

# CS2023 - Data Structures and Algorithms

## In-class Lab Exercise

Week 7

---

You are required to answer the below questions and submit a PDF to the submission link provided under this week lab section before end of the session time (no extensions will be provided). You can either write / type your answers, but either way your answers should be readable.

---

**GitHub repository** - <https://github.com/namiwijeum/CS2023-Data-Structures-and-Algorithms-In-class-Lab-Exercises.git>

### Exercise:

Modify the given program to implement a binary search tree with the following basic operations. You have to define the below functions to implement the operations.

- *insertNode()*
- *deleteNode()*
- Additionally, you have to implement *traverseInOrder()* function to traverse the BST inorder.

Do not modify the main function and other utility functions. You may implement any additional utility functions as you need.

### ***Input Format***

Each line has two space-separated integers. The first integer is the operator (corresponds to the integer above), while the second integer is the operand.

-1 marks the end of the input sequence.

### ***Constraints***

1 <= operator <= 2

-10000 <= operators <= 10000

### ***Output Format***

Prints the resulting BST after performing a sequence of insert and delete operations on the BST, using in order traversal. Each number is separated by a space.

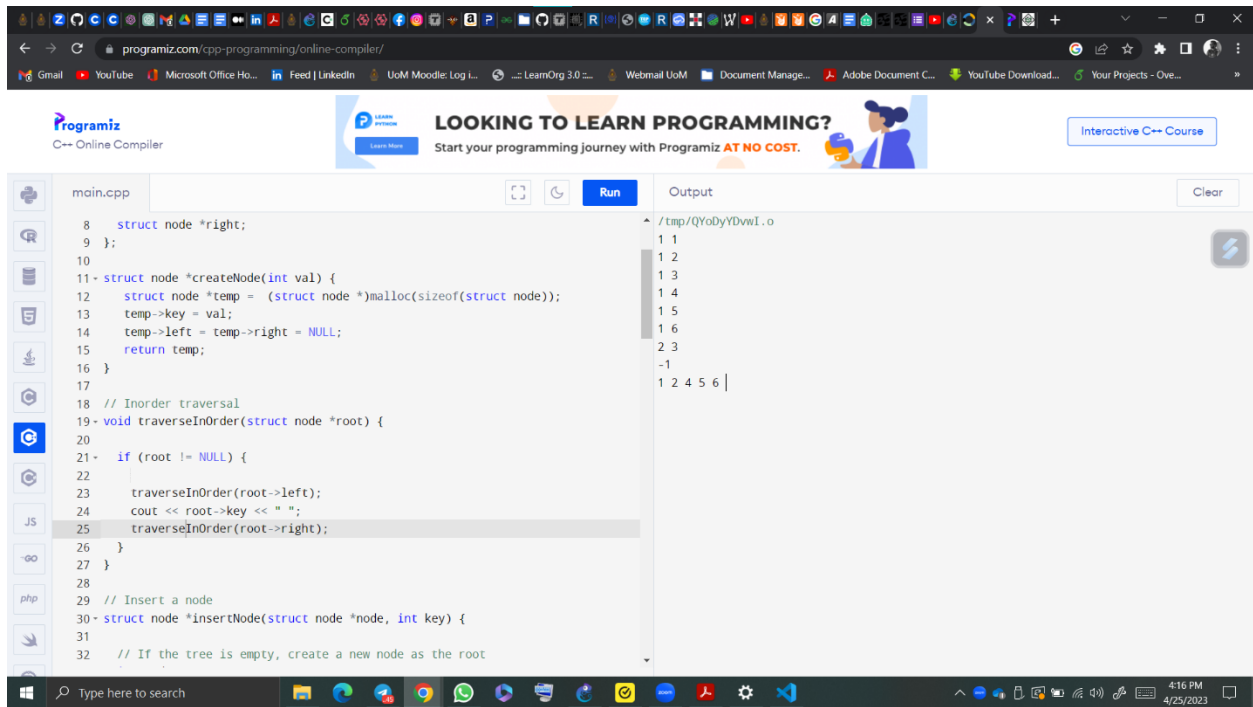
### Sample Input

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

### Sample Output

```
1 2 4 5 6
```

## Answer



The screenshot shows a web browser window with the URL `programiz.com/cpp-programming/online-compiler/`. The page features a banner for "Programiz C++ Online Compiler" and a promotional message: "LOOKING TO LEARN PROGRAMMING? Start your programming journey with Programiz AT NO COST. Interactive C++ Course".

The main content area displays a C++ program in a code editor. The code defines a `struct node` and implements functions for creating nodes, traversing the tree in-order, and inserting new nodes. The `main` function uses these utilities to build a binary tree from the sample input and prints the result.

```
main.cpp
8 struct node *right;
9 };
10
11 struct node *createNode(int val) {
12     struct node *temp = (struct node *)malloc(sizeof(struct node));
13     temp->key = val;
14     temp->left = temp->right = NULL;
15     return temp;
16 }
17
18 // Inorder traversal
19 void traverseInOrder(struct node *root) {
20
21     if (root != NULL) {
22         traverseInOrder(root->left);
23         cout << root->key << " ";
24         traverseInOrder(root->right);
25     }
26 }
27
28
29 // Insert a node
30 struct node *insertNode(struct node *node, int key) {
31
32     // If the tree is empty, create a new node as the root
```

The output window on the right shows the execution result:

```
/tmp/QYoByYDvwI.o
1 1
1 2
1 3
1 4
1 5
1 6
2 3
-1
1 2 4 5 6 |
```

The Windows taskbar at the bottom shows the system time as 4:16 PM on 4/25/2023.