**Name:** Rohan Arun Nalawade

**Roll No:** 231012

**PRN:** 22310407

**SY IT A1**

**Assignment 6(B)**

**Code Implementation:**

#include <iostream>

#include <vector>

using namespace std;

#define MAX 999

struct edgeList

{

    int u, v, wt;

};

class bellmanFord

{

public:

    int V, E;

    vector<edgeList> edges;

    void inputGraph()

    {

        cout << "Enter number of vertices and edges: ";

        cin >> V >> E;

        edges.resize(E);

        cout << "Enter edges (u v weight):" << endl;

        for (int i = 0; i < E; i++)

        {

            cin >> edges[i].u >> edges[i].v >> edges[i].wt;

        }

    }

    vector<int> bellman\_ford(int S)

    {

        vector<int> dist(V, MAX);

        dist[S] = 0;

        for (int i = 0; i < V - 1; i++)

        {

            for (auto edge : edges)

            {

                int u = edge.u;

                int v = edge.v;

                int wt = edge.wt;

                if (dist[u] != MAX && dist[u] + wt < dist[v])

                {

                    dist[v] = dist[u] + wt;

                }

            }

        }

        for (auto edge : edges)

        {

            int u = edge.u;

            int v = edge.v;

            int wt = edge.wt;

            if (dist[u] != MAX && dist[u] + wt < dist[v])

            {

                cout << "Negative weight cycle detected." << endl;

                return {-1};

            }

        }

        return dist;

    }

    void printResult(vector<int> dist)

    {

        if (dist[0] == -1)

            return; // -ve cycle case

        cout << "Vertex   Distance from Source" << endl;

        for (int i = 0; i < V; i++)

        {

            cout << i << "\t";

            if (dist[i] == MAX)

            {

                cout << "INF";

            }

            else

            {

                cout << dist[i];

            }

            cout << endl;

        }

    }

};

int main()

{

    bellmanFord graph;

    graph.inputGraph();

    int source;

    cout << "Enter source vertex: ";

    cin >> source;

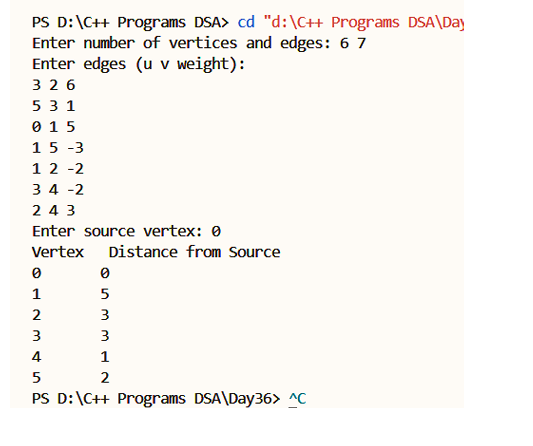
    vector<int> result = graph.bellman\_ford(source);

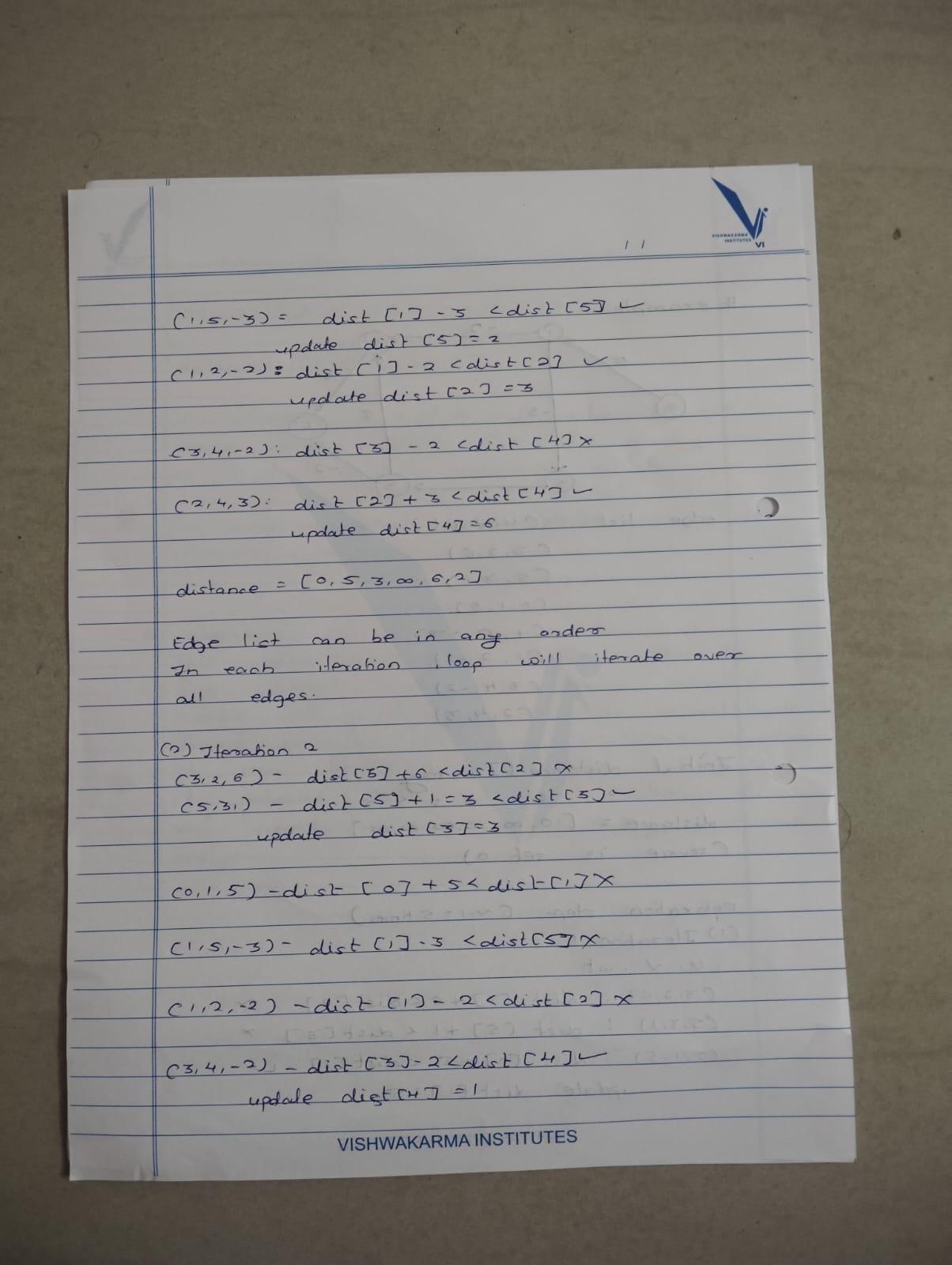
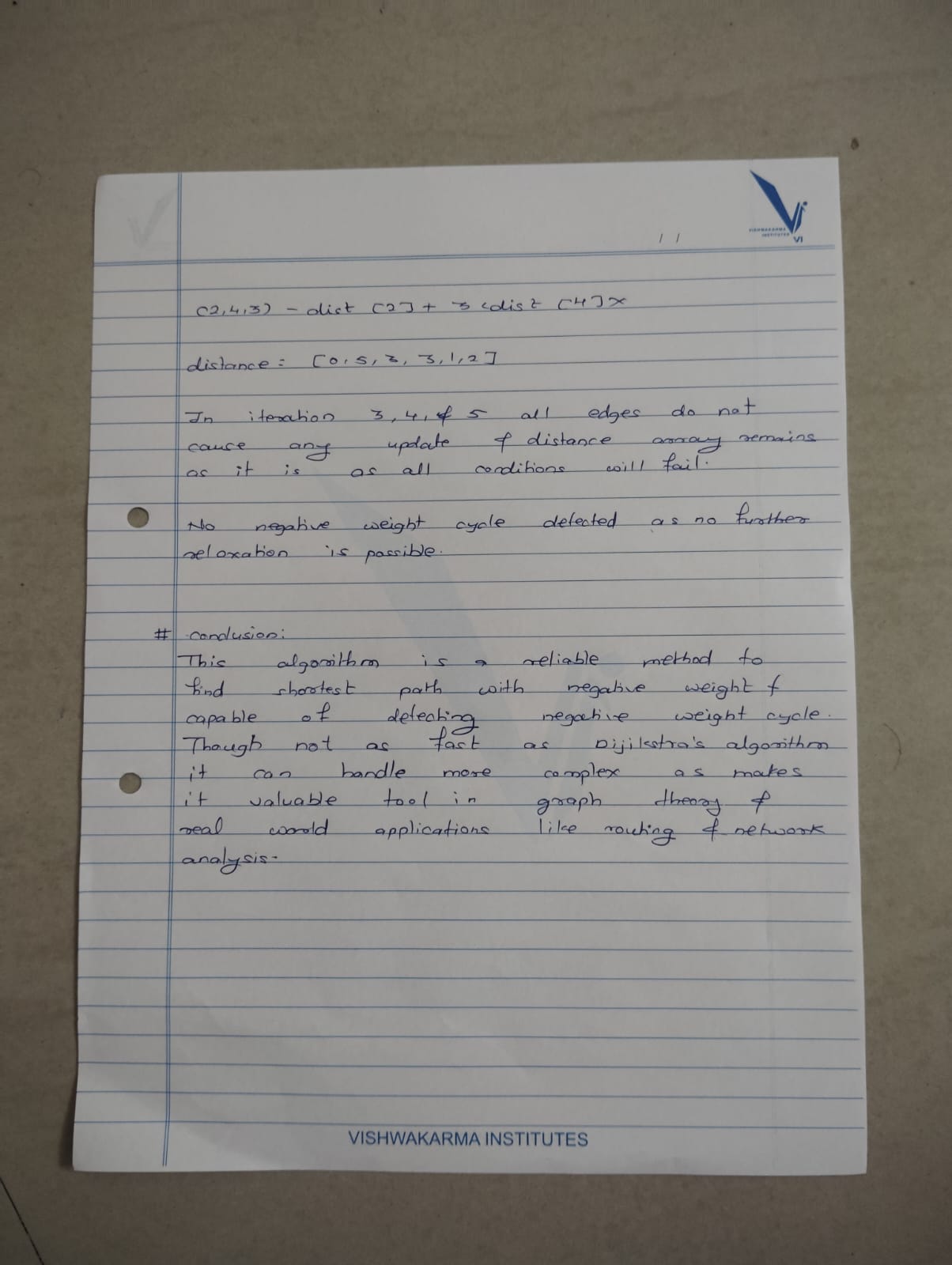
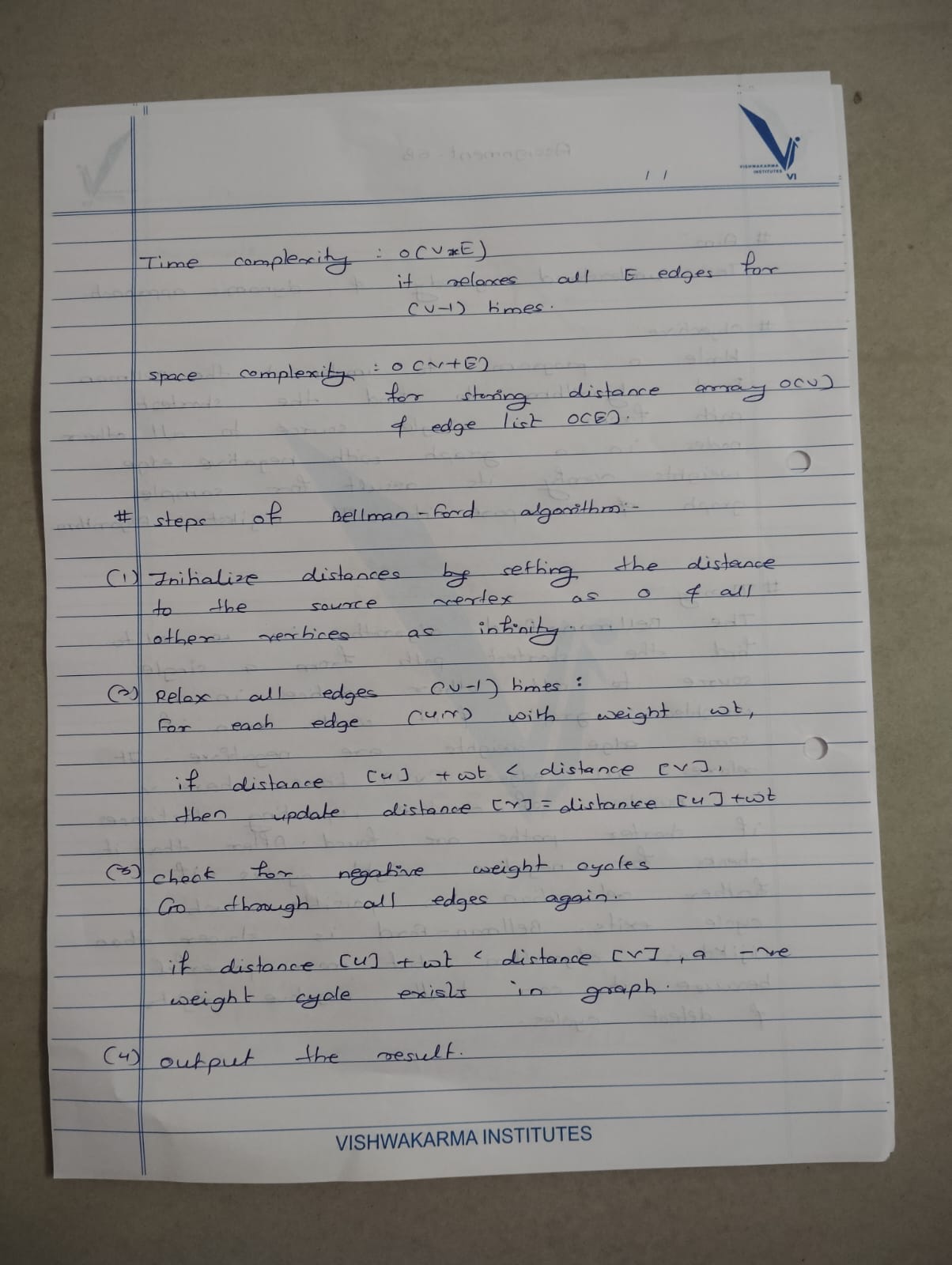
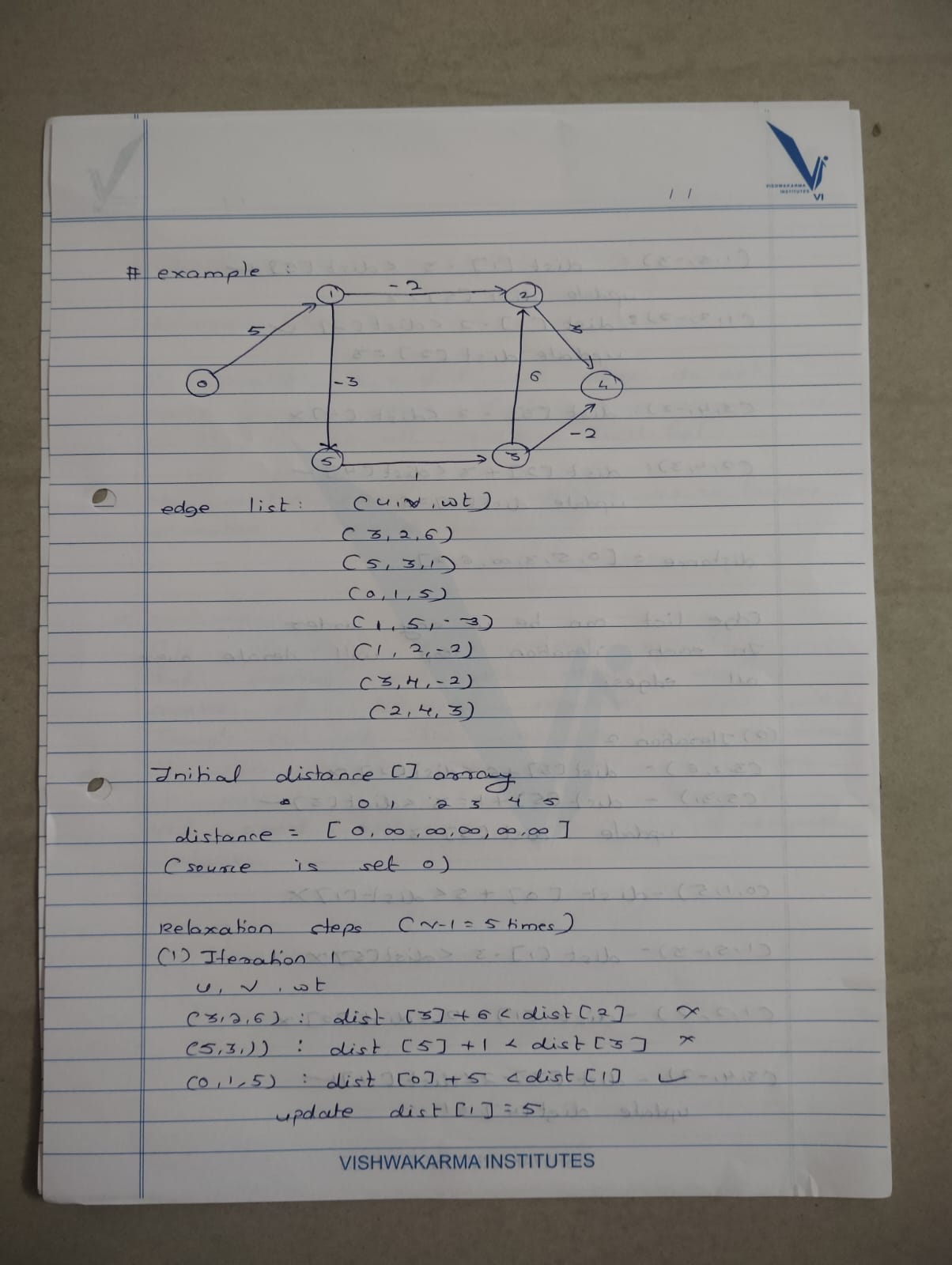
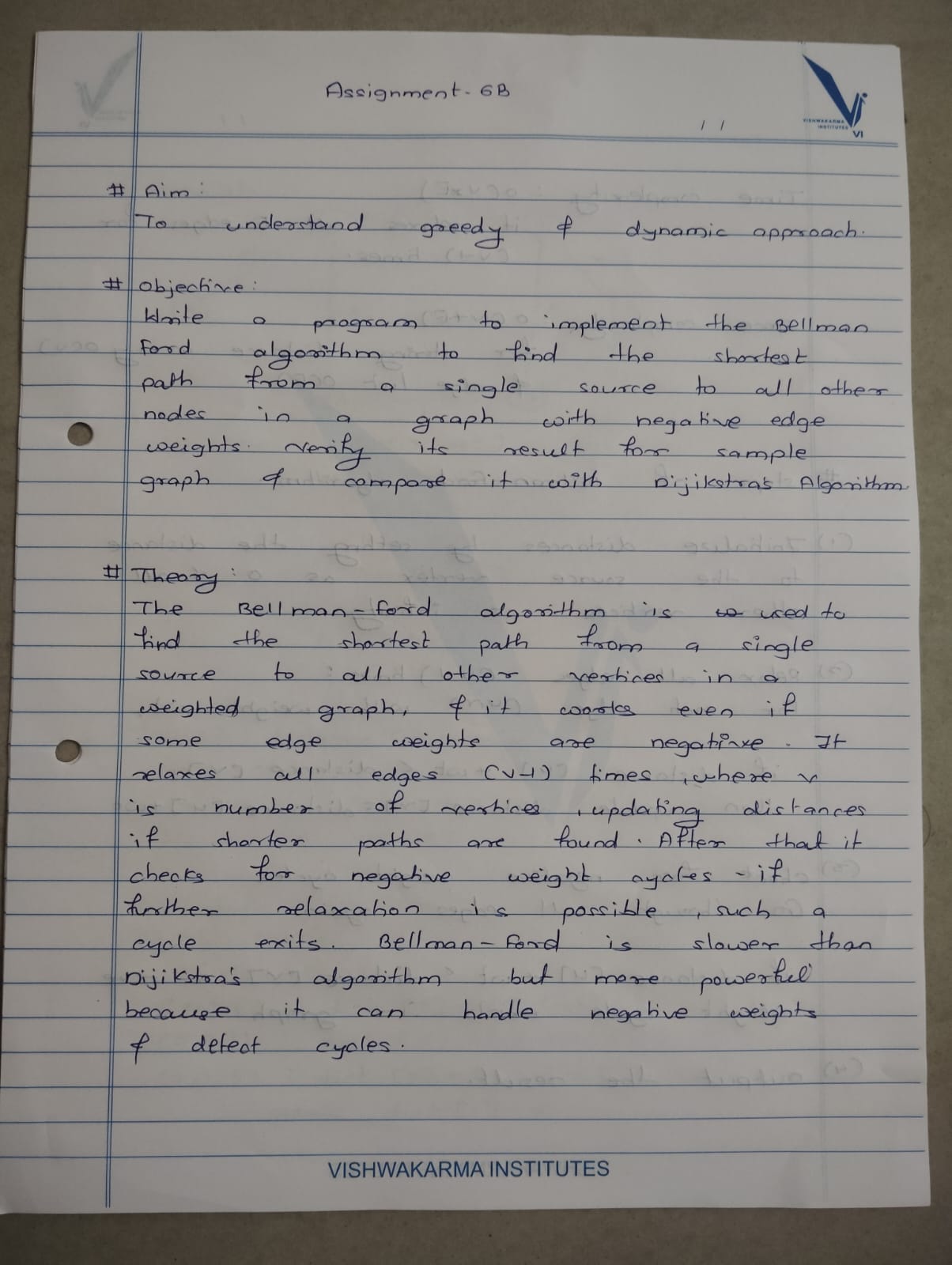
    graph.printResult(result);

    return 0;

}

**Output:**

****

****