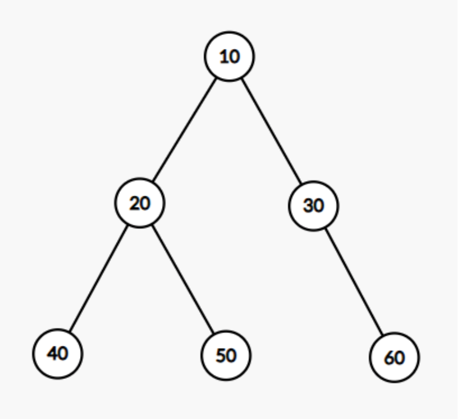
Sum of Tree

Problem Statement

You will be given a binary tree as input in level order. You need to output the sum of all node's values in that tree.

For example:



The output for the above tree will be: **210**

Input Format

- Input will contain the binary tree in level order. -1 means there is no node available.

Constraints

1. 1 <= **Maximum number of nodes** <= 10^5
2. 1 <= **Node's value** <= 1000

Output Format

- Output the total sum of that tree.

Sample Input 0

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

Sample Output 0

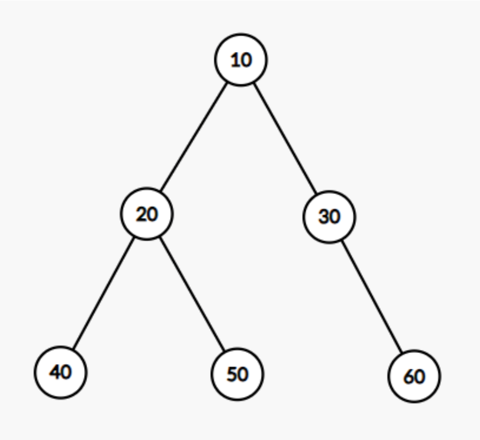
```
210
```

Max Min Leaf

Problem Statement

You will be given a binary tree as input in level order. You need to give the maximum and minimum values of all the leaf nodes available.

For example:



The output for the above tree will be: 60 40

Input Format

- **Input will contain the binary tree in level order. -1 means there is no node available.**

Constraints

1. 1 <= Maximum number of nodes <= 10^5
2. 1 <= Node's value <= 1000

Output Format

- Output the maximum value then the minimum value of all leaf nodes.

Sample Input 0

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

Sample Output 0

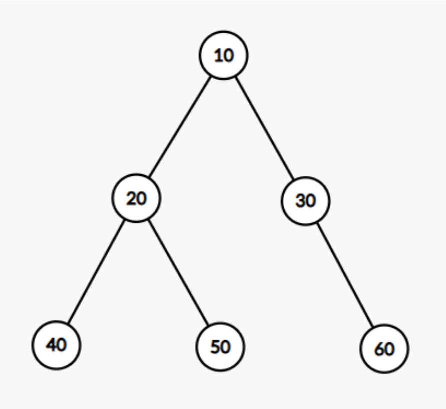
```
60 40
```

Print Tree

Problem Statement

You will be given a binary tree as input in level order. You need to print the binary tree in reverse way. Here, reverse way means you need to print from the last level and from left to right.

For example:



The output for the above tree will be: 40 50 60 20 30 10

Input Format

- **Input will contain the binary tree in level order. -1 means there is no node available.**

Constraints

1. **1 <= Maximum number of nodes <= 10^5**
2. **1 <= Node's value <= 1000**

Output Format

- **Output the tree in reverse way as described.**

Sample Input 0

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

Sample Output 0

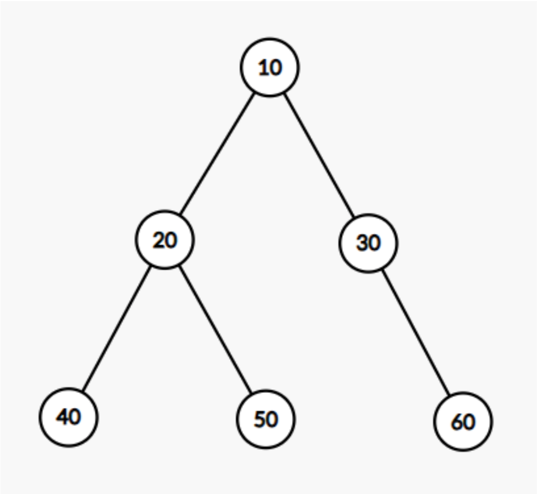
```
40 50 60 20 30 10
```

Level Nodes

Problem Statement

You will be given a binary tree as input in level order. Also you will be given a level X. You need to print all the node's values in that level from left to right. Assume that level starts from 0.

For example:



If X=2, then the output for the above tree will be: 40 50 60

Note: If the level X is not a valid level, the print "Invalid".

Input Format

- **Input will contain the binary tree in level order. -1 means there is no node available.**

Constraints

1. **1 <= Maximum number of nodes <= 10^5**
2. **1 <= Node's value <= 1000**
3. **0 <= X <= 10^5**

Output Format

- **Output all the node's values in level X.**

Sample Input 0

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

```
0
```

Sample Output 0

```
10
```

Sample Input 1

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

```
1
```

Sample Output 1

```
20 30
```

Sample Input 2

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

```
2
```

Sample Output 2

```
40 50 60
```

Sample Input 3

```
10 20 30 40 50 -1 60 -1 -1 -1 -1 -1 -1
```

```
3
```

Sample Output 3

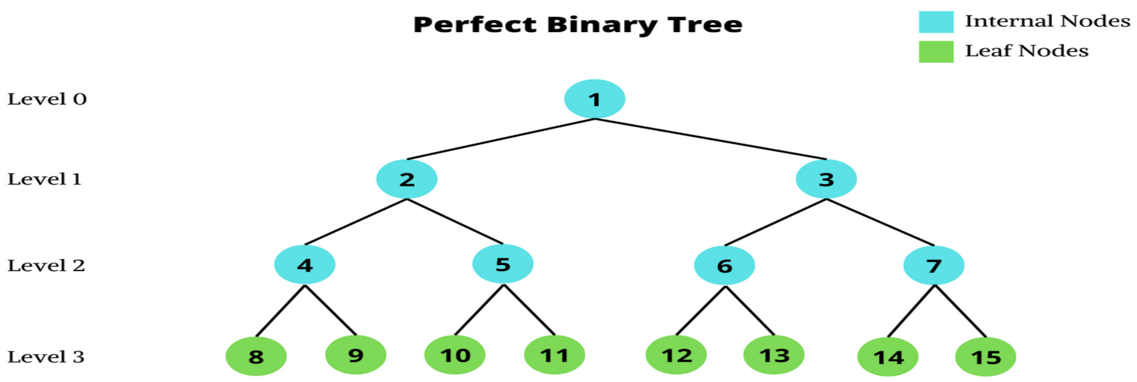
```
Invalid
```

Perfect Binary Tree

Problem Statement

You will be given a binary tree as input in level order. You need to tell if the binary tree is perfect or not. A binary tree is called perfect if all leaf nodes are at the maximum depth of the tree, and the tree is completely filled with no gaps.

Here is an example of perfect binary tree:



Also there is formula available to tell if a binary tree is perfect or not. The formula is :

- **Total number of nodes = (2^maxHeight)-1**

Note: Here height is counted from 1. In the above image maximum height is 4, so total number of nodes are 2^4-1=15. So there should be 15 nodes to call it a perfect binary tree.

Input Format

- **Input will contain the binary tree in level order. -1 means there is no node available.**

Constraints

1. **1 <= Maximum number of nodes <= 10^5**
2. **1 <= Node's value <= 1000**

Output Format

- **Output "YES" if the tree is perfect, "NO" otherwise.**

Sample Input 0

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Sample Output 0

```
YES
```

Sample Input 1

```
10 20 30 40 -1 60 -1 -1 -1 -1 -1
```

Sample Output 1

```
NO
```

Sample Input 2

```
10 20 -1 -1 -1
```

Sample Output 2

```
NO
```

Sample Input 3

```
10 20 30 40 50 60 70 -1 -1 -1 -1 -1 -1 -1 -1
```

Sample Output 3

```
YES
```