```
//Created By Ritwik Chandra Pandey on 25th March
//183215
//Level Order Traversal of n-ary Tree



#include <stdio.h>
#include <stdlib.h>
#define MAX_Q_SIZE 500


struct node
{
    int data;
    struct node* first;
    //struct node* right;
    struct node* next_sibling;
};

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

void printLevelOrder(struct node* root)
{
    int rear, front;
    struct node **queue = createQueue(&front, &rear);
    struct node *temp_node = root;
    struct node *c;

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

```
        if (temp_node->first){
            enQueue(queue, &rear, temp_node->first);

        }

       c = temp_node->first;




while(c!=NULL){



    if(c->next_sibling!=NULL)
        enQueue(queue, &rear,c->next_sibling);


    c = c->next_sibling;

      }




        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];
}


struct node* newNode(int data)
{
    struct node* node = (struct node*)malloc(sizeof(struct node));
    node->data = data;
    node->first = NULL;
    node->next_sibling = NULL;


    return(node);
}

/* Driver program to test above functions*/
int main()
{
    struct node *root = newNode(01);
    root->first       = newNode(12);
```

```
    root->first->next_sibling       = newNode(13);
    root->first->next_sibling->next_sibling  = newNode(14);
    root->first->next_sibling->next_sibling->first = newNode(25);
    root->first->next_sibling->next_sibling->next_sibling = newNode(15);
    root->first->first = newNode(26);
    root->first->first->next_sibling = newNode(27);
    root->first->first->first = newNode(38);
    printf("Level Order traversal of the tree is \n");
     printLevelOrder(root);

    printf("\n");
    return 0;
}
```