**Collage : Vishwakarma Institute of Technology**

**Course Name : Data Structure in C**

**Name : Vedika Vikas Sontakke**

**Roll no : 37**

**PRN NO 12220206**

# Assignment no 6 : Implement BFS and DFS traversal on Graphs.

# Program : ( For BFS )

#include<stdio.h>

#include<stdlib.h>

struct queue

{

    int size;

    int front;

    int rear;

    int \*arr;

};

int isEmpty(struct  queue \*queue)

{

    if(queue->front == queue->rear) return 1;

    else return 0;

}

int isFull(struct queue \*queue)

{

    if(queue->rear == queue->size-1) return 1;

    else return 0;

}

void enqueue(struct queue \*queue , int val)

{

   if(isFull(queue)) printf("queue is full\n");

   else queue->arr[++queue->rear] = val;

}

int dequeue(struct queue \*queue)

{

    int dequeue\_element = -1;

    if(isEmpty(queue)) printf("queue is empty\n");

    else dequeue\_element = queue->arr[++queue->front];

    return dequeue\_element;

}

int main()

{

    // initialization of queue

    struct queue q;

    q.size = 400;

    q.front = q.rear = 0;

    q.arr = (int\*)malloc(q.size\*sizeof(int));

    int node;

    int i=1;        // root element which will be visted first;

    int visted[7] = {0 , 0 , 0 , 0 , 0 , 0 , 0};

    int a[7][7] = {

        {0,1,1,1,0,0,0},

        {1,0,1,0,0,0,0},

        {1,1,0,1,1,0,0},

        {1,0,1,0,1,0,0},

        {0,0,1,1,0,1,1},

        {0,0,0,0,1,0,0},

        {0,0,0,0,1,0,0}

    };

    printf("%d ",i);

    visted[i] = 1;

    enqueue(&q , i); // insert 1st elemnt that is root element

    while(!isEmpty(&q))

    {

       int node = dequeue(&q);

       for(int i=0 ; i<7 ; i++)

       {

         if(a[node][i]==1 && visted[i]==0)

         {

            printf("%d ",i);

            visted[i]=1;

            enqueue(&q,i);

         }

       }

    }

     return 0;

}

# Output : ( For BFS )

# 

# Program : ( For DFS )

#include<stdio.h>

int visted[7] = {0,0,0,0,0,0,0};

 int A [7][7] = {

        {0,1,1,1,0,0,0},

        {1,0,1,0,0,0,0},

        {1,1,0,1,1,0,0},

        {1,0,1,0,1,0,0},

        {0,0,1,1,0,1,1},

        {0,0,0,0,1,0,0},

        {0,0,0,0,1,0,0}

    };

void dfs(int i)

{

    printf(" %d ",i);

    visted[i] = 1;

    for(int j=0 ; j<7 ; j++)

    {

        if(A[i][j]==1 && visted[j]==0) dfs(j);

    }

}

int main()

{

    dfs(0);

}

# Output : ( For DFS )

# 