Bellman-Ford-Adjacency-Matrix

```
#include <bits/stdc++.h>
using namespace std;

void BellmanFord(int graph[][3], int V, int E, int src){
    int dis[V];
    for (int i = 0; i < V; i++)
        dis[i] = INT_MAX;

    dis[src] = 0;

    for (int i = 0; i < V - 1; i++) {
        for (int j = 0; j < E; j++) {
            if (dis[graph[j][0]] + graph[j][2] < dis[graph[j][1]])
                dis[graph[j][1]] = dis[graph[j][0]] + graph[j][2];
        }
    }

    for (int i = 0; i < E; i++) {
        int x = graph[i][0];
        int y = graph[i][1];
        int weight = graph[i][2];
        if (dis[x] != INT_MAX && dis[x] + weight < dis[y])
            cout << "Graph contains negative weight cycle" << endl;
    }

    cout << "Vertex Distance from Source" << endl;
    for (int i = 0; i < V; i++)
        cout << i << "\t\t" << dis[i] << endl;
}

int main(){
    int V = 5;
    int E = 8;

    int graph[][3] = { { 0, 1, -1 }, { 0, 2, 4 },
                       { 1, 2, 3 }, { 1, 3, 2 },
                       { 1, 4, 2 }, { 3, 2, 5 },
                       { 3, 1, 1 }, { 4, 3, -3 } };

    BellmanFord(graph, V, E, 0);
    return 0;
}
```