Lab 8

Bellmann Ford Alogrithm

Aim : Find suitable path for transmission using Bellmann ford algorithm.

```
#include <stdio.h>
#include < stdlib.h>

int Bellmon Ford (int G[20][20] , int v, int E, int edge
  [20] [20]) {
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 size ");
      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; i++)
      printf (" Vector %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 (" 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][1] = j ;

     }
```

```
if (Bellman-ford(G, V, k, edge))
    printf ("\n No negative weight cycle \n");
else
    printf ("\n Negative weight cycle exists \n");
    return 0;
}
```

Output

```
Enter no. of vertices : 5
Enter graph in matrix:
0  6   0   7   0
0  0   5   8  -4
0 -2   0   0   0
0  0  -3
```