ASSIGNMENT 7

Web Crawling

PROGRAM 1 :-

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

int main() {
    int n = 5; // Number of pages (nodes)
    int web[6][6] = {0};

    // Adjacency matrix representation
    web[1][2] = 1;
    web[1][3] = 1;
    web[2][4] = 1;
    web[3][4] = 1;
    web[3][5] = 1;

    int visited[6] = {0};
    int queue[10];
    int front = 0, rear = -1;

    int start = 1; // Start from Page1 (seed URL)
    visited[start] = 1;
    queue[++rear] = start;

    cout << "Starting BFS from Page" << start << endl;
    cout << "Indexed pages in BFS order: ";

    while (front <= rear) {
        int page = queue[front++];
        cout << "Page" << page << " ";
```

```
        for (int i = 1; i <= n; i++) {
            if (web[page][i] == 1 && visited[i] == 0) {
                queue[++rear] = i;
                visited[i] = 1;
            }
        }
    }

    cout << endl;
    return 0;
}
```

OUTPUT :-

```
Output
Starting BFS from Page1
Indexed pages in BFS order: Page1 Page2 Page3 Page4 Page5


=== Code Execution Successful ===
```

PROGRAM 2 :-

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

int web[6][6];
int visited[6];
int n = 5;

void DFS(int page) {
    cout << "Page" << page << " ";
    visited[page] = 1;
```

```
    for (int i = 1; i <= n; i++) {
        if (web[page][i] == 1 && visited[i] == 0)
            DFS(i);
    }
}

int main() {
    // Adjacency matrix representation
    web[1][2] = 1;
    web[1][3] = 1;
    web[2][4] = 1;
    web[3][4] = 1;
    web[3][5] = 1;

    for (int i = 1; i <= n; i++)
        visited[i] = 0;

    int start = 1; // Starting page
    cout << "Starting DFS from Page" << start << endl;
    cout << "Indexed pages in DFS order: ";
    DFS(start);

    cout << endl;
    return 0;
}
```

OUTPUT :-

```
Output
Starting DFS from Page1
Indexed pages in DFS order: Page1 Page2 Page4 Page3 Page5


=== Code Execution Successful ===
```