**Min and max**

Given a binary tree, find and return the min and max data value of given binary tree.

Return the output as an object of PairAns class, which is already created.

**Input format :**

Elements in level order form (separated by space)

(If any node does not have left or right child, take -1 in its place)

**Output Format :**

Max and min (separated by space)

**Sample Input :**

8 3 10 1 6 -1 14 -1 -1 4 7 13 -1 -1 -1 -1 -1 -1 -1

**Sample Output :**

14 1

**Level order traversal**

Given a binary tree, print the level order traversal. Make sure each level start in new line.

Input format :

Elements in level order form (separated by space). If any node does not have left or right child, take -1 in its place.

Output Format :

Elements are printed level wise, each level in new line (separated by space).

**Sample Input :**

5 6 10 2 3 -1 -1 -1 -1 -1 9 -1 -1

**Sample Output :**

5

6 10

2 3

9

**Path Sum Root to Leaf**

Given a binary tree and a number k, print out all root to leaf paths where the sum of all nodes value is same as the given number k.

Input format :

Line 1 : Elements in level order form (separated by space)

(If any node does not have left or right child, take -1 in its place)

Line 2 : k

Output format : Print each path in new line, elements separated by space

**Sample Input 1 :**

5 6 7 2 3 -1 1 -1 -1 -1 9 -1 -1 -1 -1

13

**Sample Output 1 :**

5 6 2

5 7 1

**Print nodes at distance k from node**

Given a binary tree, a node and an integer K, print nodes which are at K distance from the the given node.

Input format :

Line 1 : Elements in level order form (separated by space)

(If any node does not have left or right child, take -1 in its place)

Line 2 : Node

Line 3 : K

Output format : Answer nodes in different line

**Sample Input :**

5 6 10 2 3 -1 -1 -1 -1 -1 9 -1 -1

3

1

**Sample Output :**

9

6