Name:-Dikesh Ganboi
Roll NO:- A36

```cpp
#include <iostream>
#include <vector>
#include <queue>
#include <stack>

using namespace std;

// Depth-First Search (DFS)
void DFS(vector<vector<int>>& graph, int start) {
    vector<bool> visited(graph.size(), false);
    stack<int> stk;
    stk.push(start);

    while (!stk.empty()) {
        int current = stk.top();
        stk.pop();

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

            for (int neighbor : graph[current]) {
                if (!visited[neighbor]) {
                    stk.push(neighbor);
                }
            }
        }
    }
}

// Breadth-First Search (BFS)
void BFS(vector<vector<int>>& graph, int start) {
    vector<bool> visited(graph.size(), false);
    queue<int> que;
    que.push(start);
    visited[start] = true;

    while (!que.empty()) {
        int current = que.front();
        que.pop();
        cout << current << " ";
        for (int neighbor : graph[current]) {
            if (!visited[neighbor]) {
                visited[neighbor] = true;
                que.push(neighbor);
            }
        }
    }
}

int main() {
    // Sample graph represented as an adjacency list
    vector<vector<int>> graph = {
        {1, 2},     // 0
        {0, 3, 4},  // 1
        {0, 3, 5},  // 2
        {1, 2, 4},  // 3
        {1, 3},     // 4
        {2}         // 5
    };

    cout << "DFS traversal starting from vertex 0: ";
    DFS(graph, 0);
    cout << endl;

    cout << "BFS traversal starting from vertex 0: ";
    BFS(graph, 0);
    cout << endl;

    return 0;
}
```

OUTPUT:-

```
PS C:\Users\HP\Desktop\DAA EXperiment> cd "c:\Users\HP\Desktop\DAA
EXperiment\" ; if ($?) { g++ graphtraversing.cpp -o graphtraversing
} ; if ($?) { .\graphtraversing }

DFS traversal starting from vertex 0: 0 2 5 3 4 1
BFS traversal starting from vertex 0: 0 1 2 3 4 5
```