#include<stdio.h>

#include<stdlib.h>

typedef struct graph

{

int v1;

int v2;

int weight;

}graph;

graph g[20];

int dist[10];

int source;

int v;

int e;

int relax(int u,int v,int w)

{

if(dist[v] > dist[u] + w)

{

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

}

}

int bellman\_ford()

{

int i,j;

for(i = 0;i < v;i++)

{

dist[i] = 99999;

}

dist[source] = 0;

for(i = 1;i <= v-1;i++)

{

for(j = 0; j < e;j++)

{

relax(g[j].v1,g[j].v2,g[j].weight);

}

}

for(i = 0; i < e;i++)

{

if(dist[g[i].v2] > dist[g[i].v1] + g[i].weight)

{

return -1;

}

}

return 0;

}

int main()

{

int i,j;

printf("Enter the vertices in your graph\n");

scanf("%d",&v);

printf("Enter the no of edges\n");

scanf("%d",&e);

for(i = 0;i < e;i++)

{

printf("Enter the source of edge\n");

scanf("%d",&g[i].v1);

printf("Enter the destination of edge\n");

scanf("%d",&g[i].v2);

printf("Enter the weight of edge\n");

scanf("%d",&g[i].weight);

}

printf("Enter source\n");

scanf("%d",&source);

int result = bellman\_ford();

if(result == -1)

{

printf("Graph has a negative edge cycle.Therefore no solution\n");

}

else

{

printf("Solution is - \n");

for(i = 0;i < v;i++)

{

printf("Distance of %d from source - %d\n",i,dist[i]);

}

}

}

OUTPUT:



