**BELLMAN FORD PROGRAM**

#include<stdio.h>

int A[10][10], n, d[10], p[10];

voidBellmanFord(int s){

inti,u,v;

for(i=1;i<n;i++){

for(u=0;u<n;u++){

for(v=0;v<n;v++){

if(d[v] > d[u] + A[u][v]){

d[v] = d[u] + A[u][v];

p[v] = u;

}

}

}

}

for(u=0;u<n;u++){

for(v=0;v<n;v++){

if(d[v] > d[u] + A[u][v]){

printf("Negative Edge");

}

}

}

}

int main(){

printf("Enter the no. of vertices : ");

scanf("%d",&n);

printf("Enter the adjacency matrix\n");

inti,j;

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

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

scanf("%d",&A[i][j]);

int source;

for(source=0;source<n;source++){

for(i=0;i<n;i++){

d[i] = 999;

p[i] = -1;

}

d[source] = 0;

BellmanFord(source);

printf("Router %d\n",source);

for(i=0;i<n;i++){

if(i != source){

j = i;

while(p[j] != -1){

printf("%d <- ",j);

j = p[j];

}

}

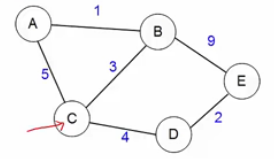
printf("%d\tCost %d\n\n",source,d[i]);

}

}

return 0;

}



**OUTPUT**

exam@dell:~$ gccdvr.c

exam@dell:~$ ./a.out

Enter the no. of vertices : 5

Enter the adjacency matrix

0 1 5 999 999

1 0 3 999 9

5 3 0 4 999

999 999 4 0 2

999 9 999 2 0

Router 0

0 Cost 0

1 <- 0 Cost 1

2 <- 1 <- 0 Cost 4

3 <- 2 <- 1 <- 0 Cost 8

4 <- 1 <- 0 Cost 10

Router 1

0 <- 1 Cost 1

1 Cost 0

2 <- 1 Cost 3

3 <- 2 <- 1 Cost 7

4 <- 1 Cost 9

Router 2

0 <- 1 <- 2 Cost 4

1 <- 2 Cost 3

2 Cost 0

3 <- 2 Cost 4

4 <- 3 <- 2 Cost 6

Router 3

0 <- 1 <- 2 <- 3 Cost 8

1 <- 2 <- 3 Cost 7

2 <- 3 Cost 4

3 Cost 0

4 <- 3 Cost 2

Router 4

0 <- 1 <- 4 Cost 10

1 <- 4 Cost 9

2 <- 3 <- 4 Cost 6

3 <- 4 Cost 2

4 Cost 0