

```

1: #include<stdio.h>
2: #include<stdlib.h>
3: #include<limits.h>
4:
5: int min(int a,int b){
6:     if(a<b)return a;
7:
8:     return b;
9:
10: }
11: struct graph* gr=NULL;
12: struct node{
13:     int dest;
14:     int weight;
15:     struct node* next;
16: };
17:
18: struct list{
19:     struct node* head;
20:
21: };
22:
23: struct graph{
24:     int v;
25:
26:     struct list* array;
27: };
28:
29: struct node* cnode(int dest,int w){
30:     struct node* temp=(struct node*)malloc(sizeof(struct node));
31:     temp->dest=dest;
32:     temp->weight=w;
33:     temp->next=NULL;
34:
35:     return(temp);
36: }
37:
38: struct graph* cgraph(int v){
39:     struct graph* gr=(struct graph*)malloc(sizeof(struct graph));
40:     gr->v=v;
41:     gr->array=(struct list*)malloc(v*sizeof(struct list));
42:     for(int i=0;i<v;++i){
43:         gr->array[i].head=NULL;
44:     }
45:     return(gr);
46: }
47: void addedge(struct graph* gr,int src,int weight,int dest){
48:     struct node* temp=cnode(dest,weight);
49:     temp->next=gr->array[src].head;
50:     gr->array[src].head=temp;
51:     printf("ADDED EDGE:: %d---(%d)--->%d\n",src,weight,dest);
52:     temp=NULL;
53:     delete(temp);
54: }

```

```

55:
56: void bellman(int src){
57:     int dist[gr->v];
58:     for(int i=0;i<gr->v;++i)
59:         dist[i]=INT_MAX;
60:     dist[src]=0;
61:     for(int j=0;j<gr->v;++j){
62:         for(int i=0;i<gr->v;++i){
63:             if(dist[i]!=INT_MAX){
64:                 struct node* temp=gr->array[i].head;
65:                 while(temp!=NULL){
66:                     int j=temp->dest;
67:                     int w=temp->weight;
68:                     dist[j]=min(dist[j],dist[i]+w);
69:
70:                     temp=temp->next;
71:                 }
72:             }
73:         }
74:     }
75: }
76: printf("::::CALCULATING SHORTEST PATH IN NEGATIVE WEIGHT EDGE
GRAPH::::\n");
77: for(int i=0;i<gr->v;++i){
78:     if(dist[i]==INT_MAX)
79:         printf("\n|%d|---->-----|INFINITE|---->-----|%d|",src+1,
i+1);
80:     else
81:         printf("\n|%d|---->---|<div>|---->---|</div>|",src+1,dist[i],i+1);
82: }
83: printf("\n-----\n");
84: }
85: int main(){
86:     gr=cgraph(8);
87:     addedge(gr,0,10,1);
88:     addedge(gr,0,8,7);
89:     addedge(gr,1,2,5);
90:     addedge(gr,2,1,1);
91:     addedge(gr,2,1,3);
92:     addedge(gr,3,3,4);
93:     addedge(gr,4,-1,5);
94:     addedge(gr,5,-2,2);
95:     addedge(gr,6,-1,5);
96:     addedge(gr,6,-4,1);
97:     addedge(gr,7,1,6);
98:     bellman(1);
99:
100: }
```