

# Tree Serialization Translator

In this challenge, you change the serialization approach used for a binary tree storing integer values. The tree will have been serialized using a level-order traversal, and you must output it given must translate it to a serialization using a pre-order traversal.

Note that the serialized tree, given as input, is generated as though there were nodes at each null child pointer. These nodes are listed in the input as having a value `-`.

*Constraint on your solution: Your solution must deserialize the tree by making use of a `TreeNode` class which stores a data field, an `LC` field, and an `RC` field.*

Notes:

Some of the test cases are very large, and may require you to speed up input and output handling. Note that you only need to try this if you are getting a time limit exceeded error when submitting your solution.

In C++, for example, you can include the following line as the first line in your main function to speed up the reading from input:

```
std::ios_base::sync_with_stdio (false);
```

And in Java, you can use a `BufferedReader` to greatly speed up reading from input, e.g.:

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
// Read next line of input which contains an integer:
int T = Integer.valueOf(reader.readLine());
```

Furthermore, in Java, you can speed up output by collecting all of your output in a `StringBuffer`, and then making just one call to `System.out.print`, e.g.:

```
// This would be slow
for (int i = 0; i < 1000000; i++) {
    System.out.println(i);
}

// This would be much faster
StringBuffer sb = new StringBuffer();
for (int i = 0; i < 1000000; i++) {
    sb.append(i);
    sb.append("\n");
}
System.out.println(sb.toString());
```

## Input Format

The input consists of list of node values (including `-` for the null nodes listed above), one per line, as they would be visited in a level-order traversal.

## Constraints

Each node value will be an integer with a value between  $-10^9$  and  $10^9$ . The total number of nodes in the tree (including the null nodes) will be less than  $5 * 10^5$ .

## Output Format

You should output the nodes as they would be printed in a pre-order traversal of the tree, including the `-` symbol for null nodes as in the input.

## Sample Input

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

Sample Output

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

Explanation

The tree in the input would be deserialized as:

