**PRACTICAL NO: 06.**

**PROGRAM CODE:**

#include <iostream>

#include <queue>

using namespace std;

int adj\_mat[50][50] = {0, 0};

int visited[50] = {0};

void dfs(int s, int n, string arr[])

{

visited[s] = 1;

cout << arr[s] << " ";

for (int i = 0; i < n; i++)

{

if (adj\_mat[s][i] && !visited[i])

dfs(i, n, arr);

}

}

void bfs(int s, int n, string arr[])

{

bool visited[n];

for (int i = 0; i < n; i++)

visited[i] = false;

int v;

queue<int> bfsq;

if (!visited[s])

{

cout << arr[s] << " ";

bfsq.push(s);

visited[s] = true;

while (!bfsq.empty())

{

v = bfsq.front();

for (int i = 0; i < n; i++)

{

if (adj\_mat[v][i] && !visited[i])

{

cout << arr[i] << " ";

visited[i] = true;

bfsq.push(i);

}

}

bfsq.pop();

}

}

}

int main()

{

cout << "Enter no. of cities: ";

int n, u;

cin >> n;

string cities[n];

for (int i = 0; i < n; i++)

{

cout << "Enter city #" << i << " (Airport Code): ";

cin >> cities[i];

}

cout << "\nYour cities are: " << endl;

for (int i = 0; i < n; i++)

cout << "city #" << i << ": " << cities[i] << endl;

for (int i = 0; i < n; i++)

{

for (int j = i + 1; j < n; j++)

{

cout << "Enter distance between " << cities[i] << " and " << cities[j] << " : ";

cin >> adj\_mat[i][j];

adj\_mat[j][i] = adj\_mat[i][j];

}

}

cout << endl;

for (int i = 0; i < n; i++)

cout << "\t" << cities[i] << "\t";

for (int i = 0; i < n; i++)

{

cout << "\n"

<< cities[i];

for (int j = 0; j < n; j++)

cout << "\t" << adj\_mat[i][j] << "\t";

cout << endl;

}

cout << "Enter Starting Vertex: ";

cin >> u;

cout << "DFS: ";

dfs(u, n, cities);

cout << endl;

cout << "BFS: ";

bfs(u, n, cities);

return 0;

}

**OUTPUT:**

**A.**

Enter no. of cities: 3

Enter city #0 (Airport Code): NSK

Enter city #1 (Airport Code): PUNE

Enter city #2 (Airport Code): MUMB

Your cities are:

city #0: NSK

city #1: PUNE

city #2: MUMB

Enter distance between NSK and PUNE : 1

Enter distance between NSK and MUMB : 0

Enter distance between PUNE and MUMB : 1

NSK PUNE MUMB

NSK 0 1 0

PUNE 1 0 1

MUMB 0 1 0

Enter Starting Vertex: NSK

DFS: NSK PUNE MUMB

BFS: NSK PUNE MUMB

--------------------------------

Process exited after 44 seconds with return value 0

Press any key to continue . . .

**B.**

Enter no. of cities: 4

Enter city #0 (Airport Code): BANGLORE

Enter city #1 (Airport Code): CHENNAI

Enter city #2 (Airport Code): KERALA

Enter city #3 (Airport Code): ANDRA

Your cities are:

city #0: BANGLORE

city #1: CHENNAI

city #2: KERALA

city #3: ANDRA

Enter distance between BANGLORE and CHENNAI : 1

Enter distance between BANGLORE and KERALA : 1

Enter distance between BANGLORE and ANDRA : 0

Enter distance between CHENNAI and KERALA : 0

Enter distance between CHENNAI and ANDRA : 1

Enter distance between KERALA and ANDRA : 0

BANGLORE CHENNAI KERALA ANDRA

BANGLORE 0 1 1 0

CHENNAI 1 0 0 1

KERALA 1 0 0 0

ANDRA 0 1 0 0

Enter Starting Vertex: CHENNAI

DFS: BANGLORE CHENNAI ANDRA KERALA

BFS: BANGLORE CHENNAI KERALA ANDRA

--------------------------------

Process exited after 76.08 seconds with return value 0

Press any key to continue . . .