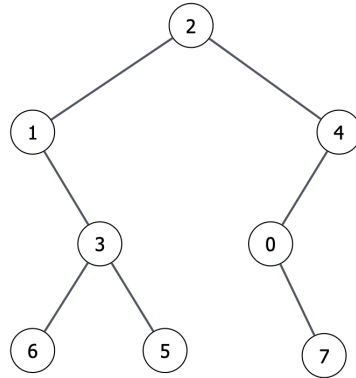


프로그래밍 과제 03

1. $N \leq 100$ 개의 노드를 가지는 하나의 이진 트리가 입력 파일 `input1.txt`로 주어진다. 파일의 첫 줄에는 노드의 개수 N 이 주어지고, 이어진 N 줄에는 각 줄마다 하나의 노드에 대해서 그 노드의 번호, 왼쪽자식 노드의 번호, 오른쪽 자식 노드의 번호를 나타내는 3개의 정수가 주어진다. 자식 노드가 없는 경우에는 -1로 표시된다. 노드가 주어지는 순서에는 아무런 규칙이 없다. 노드의 번호는 0에서 $N-1$ 까지의 정수이다. 예를 들어 아래의 왼쪽 이진트리는 오른쪽과 같은 파일로 표현된다.



input1.txt

```

8
1 -1 3
3 6 5
5 -1 -1
2 1 4
6 -1 -1
0 -1 7
7 -1 -1
4 0 -1
  
```

입력으로 주어진 이진트리를 선순위, 중순위, 후순위로 순회하면서 노드들의 번호를 출력하라.

입력 예(INPUT1.TXT)	출력
8 1 -1 3 3 6 5 5 -1 -1 2 1 4 6 -1 -1 0 -1 7 7 -1 -1 4 0 -1	2 1 3 6 5 4 0 7 1 6 3 5 2 0 7 4 6 5 3 1 7 0 4 2
6 3 5 4 5 -1 -1 4 -1 -1 0 -1 3 2 -1 1 1 0 -1	2 1 0 3 5 4 2 0 5 3 4 1 5 4 3 0 1 2
9 8 -1 -1 7 -1 -1 5 -1 -1 4 -1 7 3 8 -1 6 -1 -1 1 4 5 2 6 3 0 2 1	0 2 6 3 8 1 4 7 5 6 2 8 3 0 4 7 1 5 6 8 3 2 7 4 5 1 0

2. N 개의 서로 다른 정수들이 저장된 이진검색트리에 대한 후순위(postorder) 수열이 입력으로 주어진다. 이 이진검색트리의 높이(height)를 계산하여 출력하는 프로그램을 작성하라. 여기서 트리의 높이는 루트로 부터 가장 멀리 떨어진 리프(leaf) 노드까지의 거리에 1을 더한 값을 의미한다. 만약 주어진 수열이 이진검색트리의 후순위 수열이 아닌 경우에는 -1을 출력하라.

입출력 형식

입력은 키보드로 부터 받는다. 입력의 형식은 다음과 같다. 먼저 첫 줄에는 이진검색 트리의 노드 개수 N 이 주어지고 다음 줄에는 후순위(postorder) 수열에 해당하는 N 개의 정수가 주어진다. 출력은 화면으로 한다.

입력 예	출력 예
11 2 3 4 1 8 10 9 11 7 6	5
11 2 3 1 4 8 10 7 11 9 6	-1
5 2 1 4 3 5	4
7 1 2 3 4 5 6 7	7
16 1 3 5 4 2 7 8 9 11 10 12 14 16 15 13 6	7

3. N 개의 서로 다른 정수들이 저장된 이진검색트리에 대한 후순위(postorder) 수열이 입력으로 주어진다. 이 이진검색트리에 대한 선순위(preorder) 수열을 출력하는 프로그램을 작성하라.

입출력 형식

입력은 키보드로 부터 받는다. 입력의 형식은 다음과 같다. 먼저 첫 줄에는 이진검색 트리의 노드 개수 N 이 주어지고 다음 줄에는 후순위 수열에 해당하는 N 개의 정수가 주어진다. 출력은 화면으로 한다.

입력 예	출력 예
11 3 2 5 4 1 8 10 9 11 7 6	6 1 4 2 3 5 7 11 9 8 10
5 1 3 5 4 2	2 1 4 3 5
7 4 7 6 5 3 2 1	1 2 3 5 4 6 7
6 6 5 4 3 2 1	1 2 3 4 5 6
7 2 4 3 5 1 6 7	7 6 1 5 3 2 4

4. 주소록을 저장한 파일 `address_book.tsv`가 주어진다. 파일은 평범한 텍스트 파일이며, 파일의 한 라인은 한 사람에 대한 이름(name), 회사(company), 주소(address), 우편번호(zip), 전화번호(phone, email)의 6가지 필드로 구성되며, 파일의 첫 라인은 필드의 이름을 표시하므로 무시한다. 필드들은 하나의 탭 문자(\t)로 분리되어 있다. 이 주소록 데이터를 저장하기 위한 이진탐색트리를 구현하라. 즉, 한 사람에 대한 정보를 저장하는 구조체(struct) 혹은 클래스(class)를 정의하고, 이진탐색트리의 각 노드에 한 사람에 대한 데이터를 저장한다. 파이선을 사용할 경우 클래스를 정의하는 대신 리스트나 딕셔너리 등을 이용하여 한 사람에 대한 정보를 저장해도 무방하다. 사람들은 이름의 알파벳 순으로 저장한다. 먼저 데이터 파일을 읽어서 모든 사람들을 이진검색트리로 저장하고, 아래의 예시와 같이 삭제(delete)와 검색(find), 트레이스(trace), 그리고 이름의 알파벳 순으로 정렬하여 출력하는(list) 기능, 그리고 변경된 데이터를 파일로 저장하는(save) 기능 등을 제공해야 한다.

```
$ read address_book.tsv // 데이터 파일을 읽어서 이진검색트리로 저장한다.
$ list // 트리를 inorder로 순회하면서
// 모든 사람에 대한 모든 정보를 다음과 같은 형식으로 출력한다.

Abel
  Company: Rangoni Of Florence
  Address: 37275 St Rt 17m M Middle Island Suffolk NY
  Zipcode: 11953
```

```

Phones: 631-335-3414
Email: amaclead@gmail.com
Adelina
  Company: Courtyard By Marriott
  Address: 80 Pittsford Victor Rd #9 Cleveland Cuyahoga OH
  Zipcode: 44103
  Phones: 216-230-4892
  Email: adelina_nabours@gmail.com
...
$ find Annelle          // 이름으로 검색한다.
Annelle
  Company: Vico Products Mfg Co
  Address: 5 W 7th St Parkville Baltimore MD
  Zipcode: 21234
  Phones: 410-757-1035
  Email: annelle@yahoo.com
$ add Kim              // 새로운 사람을 추가한다. 동일 이름이 이미 있으면 거부한다.
Company? Pukyong National University    // 이런 방식으로 이름을 제외한 각 항목들을
Address? Yongso-ro 45, Nam-gu, Pusan    // 물어본다.
Zipcode? 48513
Email? kim@pknu.ac.kr
Kim was successfully added.
$ trace Kim            // 이름으로 검색하면서 루트에서 해당 노드까지 찾아가는 동안
...                    // 방문하는 모든 노드에 저장된 이름들을 한 줄에 하나씩 출력한다.

$ delete Annelle      // 삭제한다.
...
$ save address_book2.tsv // address_book2.tsv라는 파일에 동일한 형식으로 저장한다.

$ exit

```