

## 〈자료구조 실기 기말〉 학번/이름:

### ※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 ↳ 이 후는 각 입력과 출력에 대한 설명이다.

### [ 문제 1 ] 트리 만들기 (노드는 최대 100개)

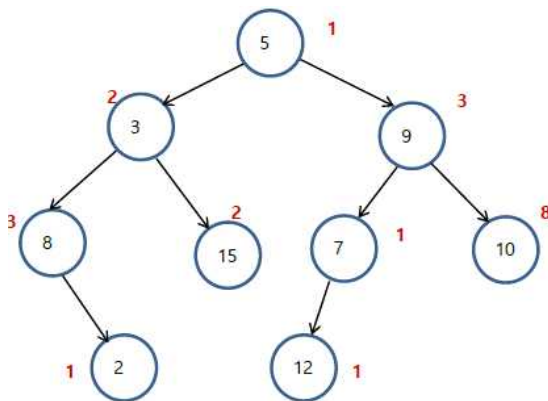
- 트리는 연결이진트리로 구현 (또는 링크 표현법 사용)하고, 각 노드에 저장되는 정보는 아래와 같다.

왼쪽 자식 링크	노드 번호	weight(값)	오른쪽 자식 링크
----------	-------	-----------	-----------

- 오직 루트(root) 노드에 대해서만 허용. 즉, 트리는 루트 노드를 통해서만 접근 가능

### 입력 상세:

- 트리 정보
  - 첫 째 줄에 노드의 개수 n이 주어진다.
  - n개의 노드 번호, 왼쪽 자식 노드번호, 오른쪽 자식 노드번호, 노드의 값
  - 노드의 값(weight)은 양의 정수
  - 노드의 입력순서는 트리의 위치와 상관없다.



노드 옆의 빨간 숫자가 노드의 값이다.

예)

```

3 8 15 2 → 3의 왼쪽 자식은 8, 오른쪽 자식은 15, 값 2
8 0 2 3 → 8의 왼쪽 자식은 없고, 오른쪽 자식은 2, 값 3
5 3 9 1 → 5의 왼쪽 자식은 3, 오른쪽 자식은 9, 값 1
2 0 0 1
15 0 0 2
9 7 10 3
    
```

7 12 0 1  
12 0 0 1  
10 0 0 8

**출력 상세:** 루트로 부터 모든 노드를 inorder 로 순회하여 각 노드의 번호를 출력한다.  
(임의의 트리를 만들지 못하면 위의 그림의 트리를 만들어서 출력한다.)

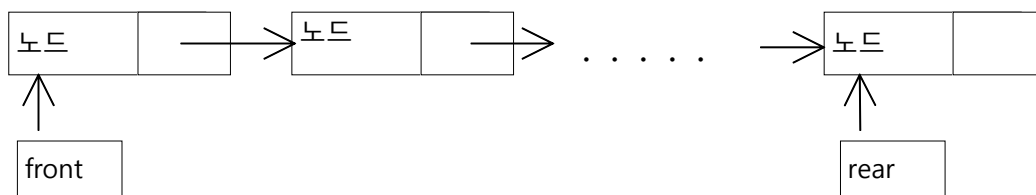
입력 예시 1

출력 예시 1

<p>9            ↳ 노드 개수</p> <p>3 8 15 2</p> <p>8 0 2 3</p> <p>2 0 0 1</p> <p>15 0 0 2</p> <p>9 7 10 3</p> <p>7 12 0 1</p> <p>12 0 0 0</p> <p>10 0 0 8</p> <p>5 3 9 1</p>	<p>8 2 3 15 5 12 7 9 10□</p>
--	------------------------------

**[문제 2]** 문제1와 같은 트리 모양에서, 루트로부터 레벨별 순회를 하여 노드의 번호를 차례로 출력하는 프로그램을 작성하여라. 반드시 큐를 사용해야 한다.

헤드노드나 테일 노드가 없는 연결 리스트를 이용하여 큐를 구현하고, 필요한 메소드를 모두 구현해야한다. (enqueue, dequeue, isempty, initQueue 등)



**주의1)** 문제1의 트리 만들기를 하지 못했으면, 자신만의 방법으로 위 모양의 트리를 구축하고, 이를 바탕으로 문제2의 알고리즘을 구현한다.

**주의2)** 연결리스트로 구성된 큐의 모양은 위와 같다.

입력 예시 1

9                    ↳ 노드 개수  
5 3 9 1  
3 8 15 2  
8 0 2 3  
2 0 0 1  
15 0 0 2  
9 7 10 3  
7 12 0 1  
12 0 0 0  
10 0 0 8

출력 예시 1

5 3 9 8 15 7 10 2 12□

**[문제 3]** 트리에서 루트로부터 모든 외부노드까지의 path 는 루트노드로 부터 그 노드까지 도달하는 경로에 있는 모든 노드를 포함한다. path sum은 그 경로에 속한 모든 노드들의 weight의 합이라고 하자. 모든 path sum들 중 가장 큰 것을 구하고, 이 때 거치는 노드들의 번호를 차례로 출력하여라. (외부노드 -> 루트 방향)

위의 트리에서 보면

- 5 - 3 - 8 - 2 경로로 순회하면 노드값의 합은 7
- 5 - 3 - 15 경로로 순회하면 노드값의 합은 5
- 5 - 9 - 7 - 12 경로로 순회하면 노드값의 합은 3
- 5 - 9 - 10 경로로 순회하면 노드값의 합은 12

그러므로 maximum path sum은 12 가 된다.

단계1) maximum path sum값을 구한다.

입력 예시 1

9                    ↳ 노드 개수  
3 8 15 2  
8 0 2 3  
2 0 0 1  
15 0 0 2  
9 7 10 3  
7 12 0 1  
12 0 0 1  
10 0 0 8  
5 3 9 1

출력 예시 1

12                    ↳ maximum path sum

단계2) maximum path sum값을 가지는 path에 속하는 모든 노드 번호를 출력한다.

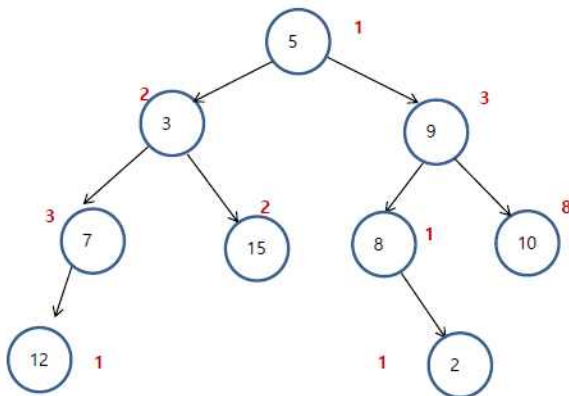
(루트로부터 먼 노드번호 부터 루트노드 번호까지.).

입력 예시 1

출력 예시 1

9 3 8 15 2 8 0 2 3 2 0 0 1 15 0 0 2 9 7 10 3 7 12 0 1 12 0 0 1 10 0 0 8 5 3 9 1	↳ 노드 개수	12 10 9 5□	↳ maximum path sum ↳ 5 - 9 - 10 경로
--	---------	---------------	---------------------------------------

단계3) 2개의 노드 번호를 입력받아, 이 노드 하위트리들을 위치를 서로 교환하는 프로그램을 작성하여라. 교환이 완료되면 inorder로 노드번호를 출력한다. (루트노드만을 통하여 수정을 하여야 한다. 노드 배열이나 기타 다른 구조를 이용해서 교환하면 안됨.) 문제1의 트리가 아래처럼 변경)



입력 예시 1

출력 예시 1

9 3 8 15 2 8 0 2 3 2 0 0 1 15 0 0 2 9 7 10 3 7 12 0 1 12 0 0 1 10 0 0 8 5 3 9 1 7 8	↳ 노드 개수 ↳ 노드 번호	12 7 3 15 5 8 2 9 10□
---	--------------------	-----------------------