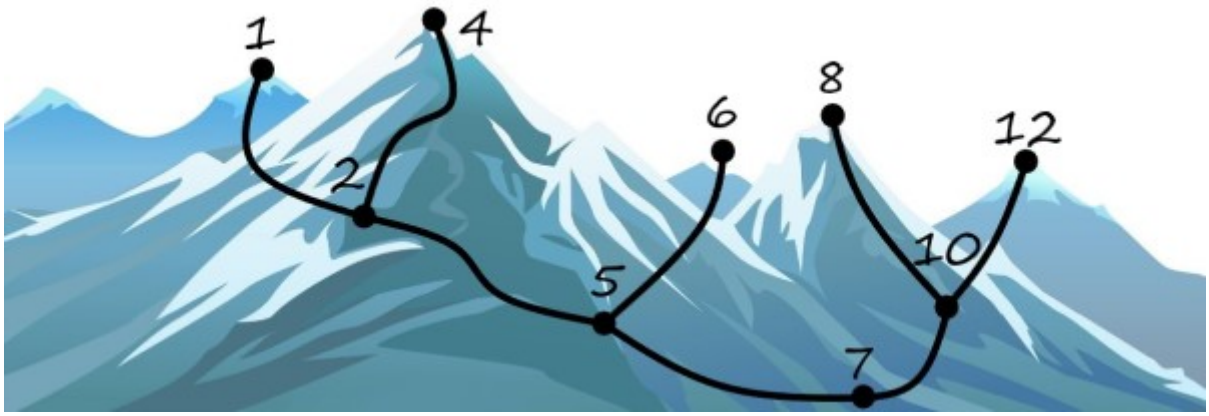
Problem A. 111284. Mountains

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are going to hike in the mountains. You have written directions (left or right) on a piece of paper, in which direction you need to turn on each of the branches of the path, following which you can reach the peak. In the mountains there are several peaks and you have recorded the path to each peak. Since you are a guide, you need to check the day before the hike which of the recorded paths is available. All paths start at one point at the foothills.

The path is presented in the form of "$RRLLRLR$", which means to reach the peak you need to turn right at the beginning, then right again, then left, left, right, left, right. Peak located after last turn and it is possible that it does not exist. You are given all the paths in the mountains that are available in the form of a Binary Search Tree and the paths to the peaks that are written on a piece of paper. You need to tell which of the paths written on the piece of paper is available.



Input

Given the number N, M $(1 \leq N, M \leq 10^5)$-the number of branches in the available path in the mountains and the number of paths written on a piece of paper. In the next line N integers describe the BST. The following M lines describe the paths $p_1, p_2, p_3 ... p_m (2 \leq |p_i| \leq 100)$ written on a piece of paper.

Output

You need to print M lines where i^{th} line should be "YES" if the path p_i to the peak is available, otherwise "NO".

Example

standard input	standard output
9 4 7 10 12 8 5 6 2 1 4 LLL LRR RL RR	YES NO YES YES

Problem B. 105587. Get subtree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given Binary Search Tree. Your task is to calculate the size of the subtree of the node X.

Remember, that the subtree of node X is the set of all nodes whose ancestor is node X, including it. The size of the subtree is the size of such set.

Input

The first line of the input contains an integer N - number of nodes in Binary Search Tree ($1 \leqslant N \leqslant 10^3$).

The second line contains N integers a_i - values of nodes in order of insertion to the Binary Search Tree.

The third line contains single integer - value of the node which subtree's size you must calculate.

Output

Print the size of the subtree of the given node.

Example

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

Problem C. 111632. Christmas Gifts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Christmas is coming! Everyone is preparing gifts for their families. The Damir's family is also preparing for this. The Damir's family has many children, so the parents decided to buy n various gifts. Parents decided to number these gifts - i-th as a_i. They hang them on Christmas tree in socks, following the form binary search tree. In particular, they insert i-th gift with the value a_i following the rules of binary search tree.

As you know, Damir is the smallest among the whole family. Therefore, parents for the holiday allowed him to pick up his gift first. Damir knew that his gift has number k, but he mistakenly assumed that all the gifts below his gift were also intended for him. Now, parents are confused and want to find out what gifts Damir wants to grab for himself.

Input

The first line contains one integers n ($1 \leqslant \mathrm{n} \leqslant 10^3$) - the number of gifts. The second line contains n integers a_i. ($1 \leqslant a_i \leqslant 10^3$) The third line contains one integer k ($1 \leqslant \mathrm{k} \leqslant 10^3$). It is guaranteed that the array has the number k.

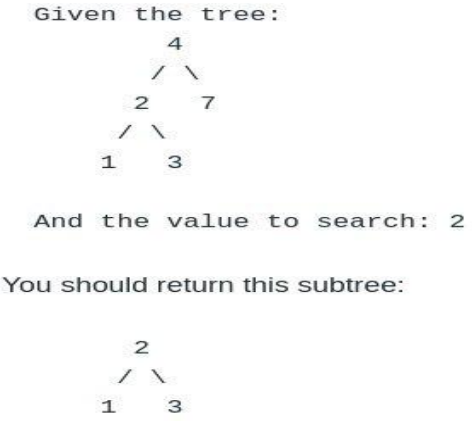
Output

You should output this subtree in pre-order (root, left subtree, right subtree).

Example

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

Note



Problem D. 105816. Aureole

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a permutation of size n. Create an empty BST, and insert into BST values $p_1, p_2, ..., p_n$ in this order. You need to find how many levels are there and sum the of values for each level.

Input

In the first line there is a single integer $1 \leq n \leq 5000$ size of permutation. Second line contains n distinct numbers from 1 to n - the permutation.

Output

In first line output k - maximum level in bst. In second line output k integers - sum of values for each level.

Examples

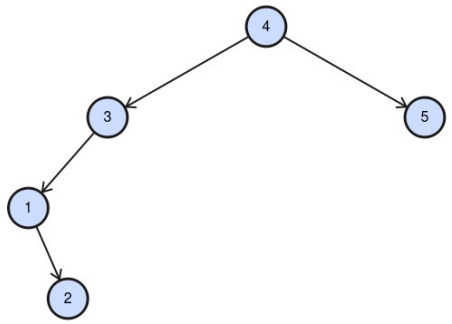
standard input	standard output
1 1	1 1
5 4 3 5 1 2	4 4 8 1 2

Note

Level of vertex is defined as:

Level of root is 0, and level of each non-root vertex is (level of it's parent) + 1.

In second testcase, BST looks like this.



There are 4 levels, and sum for each level is 4, 3 + 5, 1, 2.

Problem E. 52477. Width

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Given a binary tree, write a program to get the width of the given tree.

The level of a node is the number of vertices on the path from this node to the root. The width of a level h is the number of vertices with level h. The width of a tree is the maximal width over the levels.

Vertex number 1 always will be root.

Input

Given integer n — ($1 \leq |n| \leq 10^3$), number of vertexes. The next $n-1$ lines has 3 numbers x, y, z - description of binary tree, meaning that vertex y son of vertex x, if z = 0, it is left son, if z = 1 it is right son.

Output

Print one integer maximum width.

Examples

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

Note

sample 1:

Explanation: The maximum width existing in the third level with the length 3 (5,6,4).

sample 2:
Explanation: The maximum width existing in the third level with the length 2 (3,4).

Problem F. 106735. Triangle Binary Search Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given N integers, that form a binary search tree by inserting them in the given order. You draw a set of horizontal lines that goes through nodes with the same height (level, depth). After that you can see triangles with nodes instead vertices and edges instead sides. Your task is to calculate the number of the smallest triangles.

Input

The first line consists of an integer N - number of nodes in Binary Search Tree ($1 \leqslant N \leqslant 10000$).

The second line contains N integers a_i - value of each node in Binary Search Tree in order of their insertion ($1 \leqslant a_i \leqslant N$).

It is guaranteed that there are no duplicates.

Output

Print the number of mini-triangles in resulting Binary Search Tree.

Examples

standard input	standard output
3 3 5 1	1
3 1 3 5	0
16 13 9 3 7 6 16 1 11 12 10 4 2 14 5 8 15	5

Problem G. 197831. Killua and Hunter exam

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

While Gon is surviving on the Greed Island, Killua, after the first unsuccessful attempt to pass the hunter exam, decides to test himself again. This time one of his tasks is to find the maximum distance between any two vertices in a binary search tree. Since Killua is pretty bad at algorithms, he asks for your help.

Input

In the first line you will be given single number N ($1 \leq N \leq 200000$). Next line consists of N numbers, where a_i ($1 \leq a_i \leq 10^9$) represents the i-th number inserted to a binary search tree. If a_i was found in a tree, then you don't have to insert it again.

Subtasks

1. (30%) $N \leq 100$.
2. (30%) $N \leq 1000$
3. (40%) No additional constraints.

Output

Print one single number - the maximum distance between any two vertices in a binary tree.

Examples

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

Note

In the first test, the answer is the distance between nodes 1 and 8.

Problem H. 111743. Greater Sum Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given the root of a **binary search** tree with distinct keys. Replace the key of each node with the sum of the keys over the nodes that has greater than or equal key. Print new keys in increasing order.

As a reminder, a binary search tree is a tree that satisfies these constraints:

- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- Both the left and right subtrees must also be binary search trees.

Input

The number of nodes n in the tree is between 1 and 100. Each node will have value between 0 and 1000. The given tree is a binary search tree.

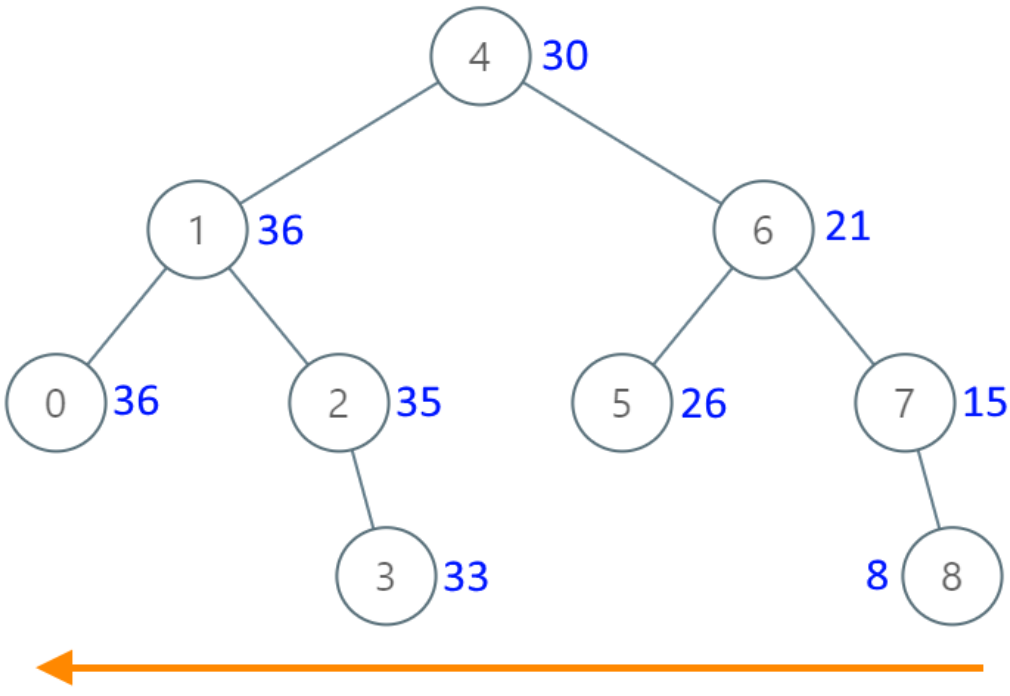
Output

In a single line print the answer.

Example

standard input	standard output
9 4 1 6 0 2 3 5 7 8	8 15 21 26 30 33 35 36 36

Note



NOTE: Solve with **BST**!

Problem I. 105712. BST with multiplicities

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given implementation of Binary Search Tree. You need to introduce new feature to the Binary Search Tree. It must be able to contain not only nodes with unique keys, but the number of copies of that key as well.

You are given several queries, each of them is of one the type:

- '*insert* X' - insert X into the tree, if it already in a tree - increase its multiplicity,
- '*delete* X' - decrease multiplicity of X by one, if it became zero - delete the node from the tree,
- '*cnt* X' - output the multiplicity of X.

To complete the task you need to download solution code from https://bit.ly/3kWufJa and make some extra changes in it. Remaining code was written for you.

Input

The first line contains single integer Q - number of queries ($1 \leqslant Q \leqslant 10^3$). Each of the next Q lines contains one query.

Output

Print answer on each query of type '*cnt* X' in separate line.

Examples

standard input	standard output
4 insert 1 cnt 1 insert 1 cnt 1	1 2
8 insert 1 cnt 1 insert 1 cnt 1 delete 1 cnt 1 delete 1 cnt 1	1 2 1 0

Problem J. 106664. Balanced Binary Search Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You have an array with 2^N - 1 elements in it. You want to build the Binary Search Tree on this array, adding elements in order of their appearance in array (from left to right). But there is a probability that the tree would be imbalanced. That's why you decided to shuffle your array to obtain **perfectly balanced** Binary Search Tree (i.e. BST with prefect binary tree underneath) after adding elements (from left to right, again). Your task is to print your array after appropriate shuffle. If there are several possible shuffles, print the array after applying any of them.

Note, that you are not asked for building Binary Search Tree, but only for shuffling array.

Input

The first line of input consists of single integer N that describes the length of the array ($1 \leqslant N \leqslant 15$).

The next line contains 2^N - 1 integers a_i - elements of the array ($0 \leqslant a_i \leqslant 2 \cdot 10^9$).

It is guaranteed that there are no duplicates in the array.

Output

Print 2^N - 1 integers - elements in your array after applying required shuffle.

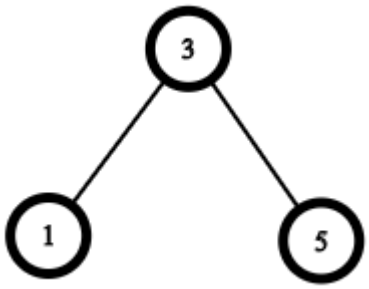
Examples

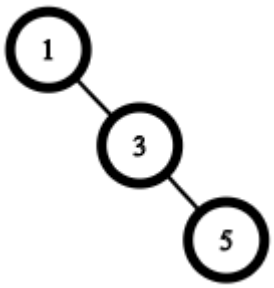
standard input	standard output
2 3 5 1	3 5 1
2 1 3 5	3 1 5

Note

In the first sample given array can be used for building balanced BST (left picture).

In the second sample given array gives such chain tree (right picture), so it must be shuffled.





Note, that for both samples [3, 5, 1] and [3, 1, 5] are correct answers.

Hint: Use divide and conquer method (recall advanced sorting algorithms) and implement recursive function to solve this problem