**Code:-**

# **Name : Rajkumar B L**

# **Reg.No : 2047120**

# **Course : MCS 271 Data Structure (Lab 10 – BFS & DFS)**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Name : Rajkumar B L

 \* Reg  : 2047120

 \* Lab  : 10

 \* Program : BFS & DFS

 \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

int q[20], top = -1, front = -1, rear = -1, a[20][20], vis[20], stack[20];

int delete ();

void add(int item);

void bfs(int s, int n);

void dfs(int s, int n);

void push(int item);

int pop();

void main()

{

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\*  Name : Rajkumar B L     \*\n\*  Reg  : 2047120          \*\n\*  Lab  : 10               \*\n\*  Prg  : BFS & DFS        \*\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

    int n, i, s, ch, j;

    char c, dummy;

    printf("Enter the no of vertices:");

    scanf("%d", &n);

    printf("Lets draw the graph :\n");

    for (i = 1; i <= n; i++)

    {

        printf("Enter row %d : ", i);

        for (j = 1; j <= n; j++)

            scanf("%d", &a[i][j]);

    }

    do

    {

        for (i = 1; i <= n; i++)

            vis[i] = 0;

        printf("\n=========================\n\tMenu\n=========================\n");

        printf("1. Breadth First Search\n");

        printf("2. Depth First Search\n");

        printf("3. Exit\n");

        printf("=========================\n");

        printf("Enter your choice: ");

        fflush(stdin);

        scanf("%d", &ch);

        switch (ch)

        {

        case 1:

            printf("Enter the source vertex: ");

            fflush(stdin);

            scanf("%d", &s);

            printf("Output:");

            bfs(s, n);

            break;

        case 2:

            printf("Enter the source vertex: ");

            fflush(stdin);

            scanf("%d", &s);

            printf("Output:");

            dfs(s, n);

            break;

        case 3:

            printf("Bye.\n");

            break;

        default:

            printf("Invalid choice.\n");

            break;

        }

        printf("\n");

    } while (ch != 3);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*BFS(breadth-first search) code\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void bfs(int s, int n)

{

    int p, i;

    add(s);

    vis[s] = 1;

    p = delete ();

    if (p != 0)

        printf(" %d", p);

    while (p != 0)

    {

        for (i = 1; i <= n; i++)

            if ((a[p][i] != 0) && (vis[i] == 0))

            {

                add(i);

                vis[i] = 1;

            }

        p = delete ();

        if (p != 0)

            printf(" %d ", p);

    }

    for (i = 1; i <= n; i++)

        if (vis[i] == 0)

            bfs(i, n);

}

void add(int item)

{

    if (rear == 19)

        printf("QUEUE FULL");

    else

    {

        if (rear == -1)

        {

            q[++rear] = item;

            front++;

        }

        else

            q[++rear] = item;

    }

}

int delete ()

{

    int k;

    if ((front > rear) || (front == -1))

        return (0);

    else

    {

        k = q[front++];

        return (k);

    }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DFS(depth-first search) code\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void dfs(int s, int n)

{

    int i, k;

    push(s);

    vis[s] = 1;

    k = pop();

    if (k != 0)

        printf(" %d", k);

    while (k != 0)

    {

        for (i = 1; i <= n; i++)

            if ((a[k][i] != 0) && (vis[i] == 0))

            {

                push(i);

                vis[i] = 1;

            }

        k = pop();

        if (k != 0)

            printf(" %d", k);

    }

    for (i = 1; i <= n; i++)

        if (vis[i] == 0)

            dfs(i, n);

}

void push(int item)

{

    if (top == 19)

        printf("Stack overflow ");

    else

        stack[++top] = item;

}

int pop()

{

    int k;

    if (top == -1)

        return (0);

    else

    {

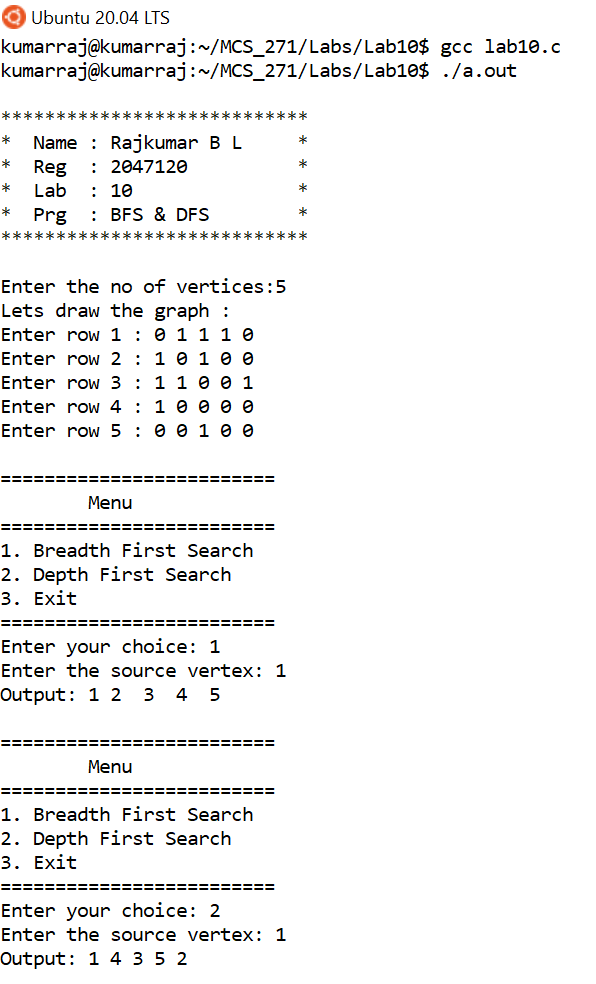
        k = stack[top--];

        return (k);

    }

}

**Output:**

****