int G[N][N]=

{

{0,1,1,0},

{0,0,1,1},

{0,0,0,1},

{0,0,0,0},

};

struct vzaim

{

int D;

int T;

};

void WriteGraph()

{

int i,j;

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

{

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

printf("%i ", G[i][j]);

printf("\n");

}

printf("\n");

}

void Dijkstra(int start, struct vzaim \*M)

{

int minn, index,i;

int\* V;

V = (int\*)malloc(N\*sizeof(int));

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

{

V[i] = 0;

M[i].D = INT\_MAX;

M[i].T = -1;

}

M[start].T = 0;

M[start].D = 0;

index=start;

int f=1;

while (f)

{

f=0;

minn = INT\_MAX;

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

if((V[i] == 0) && (M[i].D <= minn))

{

minn = M[i].D;

index = i;

}

V[index] = 1;

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

if((V[i] == 0) && (G[index][i] != 0) && (M[index].D != INT\_MAX) && ((M[index].D + G[index][i]) < M[i].D))

{

M[i].D = M[index].D + G[index][i];

M[i].T = index;

f=1;

}

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

if (!V[i]) f=1;

}

}

int main()

{

int i,j;

struct vzaim M[N][N];

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

{

Dijkstra(i,M[i]);

}

int f1=1,f2=1;

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

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

if ((M[i][j].D==INT\_MAX&&M[j][i].D!=INT\_MAX)||(M[i][j].D!=INT\_MAX&&M[j][i].D==INT\_MAX)) f1=0;

else if (!(M[i][j].D!=INT\_MAX&&M[j][i].D!=INT\_MAX)) f1=f2=0;

if (f1)printf("Silno\n");

if(f2)printf("Odno\n");

return 0;

}