

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 1024 megabytes

A balanced bracket string is a string consisting of ‘(’ and ‘)’, which can make a valid mathematical expression by inserting 0 or + in the string 0 or more times.

Bob had learned how to find the balanced bracket string with minimum cost, when the cost is defined as follows.

- If position i contains ‘(’, the cost of the position is a_i ;
- If position i contains ‘)’, the cost of the position is b_i ;
- The cost of a balanced bracket string is the sum of costs of all positions.

While rethinking about the solution, Bob became curious about the following problem.

- Assuming I can make a_i and b_i both 0 for at most k different indices, what is the new minimum cost?

As he was unable to solve the modified problem, he asked you to solve it. As Bob is a very selfish person, he wants you to solve it for each possible value of k . The indices chosen for $k = x$ need not be a strict subset of the indices chosen for $k = x + 1$. In other words, the problem must be solved independently for all values of k .

Input

The first line contains an **even** integer n , the length of the bracket sequence. ($2 \leq n \leq 2 \cdot 10^5$)

The second line contains n integers $a_1, a_2, \dots, a_n$, the costs of using ‘(’ on each index. ($0 \leq a_i \leq 10^9$)

The third line contains n integers $b_1, b_2, \dots, b_n$, the costs of using ‘)’ on each index. ($0 \leq b_i \leq 10^9$)

Output

Output $n + 1$ integers $c_0, c_1, \dots, c_n$ separated by spaces. c_x is defined by the minimum cost of the problem when $k = x$.

Example

standard input	standard output
4 0 5 0 5 3 3 3 3	6 3 0 0 0

Problem C. Candidate Elimination

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

James has been learning how to solve sudoku puzzles recently. He had recently learned about a concept called “Naked Pairs”. A pair of cells P is called a **Naked Pair** when it satisfies the following conditions.

- The two cells lie in the same group. In sudoku, a group is a set of n cells that must contain all integers from 1 to n .
- Both cells contain a subset of the same set of candidates S , where $|S| = 2$.

Given a naked pair, one can remove the candidates in S from all other cells in the group, since you know that in a valid sudoku each candidate in S must appear in one of the two cells.

While learning more about it, James realized that the concept of a “Naked Subset” can generalize the naked pairs. A set of cells Q with k cells is called a **Naked Subset** when it satisfies the following conditions.

- The k cells lie in the same group.
- All cells in Q contain a subset of the same set of candidates S' , where $|S'| = k$.

Given a naked subset, one can remove the candidates in S' from all other cells in the group, since you know that in a valid sudoku each candidate in S' must appear in one of the k cells.

James has been whining about how hard it is to find naked subsets in very big sudoku puzzles. Given a group, your job is to find every candidate that can be eliminated using **exactly one** naked subset.

Input

The first line consists of one integer n , the size of one group. ($1 \leq n \leq 10^5$)

The following n lines represent cells that belong to the group. The i -th line consists of one integer S_i , followed by S_i distinct integers. The S_i integers represent the candidates possible on the i -th cell.

It is guaranteed that the sum of S_i is at most $5 \cdot 10^5$.

Additionally, it is guaranteed that there is a valid assignment to all cells in the group. In other words, there exists at least one way to choose one candidate per cell, such that all integers chosen are distinct.

Output

Output n lines. The i -th line must consist of the candidates that can be removed from the i -th cell in **increasing order**, with the format being same as the input format.

Example

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

Note

In the first sample, the cells in the group belong to the box with green digits. The removals are done using the following naked subsets.

- Cells 1, 2, 3 have candidates $\{2, 4, 5\}$, so you can remove $\{4, 5\}$ from cell 4, $\{5\}$ from cell 5 and $\{2\}$ from cell 9.
- Cells 1, 2, 3, 5 have candidates $\{2, 4, 5, 7\}$, so you can remove $\{7\}$ from cells 4 and 9.
- Cells 1, 2, 3, 4, 5 have candidates $\{2, 4, 5, 7, 8\}$, so you can remove $\{8\}$ from cell 9.

2 9	4	5	7	1	2 3	8	2 6	2 3 6
1 2 7 8	3	1 2 7 8	6	9	2 5	4 5 7	1 2 5	1 2 4 5 7
1 2 7	1 2 7	1 2 6	8	5 3	4	5 6 7	9	1 2 3 5 6 7
2 4 5	2 5	2 4	1	5 3 6	8	4 5 6	7	9
4 5 7 8	5	3	2	5 6 7	9	1	5 6 8	4 5 6 8
6	9	1 2 7 8	4	5 3 7	5	3 5	2 5 8	2 3 5 8
1 2 7	8	1 2 7	3	4	7 6	5 6 7 9	1 5 6	1 5 6
5 3 7	5 6 7	6	9	2	1	5 6 7	4	5 6 7 8
1 4 7	1 7	6 9	5	8	7 6	2	3	1 6

The sudoku board corresponding to the sample. Generated using HoDoKu, an open source sudoku solver.

Problem D. Depth of Cartesian Tree

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 1024 megabytes

The **cartesian tree** of a sequence of distinct integers, is an unique binary tree defined as follows.

- The cartesian tree of a sequence with one element is the tree of one node;
- The root of the cartesian tree of a sequence corresponds to the **maximum value** of the sequence;
- If the root does not correspond to the leftmost index, the left subtree is the cartesian tree of the subarray left to it;
- If the root does not correspond to the rightmost index, the right subtree is the cartesian tree of the subarray right to it.

In a binary tree, the depth of a node is defined as the distance from the root to the node.

You are given a permutation p of elements $1, 2, \dots, n$. You must solve q queries of the following kind.

- $l\ r$: Assume that you construct a cartesian tree of the subsequence $p_l, p_{l+1}, \dots, p_r$. Answer the sum of the depths of all nodes in the tree.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 10^6$) — the length of the permutation and number of queries respectively.

The second line contains n distinct integers $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) — the permutation p .

The next q lines contain two integers l_i, r_i ($1 \leq l_i \leq r_i \leq n$) — the bounds of query i .

Output

For each query, output the answer on a new line.

Example

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

Problem E. Extra Character

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

Given a string s of length n , the **Z-function** of s is a sequence of n integers such that the i -th element is equal to the largest nonnegative integer x no greater than n such that $s_j = s_{i+j-1}$ for all $1 \leq j \leq x$. For example, the i -th element is 0 if $s_i \neq s_1$, and the first element is always n .

Your friend recently sent you a string s of length n . He needs to compute the Z-function of this string but does not have enough time to compute it himself. You quickly compute it for him and send it back to your friend.

However, it seems like your friend made a mistake; he made a typo, and it turned out that the string he sent you had an extra character in the front. Unfortunately, neither you nor your friend kept a copy of the original string. Still, you want to know the Z-function of the intended string.

Please find the Z-function of the intended string $t = s_{2\dots n}$. For each position, output -1 instead if the value on the specific position cannot be uniquely determined.

Input

The first line contains a single integer n , the length of the original string. ($2 \leq n \leq 10^6$)

The second line contains n integers $p_1, p_2, \dots, p_n$, the Z-function of the original string. ($0 \leq p_i \leq n$)

It is guaranteed that a string corresponding to the given Z-function exists.

Output

Output $n - 1$ integers $q_1, q_2, \dots, q_{n-1}$, the Z-function of the intended string $s_{2\dots n}$.

If the value cannot be uniquely determined for some position, output -1 for the position.

Examples

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

Problem F. Five Steiner

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 512 megabytes

After reading the paper *Steiner minimal trees on sets of four points*, one has decided to make things a bit harder. Why not five points instead?

You are given 5 points with integer coordinates on the plane. The distance between two points is defined by the Euclidean distance.

Please find the sum of distances of the minimum weight[†] steiner tree[‡] of the given points.

[†] The weight of a tree is defined by the sum of distances between all pairs of points connected by edges.

[‡] A steiner tree is defined as a tree spanning the given points, possibly containing some arbitrary number of points added to the point set. The added points need not have integer coordinates.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10$). The description of the test cases follows.

Each test case consists of five lines, containing the following input.

1. Two integers x_a and y_a , the coordinates of the first point A . ($-10^5 \leq x_a, y_a \leq 10^5$)
2. Two integers x_b and y_b , the coordinates of the second point B . ($-10^5 \leq x_b, y_b \leq 10^5$)
3. Two integers x_c and y_c , the coordinates of the third point C . ($-10^5 \leq x_c, y_c \leq 10^5$)
4. Two integers x_d and y_d , the coordinates of the fourth point D . ($-10^5 \leq x_d, y_d \leq 10^5$)
5. Two integers x_e and y_e , the coordinates of the fifth point E . ($-10^5 \leq x_e, y_e \leq 10^5$)

Output

For each test case, print the sum of distances of the minimum steiner tree of the given points on one line.

Your output will be accepted if the absolute or relative error does not exceed 10^{-5} .

Example

standard input	standard output
2	7.464101
-2 -1	11.332503
-1 1	
0 -1	
1 1	
2 -1	
0 2	
3 1	
-3 1	
-1 -2	
1 -2	

Problem G. Get Mex Range Add Linear

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

Sorry, we have had the theme of adding up integers so many times elsewhere. So we give you a different definition of addition — adding an element to a set.

You are given a , a sequence of n sets. Initially, $a_i = \{0\}$ (set only containing 0) for all $1 \leq i \leq n$.

You are asked to solve q queries of the following kinds.

- 1 l r : Set $a_i \leftarrow a_i \cup \{i - l + 1\}$ for all $l \leq i \leq r$. ($1 \leq l \leq r \leq n$)
- 2 i : Output the value of $\text{mex}(a_i)^\dagger$. ($1 \leq i \leq n$)

† Given a set of nonnegative integers S , $\text{mex}(S)$ is defined as the smallest nonnegative integer **not** in S .

Input

The first line contains two integers n and q — the number of sets and queries. ($1 \leq n, q \leq 5 \cdot 10^5$)

Each of the q following lines contains a query. Each query is given in the format described above.

Output

For each query of type 2, print the answer on a new line.

Example

standard input	standard output
5 9	2
1 1 5	1
2 1	3
2 5	4
1 2 3	2
1 4 5	
1 3 3	
2 2	
2 3	
2 4	

Note

The sample input is explained as follows.

After the first query of type 1, a changes to $[\{0, 1\}, \{0, 2\}, \{0, 3\}, \{0, 4\}, \{0, 5\}]$.

Then, the mex of $\{0, 1\}$ and $\{0, 5\}$ are 2 and 1 correspondingly.

After three more queries of type 1, a changes to $[\{0, 1\}, \{0, 1, 2\}, \{0, 1, 2, 3\}, \{0, 1, 4\}, \{0, 2, 5\}]$.

Then, the mex of $\{0, 1, 2\}$, $\{0, 1, 2, 3\}$, $\{0, 1, 4\}$ are 3, 4, 2 correspondingly.

Problem H. Hell of Optimizing Geometric Construction

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

One day, dnialh mentioned that optimizing geometric construction perfectly is not possible. Oh, very well. You will see about that.

You are given a positive integer n such that $2 \leq n \leq 1\,320$. Please find a sequence of n points on the plane, $X_1, X_2, \dots, X_n$, satisfying the following constraints.

- The coordinates of each point are integers in the range $[-1\,000, 1\,000]$.
- No two points share the same coordinates.
- For each $1 \leq i \leq n$, the closest point to X_i other than X_i itself is unique. Let the index of this unique point be $f(i)$.
- Let p be a sequence of n integers defined by $p_1 = 1$ and $p_{i+1} = f(p_i)$ for $1 \leq i < n$. Then, p is a permutation of $1, 2, \dots, n$.

It is proven that such a sequence of points exists under the constraints of this task.

Input

A positive integer n is given on one line. ($2 \leq n \leq 1\,320$)

Output

Output n lines. The i -th line must contain x_i and y_i , the coordinates of X_i , separated by a space. ($-1\,000 \leq x_i, y_i \leq 1\,000$)

Example

standard input	standard output
4	-2 -2 1 2 -2 2 2 1

Note

In the samples, $n = 4$ and $X = [(-2, -2), (1, 2), (-2, 2), (2, 1)]$.

Here, $f(i)$ is determined as follows.

- Other than X_1 , the unique closest point to X_1 is X_3 . Therefore, $f(1) = 3$.
- Other than X_2 , the unique closest point to X_2 is X_4 . Therefore, $f(2) = 4$.
- Other than X_3 , the unique closest point to X_3 is X_2 . Therefore, $f(3) = 2$.
- Other than X_4 , the unique closest point to X_4 is X_2 . Therefore, $f(4) = 2$.

Now, one can manually verify that the resultant sequence $p = [1, 3, 2, 4]$ is a permutation of $1, 2, 3, 4$. Therefore, the sequence of points satisfies the constraints.

Problem I. Inequality Satisfying Subsequences

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 1024 megabytes

A sequence of three positive integers $[a, b, c]$ is called a **triangle** if $a + b > c$ holds when reordered such that $a \leq b \leq c$.

A sequence of positive integers is **triangle-free** if no subsequence of length 3 forms a triangle.

Given a sequence of positive integers L , count how many non-empty subsequences of L are triangle-free. As the answer may be very large, you are only required to find the value modulo 998 244 353.

Input

The first line contains a single integer n ($1 \leq n \leq 7\,000$) — the size of the input sequence.

The second line contains n integers $L_1, L_2, \dots, L_n$ ($1 \leq L_i \leq 10^9$) — the elements of the sequence L .

Output

Output the number of non-empty triangle-free subsequences modulo 998 244 353 on one line.

Examples

standard input	standard output
10 1 2 3 4 5 6 7 8 9 10	173
6 9 6 7 8 4 4	23
10 1 3 9 27 81 243 729 2187 6561 19683	1023

Problem J. Jenga Tower

Input file: `standard input`
Output file: `standard output`
Time limit: 3 seconds
Memory limit: 1024 megabytes

The city of Ruralia is building a tower consisting of n Jenga blocks.

Each block is rectangular with height 1, width $r_i - l_i$, and density 1 and will be placed on the plane, axis-aligned with its bottom left corner lying on the coordinates $(l_i, n - i)$.

After all the blocks have been placed, all of the scaffolding will be removed, and the tower will stay standing if for each block i the center of mass of all blocks above i is between l_i and r_i .

More formally, a tower is **stable** if and only for all $2 \leq i \leq n$ the following inequality holds.

$$l_i \leq \frac{\sum_{j=1}^{i-1} (r_j - l_j) \cdot \frac{l_j + r_j}{2}}{\sum_{j=1}^{i-1} (r_j - l_j)} \leq r_i$$

Unfortunately, you have received news that one of the Jenga blocks has been missing but you are unsure which one. When the shipment arrives, the builders of Ruralia will place the blocks as originally planned, except the one missing block will not be placed and all blocks above it will be moved downwards by 1.

However, before building the tower, the builders want to know if the resulting tower will be **stable**. Thus, they have asked you to determine for each block i whether the structure with block i missing is stable.

Do note that it is not necessarily the case that the original tower plan was **stable**.

Input

The first line contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of blocks in the original tower.

The following n lines each contain two integers l_i, r_i ($1 \leq l_i < r_i \leq 10^6$) — the endpoints of block i .

It is not necessary that the tower in the input is **stable**.

Output

Output n lines.

On line i , output "YES" or "NO" — corresponding to whether the tower will be stable if block i is removed.

Examples

standard input	standard output
5	NO
0 10	YES
4 10	NO
0 6	YES
4 10	YES
0 6	
4	YES
6 8	YES
0 3	NO
0 8	NO
0 6	

Problem K. Kaz's Party

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 megabytes

Kaz is inviting n of his friends over for a party! He has currently prepared the cocktails for them (each friend has ordered a different drink), with friend i having a preference for cocktail i .

However, Kaz is looking to pull a prank. When handing out drinks if he hands out a drink to the wrong person, he knows the following round of an 'exchange process' will occur: all partygoers who do not have their desired drinks will leave their drinks on a table. All such drinks will be shuffled and handed back in a completely random order to those still in the exchange process. Any participants who receive their desired drink will leave to enjoy the party, while those still with the wrong drink will shuffle again.

To maximize the humor of his prank, Kaz wants the exchange process to take the maximum expected number of rounds. Help Kaz find an assignment of drinks that will maximize this value.

Input

The input consists of a single integer n ($1 \leq n \leq 1\,000$) — the number of friends invited to the party.

Output

On the first line, output a single real number R — the maximum expected number of rounds.

On the second line, output n distinct integers $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) — the assignment of the drinks. Here, drink p_i will be handed to friend i .

Your answer will be accepted if the output satisfies the following conditions.

- The value of R differs from the actual answer by at most 10^{-9} absolute or relative error.
- The value of R differs from the expected number of rounds given the permutation p by at most 10^{-9} absolute or relative error.

Example

standard input	standard output
2	2.0 2 1

Problem L. LIS on Tree

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

You are given a tree of n nodes. Each node has a non-negative integer value v_i .

Let a **tree subsequence** be a sequence of nodes $S = s_1, s_2, \dots, s_k$ such that there exists vertices u, v in the tree such that S is a subsequence of the unique shortest path starting at u and ending at v .

A **tree subsequence** is increasing if for all $1 \leq i \leq k - 1$ we have that $v_{s_i} < v_{s_{i+1}}$ (Note that this corresponds to a **strictly** increasing sequence).

Find the length of the longest increasing **tree subsequence**.

Input

The first line of input contains a single integer n ($1 \leq n \leq 3 \cdot 10^5$) — the number of nodes in the tree.

The second line of input contains n integers $v_1, v_2, \dots, v_n$ ($1 \leq v \leq 10^9$) — the value of each node in the tree.

The following $n - 1$ lines each contain two integers a_i, b_i ($1 \leq a_i, b_i \leq n$) — the endpoints of edge i .

It is guaranteed that the given edges form a tree.

Output

Output a single integer — the length of the longest increasing tree subsequence.

Examples

standard input	standard output
4 7 7 7 7 2 4 2 3 1 2	1
5 3 9 14 7 12 1 4 3 4 4 5 2 3	3
12 10 3 8 13 6 2 3 14 1 5 10 6 1 10 2 6 2 10 7 9 2 9 9 11 3 7 2 8 5 7 4 7 2 12	4