Assignment 1

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

vector<bool> vis;
vector<vector<int>> adjlist;

// Function of DFS traversal
void DFS(int node) {
   vis[node] = true;
   cout << node << " ";

   for (int NODE : adjlist[node]) {
      if (!vis[NODE]) {
         DFS(NODE);
      }
   }
}

void addEdge(int parent, int child) {
   adjlist[parent].push_back(child);
   adjlist[child].push_back(parent);
}

int main() {
```

```
    // edges of the graph
    vector<pair<int, int>> edges = {
        {0, 1}, {0, 2}, {1, 3},
        {3, 7}, {3, 8},
        {1, 4}, {4, 9},
        {9, 13}, {9, 14},
        {2, 5}, {5, 10},
        {2, 10}, {2, 6},
        {6, 11},{6,12}
    };

    int elements = edges.size();

    // the adjacency list
    adjlist.resize(elements);
    for (auto EDGE : edges) {
        addEdge(EDGE.first, EDGE.second);
    }
    // starting from node 0
    cout << "starting DFS traversal from node 0:" << endl;
    vis.resize(elements, false);
    DFS(0);

    return 0;
}
```