```
Bellmann Fold algun: Program for distance vector algun to
  find suitable path to transmission
#include < stoio. h>
#include < stdlib. h7

#include < stdlib. h7

int Gellman Ford (int 6 [20][20], int V, int E, int edge
int i, u, v, k, distance [20], parent [20], S, Hag=1;
 for (i=0; i< V; i++) ([[] ([] ([] )] ) tool }
  distante [i]= 1000,
                              ( = : [[][]]) };
   Parent [i] = -1; = [i] = [i] ; do i= [o] [i] ) ; do i= [o] [i] selo
    scart ("/,d", & s); ((splas x, V, D)) book mentes)
   (sellness total (6, 1, 2) a) motor (5) = (1-2) a) motor (6) (1-1) (1-V) i i (0=i) 8)
     to ( K=0; KCE; K++) [
        u.edge[k][0], v=edge[k][1];
      if (distance [u] + dis G[u][v] c distance [v])
          distance [v] = distance [u)+ G[u][v];
           parent [v] = u;
                             no of vortice: 5
   to ( k = 0; k < E) k++){ xidtony N
      u = edge [ k][0], v = edge [ k][1];
     if (distance [u] +6, [u][v] < distance [v])
         flag = 0;
   if (blag)
    60 (i=0; i < V; i++)
                             -> 68t = 1/d parent = 1/d/n
       Point f (" vertex 1.d
         i+1, distance [i], parent [i]+1);
      Veturn blag;
```

told affect pastion by distance nects after int main(){

int V, edge [20][2], G[20][20], i, j, K = 0;

int V, edge (20][2], G[20][20], iiii Pf ("Enter no. of vertices:"); Sant ("1.d", LV);

Pf ("Enter graph in matrix form: \"); k, distante [20], possent (++1; 1) i ;0=i) & 18 (j=0; j<V; j++) {

if scant ("/d", l6[i][j]); if (6(i)[j]!=0) if (6[i][i]:=i] edge [k+i][i]=j:
edge [k][o]=i, edge [k+i][i]=j: 3
if (Bellman-tord (G, V, k, edge))

Printf ("InNo regetive weight Gycle In"); e print f ("In Negotive weight Cycle exists In")
return 0; getwer of worse of England sib + [N] anotais passent [v] = u, Enter up. of vestices: 5 Enter graph in matsix: 3(++)000

```
Bellmann:
olp: Enter no. of souters ((10): $6
Enter 99 if Corresponding nonter is not agricult to souter A.
BCOEF
Entos matrix; 4 5 99 99 99
Enter 99 if Corresponding souter is not adjacent to B:
Enter 99 if corresponding souter is not adjacent to C:
Enter 99 if corresponding router is not adjacent to D:
Enter motor: 99 9 99 13 2
Enter 99 if corresponding router is not sejacent to E:
Enter matrix: 49 7 3 13 2 Marcha to
Enter 99 it corresponding souter is not agacent to E:
ABLDE
Enter montrix: 99 99 99 2 6
Routes Table entries for A:
Destination forutes: A B C D E F
Outgoing Line: A B C D E F
Hol Count : 0 4 5 99 99 99
Router Table entries '68 B:
 Destination Pouter: ABCOEF
 Outgoing Line: ABCD EF
 Hop count : 4 0 11 9 7 98
 Router Table entries to C:
 Destination Routes: A B C D E F
Outgoing Line: A B CDEF
                        0 99 3 99
               : 5 11
HOP count
Router Table entries for D:
Destination Reuter: A B C D E F
Ortgoing Line: A B C D E F
                        99 0 13 2
Hop word
           : 98 9
```

Pautes Table entries (58 houter E;

Destruction Routers: A B C D E F

Outgoing Cine: A B C D E F

Hof Count; 99 7 3 13 0 2

Rout Eable entries for routers F:

Pout bable entries for routers F:

Destruction Routers: 99 99 99 2 6 0