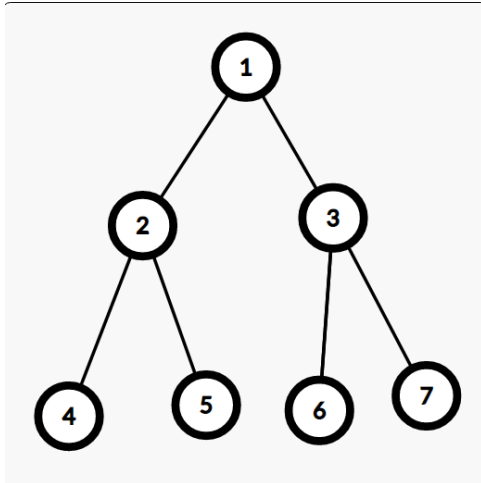


[Binary Tree]. Bài 1. Duyệt Inorder (Left, Root, Right)

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo thứ tự Inorder. Thứ tự duyệt Inorder của cây nhị phân bên dưới xuất hiện trong test case mẫu.



Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra thứ tự duyệt cây Inorder

Sample Input 0

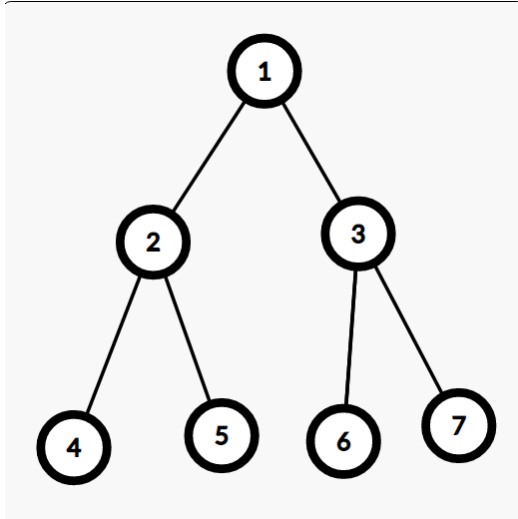
```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

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

[Binary Tree]. Bài 2. Duyệt Preorder (Root, left, right)

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo thứ tự Preorder. Thứ tự duyệt Preorder của cây nhị phân bên dưới xuất hiện trong test case mẫu.



Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra thứ tự duyệt cây Preorder

Sample Input 0

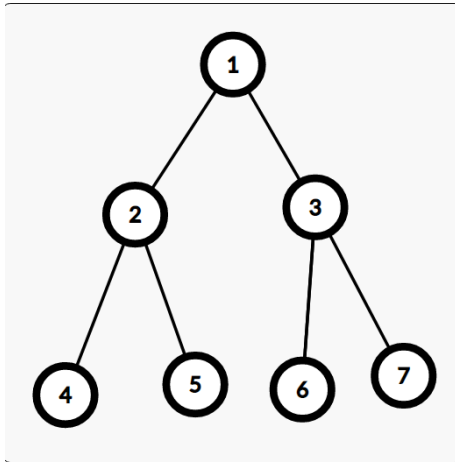
```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

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

[Binary Tree]. Bài 3. Duyệt Postorder (Left, right, root)

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo thứ tự Postorder. Thứ tự duyệt Postorder của cây nhị phân bên dưới xuất hiện trong test case mẫu.



Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra thứ tự duyệt cây Postorder

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

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

[Binary Tree]. Bài 4. Đếm node lá

Cho cây nhị phân, nhiệm vụ của bạn là đếm số lượng node trên cây là node lá

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra số lượng node lá trên cây

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
4
```

[Binary Tree]. Bài 5. Độ cao của cây

Cho cây nhị phân, nhiệm vụ của bạn là tìm độ cao lớn nhất của 1 node lá trên cây.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra độ cao của cây

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
2
```

Sample Input 1

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

Sample Output 1

```
3
```

[Binary Tree]. Bài 6. Cây nhị phân đầy đủ (full binary tree)

Cây nhị phân được gọi là đầy đủ nếu mọi node trung gian của nó đều có 2 con trái và phải. Nhiệm vụ của bạn là kiểm tra xem cây nhị phân đã cho có phải là cây nhị phân đầy đủ hay không, có in ra YES, ngược lại in ra NO.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x .

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra đáp án của bài toán

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
YES
```

Sample Input 1

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

Sample Output 1

```
NO
```

[Binary Tree]. Bài 7. Kiểm tra node lá cùng mức

Cho một cây nhị phân, nhiệm vụ của bạn là kiểm tra xem mọi node lá của cây có cùng mức hay không.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In YES nếu mọi node lá trên cây có cùng mức, ngược lại in NO.

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
YES
```

Sample Input 1

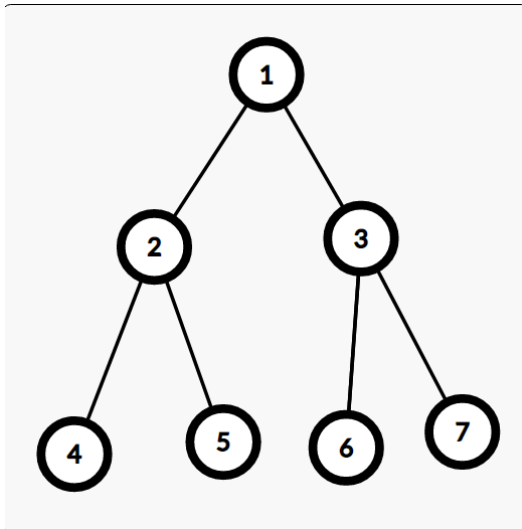
```
7
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R 6 8 L
```

Sample Output 1

```
NO
```

[Binary Tree]. Bài 8. Duyệt theo mức

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo mức. Ví dụ cây nhị phân dưới đây sẽ có thứ tự duyệt theo mức như sau : 1 2 3 4 5 6 7



Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra thứ tự duyệt theo mức

Sample Input 0

```

6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
  
```

Sample Output 0

```

1 2 3 4 5 6 7
  
```

Sample Input 1

```

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

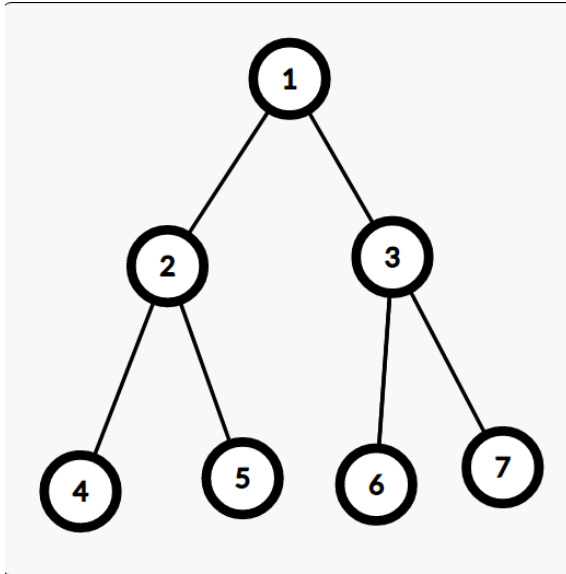
Sample Output 1

```

1 2 3 4 5 6 7 8
  
```

[Binary Tree]. Bài 9. Duyệt xoắn ốc

Cho cây nhị phân, nhiệm vụ của bạn in ra thứ tự duyệt cây kiểu xoắn ốc. Ví dụ với cây nhị phân bên dưới thì thứ tự duyệt xoắn ốc sẽ là : 1 2 3 7 6 5 4



Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra thứ tự duyệt theo kiểu xoắn ốc.

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

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

[Binary Tree]. Bài 10. Cây nhị phân hoàn hảo.

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân đã cho có phải là cây nhị phân hoàn hảo hay không. Cây nhị phân hoàn hảo nếu mọi node lá của nó có cùng mức và mỗi node trung gian đều có 2 con.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10^5$;

Output Format

In ra YES hoặc NO tùy theo kết quả của bài toán

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
YES
```

Sample Input 1

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

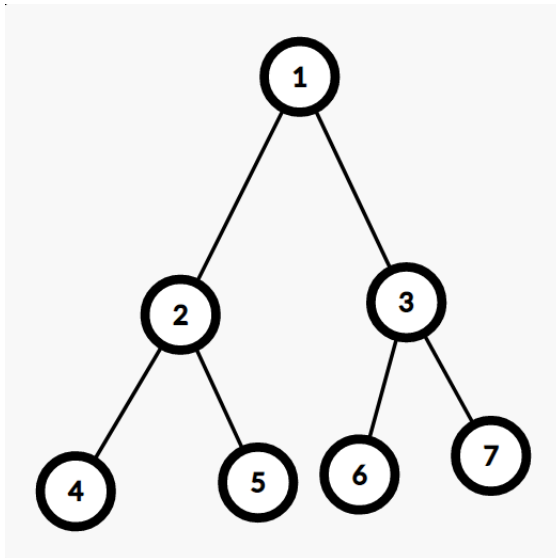
Sample Output 1

```
NO
```

[Binary Tree]. Bài 11. Duyệt cây

Cho phép duyệt cây theo thứ tự giữa (Inorder) và duyệt trước (Preorder), bạn hãy in ra cây dưới dạng duyệt sau (Postorder).

Thứ tự duyệt Preorder : 1 2 4 5 3 6 7. Thứ tự duyệt Inorder : 4 2 5 1 6 3 7.
Thứ tự duyệt Postorder : 4 5 2 6 7 3 1.



Input Format

Dòng đầu tiên là N : số lượng node của cây. Dòng thứ 2 là thứ tự duyệt trước của cây. Dòng thứ 3 là thứ tự duyệt giữa của cây.

Constraints

$1 \leq N \leq 100$;

Output Format

In ra thứ tự duyệt sau của cây.

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 12. Cây nhị phân bằng nhau

Cho 2 cây nhị phân, nhiệm vụ của bạn là kiểm tra xem 2 cây nhị phân đã cho có giống nhau hoàn toàn hay không ?

Input Format

Dòng đầu tiên là số N1 : Số lượng cạnh của cây nhị phân thứ 1. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con

bên trái hoặc node con bên phải của x. Dòng tiếp theo là số N2 : Số lượng cạnh của cây nhị phân thứ 2. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N1, N2 \leq 100$;

Output Format

In ra YES hoặc NO.

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
6
1 2 L 1 3 R 2 4 L 2 5 R 3 7 R 3 6 L
```

Sample Output 0

```
YES
```

[Binary Tree]. Bài 13. Cây nhị phân DFS

Cho cây nhị phân hãy đưa ra thứ tự duyệt cây theo thuật toán DFS, khi mở rộng 2 node con bên trái hoặc bên phải, ưu tiên mở rộng node con bên phải trước.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$;

Output Format

In ra thứ tự duyệt cây theo thuật toán DFS

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

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

[Binary Tree]. Bài 14. Chiều cao của node

Cho cây nhị phân, nhiệm vụ của bạn là xác định chiều cao của từng node trên cây, node gốc được quy định có chiều cao là 0, chiều cao của các node khác được tính bằng số cạnh trên đường đi ngắn nhất từ nó tới gốc.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10000$;

Output Format

In ra chiều cao từng node trên cây theo thứ tự node tăng dần.

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
0 1 1 2 2 2 2
```

[Binary Tree]. Bài 15. Node lá có mức cao nhất

Cho cây nhị phân, nhiệm vụ của bạn là đếm xem trên cây có bao nhiêu node lá có chiều cao lớn nhất, node gốc được quy định có chiều cao là 0,

chiều cao của các node khác được tính bằng số cạnh trên đường đi ngắn nhất từ nó tới gốc.

Input Format

Dòng đầu tiên là N : số lượng cạnh của cây. Các dòng tiếp theo mô tả cây nhị phân bằng 3 số (x, y, z) : x là node cha, y là node con, z có thể nhận 1 trong 2 giá trị là L và R, tương ứng với y là node con bên trái hoặc node con bên phải của x.

Constraints

$1 \leq N \leq 1000$; $1 \leq x, y \leq 10000$;

Output Format

In ra số node lá có chiều cao lớn nhất

Sample Input 0

```
6
1 2 L 1 3 R 2 4 L 2 5 R 3 6 L 3 7 R
```

Sample Output 0

```
4
```

[Binary Tree]. Bài 16. BST 1

Cho mảng số nguyên A[] có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là A[0]. In ra thứ tự duyệt cây Inorder.

Input Format

Dòng đầu tiên là N : số lượng phần tử trong mảng A[]; Dòng thứ 2 gồm N phần tử trong mảng A[];

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra thứ tự duyệt in order của cây nhị phân tìm kiếm tạo được.

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 17. BST 2.

Cho mảng số nguyên $A[]$ có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là $A[0]$. In ra thứ tự duyệt cây Preoder.

Input Format

Dòng đầu tiên là N : số lượng phần tử trong mảng $A[]$; Dòng thứ 2 gồm N phần tử trong mảng $A[]$;

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra thứ tự duyệt in Preorder của cây nhị phân tìm kiếm tạo được.

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 18. BST 3

Cho mảng số nguyên $A[]$ có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là $A[0]$. In ra thứ tự duyệt cây Postorder.

Input Format

Dòng đầu tiên là N : số lượng phần tử trong mảng A[]; Dòng thứ 2 gồm N phần tử trong mảng A[];

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra thứ tự duyệt in Postorder của cây nhị phân tìm kiếm tạo được.

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 19. Max min BST

Cho mảng số nguyên A[] có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là A[0]. Tìm phần tử có giá trị lớn nhất và nhỏ nhất trên cây nhị phân tìm kiếm.

Input Format

Dòng đầu tiên là N : số lượng phần tử trong mảng A[]; Dòng thứ 2 gồm N phần tử trong mảng A[];

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra giá trị lớn nhất, nhỏ nhất của cây nhị phân tìm kiếm

Sample Input 0

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

Sample Output 0

[Binary Tree]. Bài 20. Search

Cho mảng số nguyên $A[]$ có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là $A[0]$. Nhiệm vụ của bạn là tìm kiếm sự xuất hiện của phần tử có giá trị X trên cây nhị phân tìm kiếm.

Input Format

Dòng đầu tiên là N và X : số lượng phần tử trong mảng $A[]$ và phần tử cần tìm kiếm; Dòng thứ 2 gồm N phần tử trong mảng $A[]$;

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i]$, $X \leq 10^6$;

Output Format

In YES nếu X xuất hiện trên cây ngược lại in ra NO

Sample Input 0

```
8
8 7 5 6 2 4 1 3
13
```

Sample Output 0

```
NO
```

Sample Input 1

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

Sample Output 1

```
YES
```

[Binary Tree]. Bài 21. Delete

Cho mảng số nguyên $A[]$ có N phần tử, lần lượt thêm các phần tử trong mảng vào cây nhị phân sao cho cây nhị phân thu được là cây nhị phân tìm kiếm với gốc cây là $A[0]$. Nhiệm vụ của bạn là xóa node có giá trị X trên cây nhị phân tìm kiếm nếu tồn tại node có giá trị X .

Input Format

Dòng đầu tiên là N và X : số lượng phần tử trong mảng $A[]$ và phần tử cần xóa; Dòng thứ 2 gồm N phần tử trong mảng $A[]$;

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i]$, $X \leq 10^6$;

Output Format

In ra thứ tự duyệt cây Inorder sau khi xóa

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 22. Preorder to postorder

Cho thứ tự duyệt Preorder của cây nhị phân tìm kiếm, hãy in ra thứ tự duyệt postorder.

Input Format

Dòng đầu tiên là N : số lượng node trên cây; Dòng thứ 2 là các node theo thứ tự duyệt preorder.

Constraints

$1 \leq N \leq 1000$; Các node có giá trị dương $\leq 10^6$;

Output Format

In ra thứ tự duyệt postorder.

Sample Input 0

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

Sample Output 0

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

[Binary Tree]. Bài 23. Check BST

Cho thứ tự duyệt Inorder của một cây nhị phân, hãy xác định xem thứ tự này có phải là thứ tự duyệt inorder của một cây nhị phân tìm kiếm hay không.

Input Format

Dòng đầu tiên là N : số lượng node trên cây; Dòng thứ 2 gồm N node theo thứ tự duyệt inorder

Constraints

$1 \leq N \leq 1000$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra YES nếu cây nhị phân đã cho là cây nhị phân tìm kiếm, ngược lại in ra NO.

Sample Input 0

```
8
1 3 5 7 9 11 14 16
```

Sample Output 0

```
YES
```