Name: Surwade Trisharan Rajesh

Roll no.: 48

//Write a program to implement breadth first search

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

int a[10][10], i, j, v[10], n, q[15], f, b, r;

class bfsl

{

public:

    void getdata();

    int bfs(int vl);

    void display();

};

void bfsl::getdata()

{

    std::cout << "Enter the no. ofn vertices:\t";

    cin >> n;

    std::cout << "Enter the matrix:=";

    for (i = 1; i <= n; i++)

        for (j = 1; j <= n; j ++ )

            cin >> a[i][j];

}

int bfsl::bfs(int vl)

{

    int w;

    v[vl] = 1;

    std::cout << vl;

    f = r = 0;

    while (1)

    {

        for (w = 1; w <= n; w++)

        {

            if (a[vl][w] == 1)

            {

                if (v[w] == 0)

                {

                    if ((f == 0) && (r == 0))

                        f = r = 1;

                    else

                        r++;

                    q[r] = w;

                    v[w] = 1;

                    std::cout << "\t" << w;

                }

            }

        }

        if ((f == 0) && (r == 0))

            return 0;

        vl = q[f];

        if (f == r)

            f = r = 0;

        else

            f++;

    }

}

void bfsl::display()

{

    std::cout << "Sequenced of node in bfs is:=";

    bfs(1);

    cout << "\n";

    for (i = 1; i <= n; i++)

        v[i] = 0;

}

int main()

{

    bfsl b;

    b.getdata();

    b.display();

    return 0;

}

Output:

Enter the no. ofn vertices: 5

Enter the matrix:=0 1 1 0 0

0 0 0 1 1

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Sequenced of node in bfs is:=1 2 3 4 5

//Write a program to implement depth first search.

#include<iostream>

#include<conio.h>

#include<stdio.h>

using namespace std;

int a[10][10], j, i, v[10], n, q[15], f, b, r;

class dfs1

{

public:

    void getdata();

    int dfs(int i);

    void display();

};

void dfs1::getdata()

{

    cout << "Enter the vertices : ";

    cin >> n;

    cout << "Enter the matrix : ";

    for (i = 1; i <= n; i++)

        for (j = 1; j <= n; j++)

            cin >> a[i][j];

}

int dfs1::dfs(int i)

{

    int w;

    v[i] = 1;

    cout << i;

    for (w = 1; w <= n; w++)

    {

        if (a[i][w] == 1)

        {

            if (v[w] == 0)

                dfs(w);

        }

    }

    return(0);

}

void dfs1::display()

{

    cout << "\n Sequence of node in dfs are : ";

    dfs(1);

    cout << "\n";

    for (i = 1; i <= n; i++)

        v[i] = 0;

}

int main()

{

    dfs1 b;

    b.getdata();

    b.display();

    return 0;

}

Output:

Enter the vertices : 5

Enter the matrix : 0 1 1 0 0

0 0 0 1 1

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

Sequence of node in dfs are : 12453