```
//Created By Ritwik Chandra Pandey on 1/4/21
//183215
//Expression Tree : Level Order Traversal

#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);

    /*Dequeue node and make it temp_node*/
    temp_node = deQueue(queue, &front);
  }
}

/*UTILITY FUNCTIONS*/
struct node** createQueue(int *front, int *rear)
{
  struct node **queue =
    (struct node **)malloc(sizeof(struct node*)
                *MAX_Q_SIZE);

  *front = *rear = 0;
  return queue;
}

void enQueue(struct node **queue, int *rear,
             struct node *new_node)
{
  queue[*rear] = new_node;
  (*rear)++;
}

struct node *deQueue(struct node **queue, int *front)
{
  (*front)++;
  return queue[*front - 1];
}
```

```
/* Helper function that allocates a new node with the
  given data and NULL left and right pointers. */
struct node* newNode(int data)
{
  struct node* node = (struct node*)
        malloc(sizeof(struct node));
  node->data = data;
  node->left = NULL;
  node->right = NULL;

  return(node);
}

/* Driver program to test above functions*/
int main()
{
  struct node *root = newNode(1);
  root->left    = newNode(2);
  root->right    = newNode(3);
  root->left->left  = newNode(4);
  root->left->right = newNode(5);

  printf("Level Order traversal of binary tree is \n");
  printLevelOrder(root);

  return 0;
}
```