

.vscode\DSA_College\almost_complete_binary_tree.cpp

```
1  #include<iostream>
2  #include<queue>
3  using namespace std ;
4
5  class node{
6      public :
7      int data ;
8      node* left ;
9      node* right ;
10
11      // constructor
12      node(int d)
13      {
14          data = d ;
15          left = NULL ;
16          right = NULL ;
17      }
18 };
19
20 void insert(node* &root, int data)
21 {
22     node* newNode = new node(data) ;
23     if(root == NULL)
24     {
25         root = newNode ;
26         return ;
27     }
28
29     queue<node*> q ;
30     q.push(root) ;
31
32     while(!q.empty())
33     {
34         node* temp = q.front() ;
35         q.pop() ;
36
37         if(temp->left == NULL)
38         {
39             temp->left = newNode ;
40             break ;
41         }
42         else{
43             q.push(temp->left) ;
44         }
45
46         if(temp->right == NULL)
47         {
48             temp->right = newNode ;
49             break ;
50         }
51         else{
```

```
52         q.push(temp->right) ;
53     }
54 }
55 }
56
57 void levelOrderTraversal(node* root)
58 {
59     if(root == NULL)
60         return ;
61
62     queue<node*> q ;
63     q.push(root) ;
64
65     while(!q.empty())
66     {
67         node* temp = q.front() ;
68         q.pop() ;
69
70         cout<<temp->data<<" " ;
71
72         if(temp->left)
73         {
74             q.push(temp->left) ;
75         }
76         if(temp->right)
77         {
78             q.push(temp->right) ;
79         }
80     }
81     cout<<endl ;
82 }
83
84 int main()
85 {
86     node* root = NULL;
87
88     // Inserting nodes to form an almost complete binary tree
89     insert(root, 1);
90     insert(root, 2);
91     insert(root, 3);
92     insert(root, 4);
93     insert(root, 5);
94     insert(root, 6);
95
96     cout << "Level-order traversal of the tree: ";
97     levelOrderTraversal(root);
98
99     return 0;
100 }
```