

bd.c

```

#include <stdio.h>
#include <limits.h>
#define MAX_VERTICES 100
#define MAX_EDGES 100
void BellmanFord(int graph[MAX_EDGES][3], int V, int E, int src)
{
    int dis[MAX_VERTICES];
    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]] != INT_MAX && 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])
            printf("Graph contains negative weight cycle");
    }
    printf("Vertex Distance from Source\n");
    for (int i = 0; i < V; i++)
    {
        if(i==0)printf("s\t\t%d\n", dis[i]);
        else if(i==1)printf("t\t\t%d\n", dis[i]);
        else if(i==2)printf("x\t\t%d\n", dis[i]);
        else if(i==3)printf("z\t\t%d\n", dis[i]);
        else if(i==4)printf("y\t\t%d\n", dis[i]);
    }
    //printf("%d\t\t%d\n", i, dis[i]);
}
int main()
{
    int V = 5;
    int E = 9;
    int graph[MAX_EDGES][3] = { {0,1,6},{1,4,8},{0,4,7},{4,3,9},{3,2,7},{2,1,-2},{4,2,-3},
{1,3,-4},{3,0,2} };
    BellmanFord(graph, V, E, 0);
    return 0;
}

```