DS TUTORIAL 7

Name:Varad Tanawade

CODE 1:

```
#include <iostream>
#include <queue>
using namespace std;

int main() {
    cout << "B24CE1086\n";
    int n;
    cout << "Enter number of web pages: ";
    cin >> n;

    string pages[20];
    cout << "Enter names of web pages:\n";
    for (int i = 0; i < n; i++)
        cin >> pages[i];

    int links[20][20];
    cout << "Enter link matrix (1 if link exists, 0 otherwise):\n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            cin >> links[i][j];

    string start;
    cout << "Enter starting page: ";
    cin >> start;

    int startIndex = -1;
    for (int i = 0; i < n; i++)
        if (pages[i] == start)
            startIndex = i;

    if (startIndex == -1) {
        cout << "Starting page not found!" << endl;
        return 0;
    }

    bool visited[20] = {false};
    queue<int> q;
    visited[startIndex] = true;
    q.push(startIndex);

    cout << "\nIndexed (Visited) Web Pages in BFS Order:\n";
```

```
    while (!q.empty()) {
        int current = q.front();
        q.pop();
        cout << pages[current] << endl;

        for (int i = 0; i < n; i++) {
            if (links[current][i] == 1 && !visited[i]) {
                visited[i] = true;
                q.push(i);
            }
        }
    }
    return 0;
}
```

OUTPUT 1:

```
============B24CE1076============
Enter number of web pages: 5
Enter names of web pages:
A B C D E
Enter link matrix (1 if link exists, 0 otherwise):
0 1 1 0 0
0 0 0 1 0
0 1 0 0 1
0 0 0 0 0
0 0 0 0 0
Enter starting page: A

Indexed (Visited) Web Pages in BFS Order:
A
B
C
D
E
```

CODE 2:

```
#include <iostream>
using namespace std;

void dfs(int node, int n, int links[20][20], string pages[20], bool
visited[20]) {
    visited[node] = true;
    cout << pages[node] << endl;
    for (int i = 0; i < n; i++) {
        if (links[node][i] == 1 && !visited[i]) {
            dfs(i, n, links, pages, visited);
        }
    }
}

int main() {
    cout << "============B24CE1076==========\n";
    int n;
    cout << "Enter number of web pages: ";
    cin >> n;

    string pages[20];
    cout << "Enter names of web pages:\n";
    for (int i = 0; i < n; i++)
        cin >> pages[i];

    int links[20][20];
    cout << "Enter link matrix (1 if link exists, 0 otherwise):\n";
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            cin >> links[i][j];

    string start;
    cout << "Enter starting page: ";
    cin >> start;

    int startIndex = -1;
    for (int i = 0; i < n; i++)
        if (pages[i] == start)
            startIndex = i;

    if (startIndex == -1) {
        cout << "Starting page not found!" << endl;
        return 0;
    }
```

```
    bool visited[20] = {false};
    cout << "\nIndexed (Visited) Web Pages in DFS Order:\n";
    dfs(startIndex, n, links, pages, visited);

    return 0;
}
```

OUTPUT 2:

```
===========B24CE1076==========
Enter number of web pages: 5
Enter names of web pages:
Home Scan History Community Profile
Enter link matrix (1 if link exists, 0 otherwise):
0 1 1 0 0
0 0 0 1 1
0 0 0 0 0
0 0 0 0 1
0 0 0 0 0
Enter starting page: Home

Indexed (Visited) Web Pages in DFS Order:
Home
Scan
Community
Profile
History
```