Practical 1 : DFS (Sequential and Parallel Algorithms)

```
#include <iostream>
#include <vector>
#include <stack>
#include <omp.h>
#include <cstdlib>

using namespace std;

const int MAX = 100000;
vector<int> graph[MAX];
bool visited[MAX];

void dfs(int node) {
   stack<int> s;
   s.push(node);

   while (!s.empty()) {
      int curr_node = s.top();
      s.pop();

      if (!visited[curr_node]) {
         visited[curr_node] = true;
         cout << curr_node << " ";

         #pragma omp parallel for
         for (int i = 0; i < graph[curr_node].size(); i++) {
            int adj_node = graph[curr_node][i];
            if (!visited[adj_node]) {
               s.push(adj_node);
            }
         }
      }
   }
}

int main() {
   cout << "This is Atharva Pingale's code";
   cout << "\nPractical 1 : DFS ( Sequential and Parallel algorithms )";
   int n, m, start_node;
   double start_time, end_time;
   cout << "\n\nEnter number of nodes : ";
   cin >> n;
   cout << "Enter number of edges : ";
   cin >> m;
   cout << "Enter the starting node of the graph : ";
   cin >> start_node;

   for (int i = 0; i < m; i++) {
      int random_u = rand() % 999999;
      int random_v = rand() % 999999;

      graph[random_u].push_back(random_v);
      graph[random_v].push_back(random_u);
   }
```

```
    // Sequential Algorithm
    for (int i = 0; i < n; i++) {
        visited[i] = false;
    }

    start_time = omp_get_wtime();
    dfs(start_node);
    end_time = omp_get_wtime();
    double seq_time = end_time - start_time;

    // Parallel Algorithm
    for (int i = 0; i < n; i++) {
        visited[i] = false;
    }

    start_time = omp_get_wtime();
    dfs(start_node);
    end_time = omp_get_wtime();
    double parallel_time = end_time - start_time;
    cout << "\n\nSequential Algorithm Time: " << seq_time << " seconds\n";
    cout << "\nParallel Algorithm Time: " << parallel_time << " seconds\n";

    return 0;
}
```

Output :

```
athar@LAPTOP-U0997R48 MINGW64 /d/GitHub/BE-8th-Semester/hpc_practicals (main)
$ g++ -fopenmp DFS.cpp -o DFS

athar@LAPTOP-U0997R48 MINGW64 /d/GitHub/BE-8th-Semester/hpc_practicals (main)
$ ./DFS
This is Atharva Pingale's code
Practical 1 : DFS ( Sequential and Parallel algorithms )

Enter number of nodes : 9999
Enter number of edges : 9999
Enter the starting node of the graph : 0
0 0

Sequential Algorithm Time: 0.0150001 seconds

Parallel Algorithm Time: 0 seconds

athar@LAPTOP-U0997R48 MINGW64 /d/GitHub/BE-8th-Semester/hpc_practicals (main)
$
```