```cpp
/** 125 - Numbering Paths:
Given the Number of roads m. Then the next m lines, each will contain two integer u and v,
indicating there is a one way road from city u to city v. You have to determine the number
of different routes between each city to all the city. City are numbered from 0
to largest all max(u,v). Consider four city are connected by the following one-way streets:
        0-->1, 0-->2, 1-->2, 2-->3.
There is one route from intersection 0 to 1, two routes from 0 to 2 (the routes are 0-->1-->2
and 0-->2), one route from 0 to 3, one route from 1 to 2, one route from 1 to 3, one route from
 2 to 3, and no other routes. It is possible that an infinite number of different routes
 to exist for two cities. For example if we added a road 3-->2 in the list, then there is
 still only one route from 0 to 1, but there are infinitely many different routes from 0 to 2.
 Because the street from 2 to 3 and back to 2 can be repeated yielding a different sequence of
 streets and hence a different route. Thus the route 0-->2-->3-->2-->3-->2 is a different
 route than 0-->2-->3-->2.
 For each test case, a square matrix of the number of different routes from city j to city k
 is printed. If the matrix is denoted M, then M[j][k] is the number of different routes
 from city j to city k. If there are an infinite number of different paths between
 two city a ‘-1’ should be printed.
***/
vector<int>ed[105]; int n,m,way[105][105];
void floydWarshall(){
    for(int k=0; k<=n; k++){
        for(int i=0; i<=n; i++){
            for(int j=0; j<=n; j++){
                if(way[i][k]!=0 && way[k][j]!=0){
                    way[i][j] += way[i][k]*way[k][j];
                }
            }
        }
    }

    for(int k=0; k<=n; k++){
        if(way[k][k]!=0) { // have a cycle
            for(int i=0; i<=n; i++){
                for(int j=0; j<=n; j++){
                    if(way[i][k]!=0 && way[k][j]!=0) {
                        way[i][j] = -1;
                    }
                }
            }
        }
    }
}
int main(){
    int ks=0;
    while(scanf("%d",&m)==1){
        n=0; memset(way,0,sizeof(way));
        for(int i=1; i<=m; i++){
            int u,v; scanf("%d%d",&u,&v);
            n = max(n,max(u,v));
            ed[u].push_back(v);
            way[u][v]=1;
        }

        floydWarshall();

        printf("matrix for city %d\n",ks++);
        for(int i=0; i<=n; i++){
            for(int j=0; j<=n; j++){
                if(j==0)printf("%d",way[i][j]);
                else printf(" %d",way[i][j]);
            }
            printf("\n");
        }
        for(int i=0; i<=n; i++)ed[i].clear();
    }
}
```