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

void dfs(int node, bool visited[], int graph[10][10], int n) {
    visited[node] = true;
    cout << "Indexed Page: " << node << endl;

    // Visit all connected pages (links)
    for (int i = 0; i < n; i++) {
        if (graph[node][i] == 1 && !visited[i]) {
            dfs(i, visited, graph, n);
        }
    }
}

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

    int graph[10][10];
    bool visited[10] = {false};

    cout << "Enter adjacency matrix (1 = link exists, 0 = no link):\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cin >> graph[i][j];
        }
    }

    int start;
    cout << "Enter starting web page (0 to " << n-1 << "): ";
    cin >> start;

    cout << "\n--- Web Crawling Order ---\n";
    dfs(start, visited, graph, n);

    cout << "\nAll reachable pages from " << start << " have been indexed.\n";
    return 0;
}
```

OUTPUT:-

```
Enter number of web pages: 4
Enter adjacency matrix:
0 1 1 0
1 0 0 1
0 1 0 1
0 0 0 1
Enter starting web page (0 to 3): 0

--- Web Crawling Order ---
Indexed Page: 0
Indexed Page: 1
Indexed Page: 3
Indexed Page: 2

All reachable pages from 0 have been indexed.
```