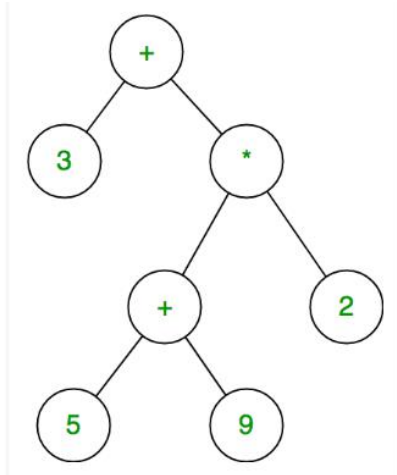


# CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

## CONTEST 11 – CÂY NHỊ PHÂN

### BÀI 1. CÂY BIỂU THỨC 1

Cây biểu thức là một cây nhị phân trong đó mỗi node trung gian là một phép toán, mỗi node lá là một toán hạng. Ví dụ với biểu thức  $P = 3 + ((5+9)*2)$  sẽ được biểu diễn như cây dưới đây. Đối với cây biểu thức, duyệt theo thứ tự trước ta sẽ được biểu thức trung tố, duyệt theo thứ tự sau ta sẽ được biểu thức hậu tố, duyệt theo thứ tự giữa ta được biểu thức tiền tố. Chú ý, cây biểu thức luôn là cây nhị phân đầy (mỗi node trung gian đều có hai node con).



Cho biểu thức hậu tố P, hãy sử dụng cây biểu thức để đưa ra biểu thức trung tố tương ứng với biểu thức P.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test là một biểu thức hậu tố P.
- T, P thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq \text{length}(P) \leq 100$ .

#### Output:

- Đưa ra biểu thức trung tố tương ứng với P.

#### Ví dụ:

Input	Output
2	a + b - e * f * g
ab+ef*g*-	w * l - r + b
wlrb+-*	

### BÀI 2. CÂY BIỂU THỨC 2

Cho một cây biểu thức là một cây nhị phân đầy đủ bao gồm các phép toán +, -, \*, / và một số toán hạng có giá trị nguyên. Nhiệm vụ của bạn là hãy tính toán giá trị biểu thức được biểu diễn

trên cây nhị phân đầy đủ. Ví dụ với cây dưới đây là biểu diễn của biểu thức  $P = ((5*4) + (100-20))$  sẽ cho ta giá trị là 100.

**Input:**

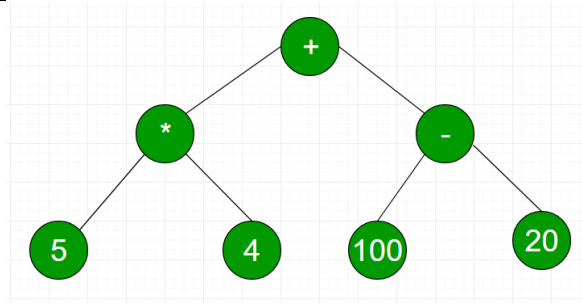
- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test là gồm hai dòng: dòng thứ nhất đưa vào N là số lượng node của cây; dòng thứ hai đưa vào nội dung các node của cây; các node được viết cách nhau một vài khoảng trống. Các số có giá trị nguyên không vượt quá 1000.
- T, N, P thỏa mãn ràng buộc :  $1 \leq T \leq 100$ ;  $1 \leq N$ ,  $\text{lenght}(P) \leq 100$ .

**Output:**

- Đưa ra giá trị của cây biểu thức.

**Ví dụ:**

Input	Output
2	100
7	-3
+ * - 5 4 100 20	
3	
- 4 7	



**BÀI 3. DUYỆT CÂY 1**

Cho phép duyệt cây nhị phân Inorder và Preorder, hãy đưa ra kết quả phép duyệt Postorder của cây nhị phân. Ví dụ với cây nhị phân có các phép duyệt cây nhị phân của cây dưới đây:

Inorder : 4 2 5 1 3 6

Preorder: : 1 2 4 5 3 6

Postorder : 4 5 2 6 3 1

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng node; dòng tiếp theo đưa vào N số theo phép duyệt Inorder; dòng cuối cùng đưa vào N số là kết quả của phép duyệt Preorder; các số được viết cách nhau một vài khoảng trống.
- T, N, node thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 1000$ ;  $1 \leq \text{giá trị node} \leq 10^4$ ;

**Output:**

- Đưa ra kết quả phép duyệt Postorder theo từng dòng.

**Ví dụ:**

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

#### **BÀI 4. DUYỆT CÂY 2**

Cho mảng A[] gồm N node là biểu diễn phép duyệt theo thứ tự giữa (Preorder) của cây nhị phân tìm kiếm. Nhiệm vụ của bạn là đưa ra phép duyệt theo thứ tự sau của cây nhị phân tìm kiếm.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng node; dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.
- T, N, node thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq A[i] \leq 10^4$ ;

**Output:**

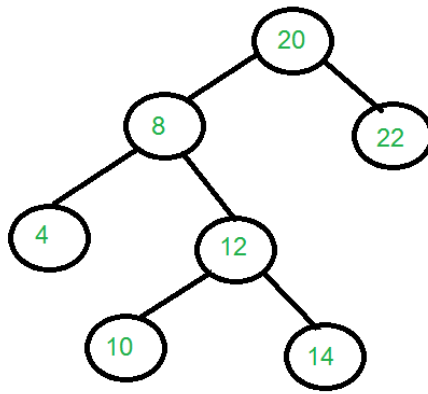
- Đưa ra kết quả phép duyệt Postorder theo từng dòng.

**Ví dụ:**

Input	Output
2	35 30 100 80 40
5	35 32 30 120 100 90 80 40
40 30 35 80 100	
8	
40 30 32 35 80 90 100 120	

#### **BÀI 5. DUYỆT CÂY 3**

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo **Level**-order. Phép duyệt level-order trên cây là phép thăm node theo từng mức của cây. Ví dụ với cây dưới đây sẽ cho ta kết quả của phép duyệt level-order: 20 8 22 4 12 10 14.



### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

### Output:

- Đưa ra kết quả phép duyệt level-order theo từng dòng.

### Ví dụ:

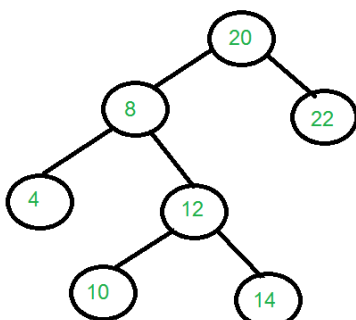
Input	Output
2	1 3 2
2	10 0 30 40 60
1 2 R 1 3 L	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

## BÀI 6. DUYỆT CÂY 4

Cho hai mảng là phép duyệt Inorder và Level-order, nhiệm vụ của bạn là xây dựng cây nhị phân và đưa ra kết quả phép duyệt Postorder. Level-order là phép duyệt theo từng mức của cây. Ví dụ như cây dưới đây ta có phép Inorder và Level-order như dưới đây:

Inorder : 4 8 10 12 14 20 22

Level order: 20 8 22 4 12 10 14



**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng node; dòng tiếp theo đưa vào N số là phép duyệt Inorder; dòng cuối cùng đưa vào N số là phép duyệt Level-order; các số được viết cách nhau một vài khoảng trống.
- T, N, node thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq A[i] \leq 10^4$ ;

**Output:**

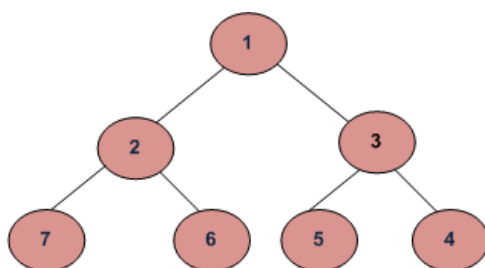
- Đưa ra kết quả phép duyệt Postorder theo từng dòng.

**Ví dụ:**

Input	Output
2	1 2 0
3	3 4 1 5 6 2 0
1 0 2	
0 1 2	
7	
3 1 4 0 5 2 6	
0 1 2 3 4 5 6	

**BÀI 7. DUYỆT CÂY 5**

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo xoắn ốc (spiral-order). Phép. Ví dụ với cây dưới đây sẽ cho ta kết quả của phép duyệt spiral-order: 1 2 3 4 5 6 7.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x=R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

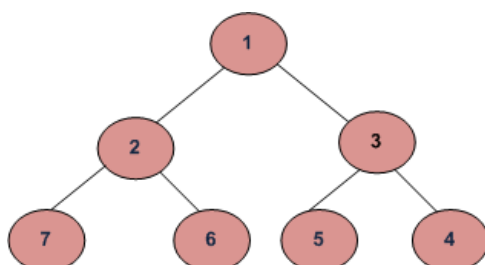
- Đưa ra kết quả phép duyệt level-order theo từng dòng.

**Ví dụ:**

Input	Output
2	1 3 2
2	10 0 30 60 40
1 2 R 1 3 L	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

## BÀI 8. DUYỆT CÂY 6

Cho cây nhị phân, nhiệm vụ của bạn là duyệt cây theo mức đảo ngược (reverse-level-order). Với cây dưới đây sẽ cho ta kết quả của phép duyệt theo mức đảo ngược là : 7 6 5 4 3 2 1.



**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

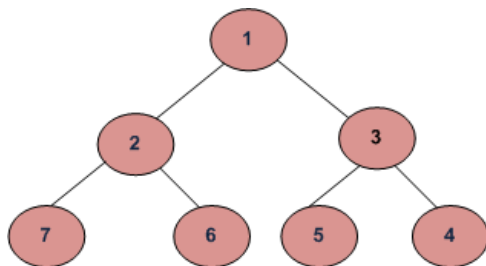
- Đưa ra kết quả phép duyệt reverse-level-order theo từng dòng.

**Ví dụ:**

Input	Output
2	3 2 1
2	40 20 30 10
1 2 R 1 3 L	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

### BÀI 9. KIỂM TRA NODE LÁ

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem tất cả các node lá của cây có cùng một mức hay không? Ví dụ với cây dưới đây sẽ cho ta kết quả là Yes.



#### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x=R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

#### Output:

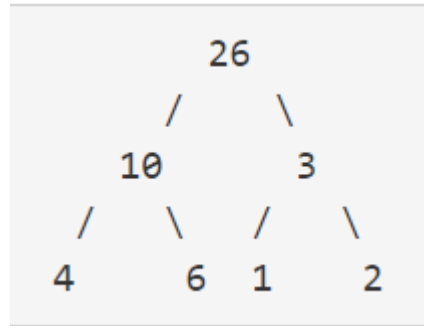
- Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input	Output
2	1
2	0
1 2 R 1 3 L	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

### BÀI 10. CÂY NHỊ PHÂN TỔNG

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có phải là một cây tổng hay không? Một cây nhị phân được gọi là cây tổng nếu tổng các node con của node trung gian bằng giá trị node cha. Node không có node con trái hoặc phải được hiểu là có giá trị 0. Ví dụ dưới đây là một cây tổng



**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

Input	Output
2	1
2	0
3 1 L 3 2 R	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

## BÀI 11. CÂY NHỊ PHÂN HOÀN HẢO

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có phải là một cây hoàn hảo hay không (perfect tree)? Một cây nhị phân được gọi là cây hoàn hảo nếu tất cả các node trung gian của nó đều có hai node con và tất cả các node lá đều có cùng một mức.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**



Input	Output
3	Yes
6	Yes
10 20 L 10 30 R 20 40 L 20 50 R 30 60 L 30 70 R	No
2	
18 15 L 18 30 R	
5	
1 2 L 2 4 R 1 3 R 3 5 L 3 6 R	

## BÀI 12. CÂY NHỊ PHÂN ĐỦ

Cho cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có phải là một cây đủ hay không (full binary tree)? Một cây nhị phân được gọi là cây đủ nếu tất cả các node trung gian của nó đều có hai node con.

### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ:

Input	Output
2	1
4	0
1 2 L 1 3 R 2 4 L 2 5 R	
3	
1 2 L 1 3 R 2 4 L	

## BÀI 13. CÂY NHỊ PHÂN BẰNG NHAU

Cho hai cây nhị phân, nhiệm vụ của bạn là kiểm tra xem cây nhị phân có giống nhau hay không?

### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó

u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái của mỗi cây; u, v, x được viết cách nhau một vài khoảng trống.

- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

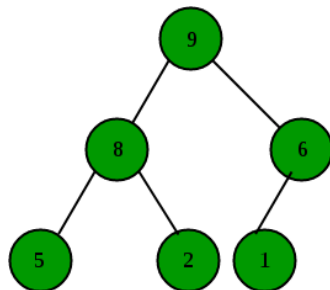
- Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

Input	Output
2	1
2	0
1 2 L 1 3 R	
2	
1 2 L 1 3 R	
2	
1 2 L 1 3 R	
2	
1 3 L 1 2 R	

#### **BÀI 14. TỔNG NODE LÁ BÊN TRÁI** chú ý hàm Init()

Cho cây nhị phân, nhiệm vụ của bạn là tính tổng của tất cả các node lá bên trái trên cây? Ví dụ với cây dưới đây ta có kết quả là 6.



**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

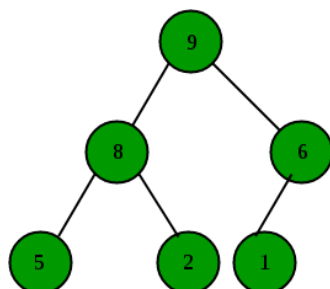
- Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

Input	Output
2	2
2	130
1 2 L 1 3 R	
5	
10 20 L 10 30 R 20 40 L 20 60 R 30 90 L	

### BÀI 15. TỔNG NODE LÁ BÊN PHẢI

Cho cây nhị phân, nhiệm vụ của bạn là tính tổng của tất cả các node lá bên phải trên cây? Ví dụ với cây dưới đây ta có kết quả là 2.



#### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x= R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

#### Output:

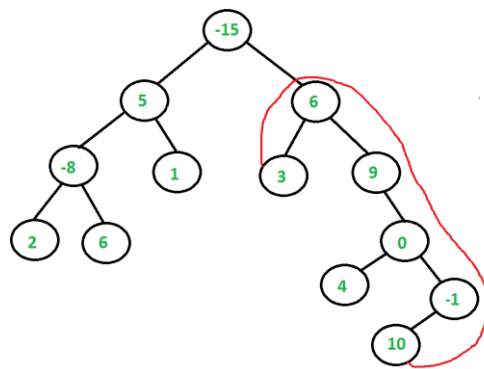
- Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input	Output
2	3
2	60
1 2 L 1 3 R	
5	
10 20 L 10 30 R 20 40 L 20 60 R 30 90 L	

### BÀI 16. TỔNG LỚN NHẤT

Cho cây nhị phân có giá trị mỗi node là một số, nhiệm vụ của bạn là tìm tổng lớn nhất từ một node lá này sang một node lá khác? Ví dụ với cây dưới đây ta có tổng lớn nhất là 27.



### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x=R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.
- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

### Output:

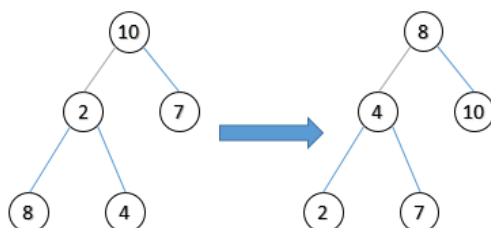
- Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ:

Input	Output
1 12 -15 5 L -15 6 R 5 -8 L 5 1 R -8 2 L -8 -3 R 6 3 L 6 9 R 9 0 R 0 4 L 0 -1 R -1 10 L	27

## BÀI 17. BIẾN ĐỔI SANG CÂY NHỊ PHÂN TÌM KIẾM

Cho cây nhị phân, nhiệm vụ của bạn là dịch chuyển cây nhị phân thành cây nhị phân tìm kiếm. Phép dịch chuyển phải bảo toàn được cấu trúc cây nhị phân ban đầu. Ví dụ dưới đây sẽ minh họa phép dịch chuyển:



### Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 3 dòng: dòng đầu tiên đưa vào số N là số lượng cạnh của cây; dòng tiếp theo đưa vào N bộ ba (u, v, x), trong đó u là node cha, v là node con, x=R nếu v là con phải, x=L nếu v là con trái; u, v, x được viết cách nhau một vài khoảng trống.

- T, N, u, v, thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq u, v \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng là phép duyệt Inorder của cây tìm kiếm.

**Ví dụ:**

Input	Output
2	1 2 3
2	10 20 30 40 60
1 2 R 1 3 L	
4	
10 20 L 10 30 R 20 40 L 20 60 R	

### **BÀI 18. XÂY DỰNG LẠI CÂY NHỊ PHÂN TÌM KIẾM**

Cho một mảng là phép duyệt level-order của cây nhị phân tìm kiếm. Nhiệm vụ của bạn là xây dựng lại cây nhị phân tìm kiếm bảo toàn được cấu trúc cây nhị phân ban đầu.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm dòng: dòng đầu tiên đưa vào số N là số lượng node của cây tìm kiếm; dòng tiếp theo đưa vào phép duyệt level-order của cây tìm kiếm; các số được viết cách nhau một vài khoảng trống.
- T, N, node thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq \text{node} \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng là phép duyệt Inorder của cây tìm kiếm.

**Ví dụ:**

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

### **BÀI 19. DUYỆT CÂY NHỊ PHÂN TÌM KIẾM 1**

Cho một mảng A[] gồm N phần tử biểu diễn phép duyệt preorder của cây nhị phân tìm kiếm. Nhiệm vụ của bạn là đưa ra phép duyệt postorder của cây nhị phân tìm kiếm.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.

- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng node của cây tìm kiếm; dòng tiếp theo đưa vào phép duyệt preorder của cây tìm kiếm; các số được viết cách nhau một vài khoảng trống.
- T, N, A[i] thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq A[i] \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng là phép duyệt postorder của cây tìm kiếm.

**Ví dụ:**

Input	Output
2	35 30 100 80 40
5	35 32 30 120 100 90 80 40
40 30 35 80 100	
8	
40 30 32 35 80 90 100 120	

**BÀI 20. DUYỆT CÂY NHỊ PHÂN TÌM KIẾM 2**

Cho một mảng A[] gồm N phần tử. Nhiệm vụ của bạn là đưa ra 1 nếu mảng A[] biểu diễn phép duyệt inorder của cây nhị phân tìm kiếm, ngược lại đưa ra 0.

**Input:**

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm 2 dòng: dòng đầu tiên đưa vào số N là số lượng node của cây tìm kiếm; dòng tiếp theo đưa vào N số A[i]; các số được viết cách nhau một vài khoảng trống.
- T, N, A[i] thỏa mãn ràng buộc:  $1 \leq T \leq 100$ ;  $1 \leq N \leq 10^3$ ;  $1 \leq A[i] \leq 10^4$ ;

**Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

Input	Output
3	1
5	0
10 20 30 40 50	0
6	
90 80 100 70 40 30	
3	
1 1 2	

## BÀI 21. NODE LÁ CỦA CÂY NHỊ PHÂN TÌM KIẾM

Cho dãy số gồm N số là phép duyệt theo thứ tự trước (Preorder) của một cây nhị phân tìm kiếm. Hãy in ra tất cả các node lá của cây ?

Ví dụ với dãy  $A[] = \{30, 20, 15, 25, 23, 28, 40, 35, 33, 38, 45\}$  là phép duyệt cây theo thứ tự trước sẽ cho ta kết quả: 15, 23, 28, 33, 38, 45.

### Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ( $T \leq 100$ ).
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên N ( $N \leq 10^6$ ). Dòng tiếp theo là N số là phép duyệt theo thứ tự trước của cây BST.

### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ:

Input	Output
2	1 7 50
6	15 3 28 33 38 45
10 5 1 7 40 50	
11	
30 20 15 25 23 28 40 35 33 38 45	

## BÀI 22. NODE TRUNG GIAN CỦA CÂY NHỊ PHÂN TÌM KIẾM

Cho dãy số gồm N số là phép duyệt theo thứ tự trước (Preorder) của một cây nhị phân tìm kiếm. Hãy đưa ra số các node trung gian của cây ?

Ví dụ với dãy  $A[] = \{30, 20, 15, 25, 23, 28, 40, 35, 33, 38, 45\}$  là phép duyệt cây theo thứ tự trước sẽ cho ta kết quả là 5 bao gồm các node: 30, 20, 25, 40, 35.

### Input:

- Dòng đầu tiên đưa vào số lượng bộ test T ( $T \leq 100$ ).
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên N ( $N \leq 106$ ). Dòng tiếp theo là N số là phép duyệt theo thứ tự trước của cây BST.

### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

### Ví dụ:

Input:	Output
2	3
6	5
10 5 1 7 40 50	
11	
30 20 15 25 23 28 40 35 33 38 45	

### BÀI 23. ĐỘ SÂU CÂY NHỊ PHÂN TÌM KIẾM

Cho dãy số gồm N số là phép duyệt theo thứ tự trước (Preorder) của một cây nhị phân tìm kiếm. Hãy tìm độ sâu của cây ?

Ví dụ với dãy  $A[] = \{30, 20, 15, 25, 23, 28, 40, 35, 33, 38, 45\}$  là phép duyệt cây theo thứ tự trước sẽ cho ta kết quả là 3.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T (T \leq 100)$ .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên N ( $N \leq 106$ ). Dòng tiếp theo là N số là phép duyệt theo thứ tự trước của cây BST.

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input:	Output
2	2
6	3
10 5 1 7 40 50	
11	
30 20 15 25 23 28 40 35 33 38 45	

### BÀI 24. CÂY NHỊ PHÂN TÌM KIẾM CÂN BẰNG 1

Hãy xây dựng một cây nhị phân tìm kiếm cân bằng từ dãy số  $A[] = \{a_0, a_1, \dots, a_{n-1}\}$ . Đưa ra nội dung node gốc của cây tìm kiếm cân bằng. Ví dụ với dãy  $A[] = \{40, 28, 45, 38, 33, 15, 25, 20, 23, 35, 30\}$  ta sẽ có cây nhị phân tìm kiếm cân bằng với node gốc là 33.

#### Input:

- Dòng đầu tiên đưa vào số lượng bộ test  $T (T \leq 100)$ .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên N ( $N \leq 10^6$ ). Dòng tiếp theo là N số của mảng  $A[]$ .

#### Output:

- Đưa ra kết quả mỗi test theo từng dòng.

#### Ví dụ:

Input	Output
2	30
11	5
40 28 45 38 33 15 25 20 23 35 30	
10	
1 2 3 4 5 6 7 8 9 10	



## **BÀI 25. CÂY NHỊ PHÂN TÌM KIẾM CÂN BẰNG 2**

Hãy xây dựng một cây nhị phân tìm kiếm cân bằng từ dãy số  $A[] = (a_0, a_1, \dots, a_{n-1})$ . Đưa ra phép duyệt theo thứ tự trước (preorder) của cây tìm kiếm cân bằng. Ví dụ với dãy  $A[] = \{40, 28, 45, 38, 33, 15, 25, 20, 23, 35, 30\}$  ta sẽ có phép duyệt theo thứ tự trước của cây nhị phân tìm kiếm cân bằng với node gốc là 33 : 33, 25, 20, 15, 23, 28, 30, 40, 38, 35, 45.

### **Input:**

- Dòng đầu tiên đưa vào số lượng bộ test  $T$  ( $T \leq 100$ ).
- Những dòng kế tiếp đưa vào  $T$  bộ test. Mỗi bộ test gồm 2 dòng: dòng thứ nhất là số tự nhiên  $N$  ( $N \leq 10^6$ ). Dòng tiếp theo là  $N$  số của mảng  $A[]$ .

### **Output:**

- Đưa ra kết quả mỗi test theo từng dòng.

### **Ví dụ:**

Input	Output
2	30 23 15 20 25 28 38 33 35 40 45
11	5 2 1 3 4 8 6 7 9 10
40 28 45 38 33 15 25 20 23 35 30	
10	
1 2 3 4 5 6 7 8 9 10	