```
#include<stdio.h>
int ballman_ford(int G[20][20],int V,int E,int edge[20][2])
{
	int i,u,v,k,distance[20],parent[20],S,flag=1;
	for(i=0;i<=V;i++)
	{
		distance[i]=1000,parent[i]=-1;
	}
	printf("enter source:");
	scanf("%d",&S);
	distance[S-1]=0;

	for(i=0;i<V-1;i++)
	{
		for(k=0;k<E;k++)
		{
			u=edge[k][0],v=edge[k][1];
			if(distance[u]+G[u][v]<distance[v])
			distance[v]=distance[u]+G[u][v],parent[v]=u;
		}
	}

	for(k=0;k<E;k++)
	{
		u=edge[k][0],v=edge[k][1];
		if(distance[u]+G[u][v]<distance[v])
		flag=0;
	}
```

```
        if(flag)
        for(i=0;i<=V-1;i++)
        printf("vertex %d-> cost= %d parent=%d\n",i+1,distance[i],parent[i]+1);
        return flag;
}

int main()
{
        int V,edge[20][2],G[20][20],i,j,k=0;
        printf("bellman ford \n");
        printf("enter no of vertices");
        scanf("%d",&V);
        printf("enter graph in matrix form: \n");
        for(i=0;i<V;i++)
        for(j=0;j<V;j++)
        {
        scanf("%d",&G[i][j]);
        if(G[i][j]!=0)
        edge[k][0]=i,edge[k++][1]=j;
        }

        if(ballman_ford(G,V,k,edge))
        printf("\n no negative weigth cycle");
        else
        printf("\n negative weight cycle");
        return 0;
}
```

/*

stud@stud-OptiPlex-380:~$ gcc assg3.c

stud@stud-OptiPlex-380:~$ ./a.out

bellman ford

enter no of vertices3

enter graph in matrix form:

2

5

3

1

8

6

7

9

1

enter source:2

vertex 1-> cost= 1 parent=2

vertex 2-> cost= 0 parent=0

vertex 3-> cost= 4 parent=1

no negative weigth cyclestud@stud-OptiPlex-380:~$ gcc assg3.c

stud@stud-OptiPlex-380:~$ ./a.out

bellman ford

enter no of vertices3

enter graph in matrix form:

3

-2

5

-1

8

9

-6

7

4

enter source:3

negative weight cyclestud@stud-OptiPlex-380:~$

*/