

```

1: // A recursive C program to print REVERSE level order traversal
2: #include <stdio.h>
3: #include <stdlib.h>
4:
5: /* A binary tree node has data, pointer to left and right child */
6: struct node
7: {
8:     int data;
9:     struct node* left;
10:    struct node* right;
11: };
12:
13: /*Function prototypes*/
14: void printGivenLevel(struct node* root, int level);
15: int height(struct node* node);
16: struct node* newNode(int data);
17:
18: /* Function to print REVERSE level order traversal of a tree */
19: void reverseLevelOrder(struct node* root)
20: {
21:     int h = height(root);
22:     int i;
23:     for (i=h; i>=1; i--) //THE ONLY LINE DIFFERENT FROM NORMAL LEVEL ORDER TRaversal
24:         printGivenLevel(root, i);
25: }
26:
27: /* Print nodes at a given level */
28: void printGivenLevel(struct node* root, int level)
29: {
30:     if (root == NULL)
31:         return;
32:     if (level == 1)
33:         printf("%d ", root->data);
34:     else if (level > 1)
35:     {
36:         printGivenLevel(root->left, level-1);
37:         printGivenLevel(root->right, level-1);
38:     }
39: }

```

```

40:
41: /* Compute the "height" of a tree -- the number of
42: nodes along the longest path from the root node
43: down to the farthest leaf node.*/
44: int height(struct node* node)
45: {
46:     if (node==NULL)
47:         return 0;
48:     else
49:     {
50:         /* compute the height of each subtree */
51:         int lheight = height(node->left);
52:         int rheight = height(node->right);
53:
54:         /* use the larger one */
55:         if (lheight > rheight)
56:             return(lheight+1);
57:         else return(rheight+1);
58:     }
59: }
60:
61: /* Helper function that allocates a new node with the
62: given data and NULL left and right pointers. */
63: struct node* newNode(int data)
64: {
65:     struct node* node = (struct node*)
66:         malloc(sizeof(struct node));
67:     node->data = data;
68:     node->left = NULL;
69:     node->right = NULL;
70:
71:     return(node);
72: }
73:
74: /* Driver program to test above functions*/
75: int main()
76: {
77:     struct node *root = newNode(1);
78:     root->left = newNode(2);

```

```
79:     root->right = newNode(3);
80:     root->left->left = newNode(4);
81:     root->left->right = newNode(5);
82:
83:     printf("Level Order traversal of binary tree is \n");
84:     reverseLevelOrder(root);
85:
86:     return 0;
87: }
88:
```