```c
// Iterative Queue based C program
// to do level order traversal
// of Binary Tree
#include <stdio.h>
#include <stdlib.h>
#define MAX_Q_SIZE 500

/* A binary tree node has data,
pointer to left child
and a pointer to right child */
struct node {
    int data;
    struct node* left;
    struct node* right;
};

/* frunction prototypes */
struct node** createQueue(int*, int*);
void enQueue(struct node**, int*, struct node*);
struct node* deQueue(struct node**, int*);

/* Given a binary tree, print its nodes in level order
using array for implementing queue */
void printLevelOrder(struct node* root)
{
    int rear, front;
    struct node** queue = createQueue(&front, &rear);
    struct node* temp_node = root;

    while (temp_node) {
        printf("%d ", temp_node->data);

        /*Enqueue left child */
        if (temp_node->left)
            enQueue(queue, &rear, temp_node->left);

        /*Enqueue right child */
        if (temp_node->right)
            enQueue(queue, &rear, temp_node->right);
```

```
40:
41:         /*Dequeue node and make it temp_node*/
42:         temp_node = deQueue(queue, &front);
43:     }
44: }
45:
46: /*UTILITY FUNCTIONS*/
47: struct node** createQueue(int* front, int* rear)
48: {
49:     struct node** queue = (struct node**)malloc(
50:         sizeof(struct node*) * MAX_Q_SIZE);
51:
52:     *front = *rear = 0;
53:     return queue;
54: }
55:
56: void enQueue(struct node** queue, int* rear,
57:              struct node* new_node)
58: {
59:     queue[*rear] = new_node;
60:     (*rear)++;
61: }
62:
63: struct node* deQueue(struct node** queue, int* front)
64: {
65:     (*front)++;
66:     return queue[*front - 1];
67: }
68:
69: /* Helper function that allocates a new node with the
70: given data and NULL left and right pointers. */
71: struct node* newNode(int data)
72: {
73:     struct node* node
74:         = (struct node*)malloc(sizeof(struct node));
75:     node->data = data;
76:     node->left = NULL;
77:     node->right = NULL;
78:
```

```
79:     return (node);
80: }
81:
82: /* Driver program to test above functions*/
83: int main()
84: {
85:     struct node* root = newNode(1);
86:     root->left = newNode(2);
87:     root->right = newNode(3);
88:     root->left->left = newNode(4);
89:     root->left->right = newNode(5);
90:
91:     printf("Level Order traversal of binary tree is \n");
92:     printLevelOrder(root);
93:
94:     return 0;
95: }
96:
```