Name: Shikhar Sahu

Roll No: 74

Sec: A

Semester: 5

Shift: 1

Explanation and Thoughts:

I've used the **backtracking + recursion + DFS** approach for this question.

We apply depth first search first for each node to all the nodes that are connected to that node.

We keep a visited hash map. It keeps track of all the nodes that we have traversed.

We also keep a Path vector. It stores the path of the cycle.

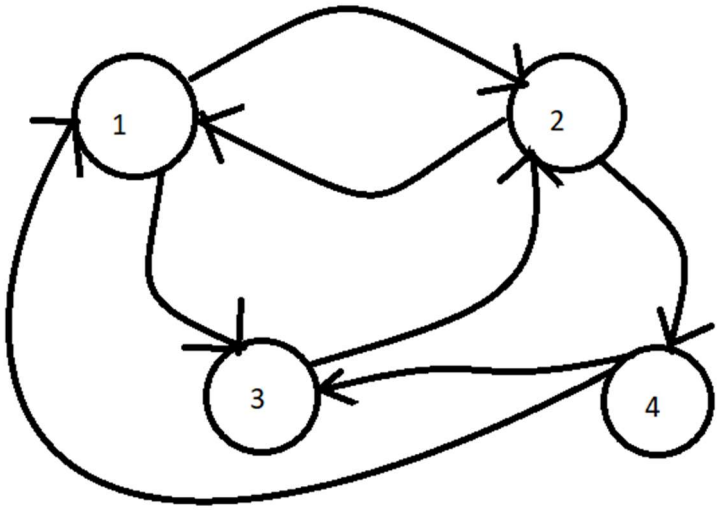
As we go deeper into the recursion tree, we push elements to the back of the path vector and insert elements to the visited set.

As we come out of the recursion tree, we pop elements back from the path vector and also erase that element from the visited set.

In the recursion, we check if there is possibility to return to the initial node, if it possible, then the graph is a partial cyclic graph.

If the number of elements in the partial cyclic graph is equal to the number of elements in graph + 1, then the cycle is a complete cycle.

Now, for the discussion of test cases: I've used 5 sample test cases explained in code.
I'll take an example test case:
For n = 4.



Output:

```
Number of complete of complete cycles: 4

complete path
1  ->  3  ->  2  ->  4  ->  1
2  ->  4  ->  1  ->  3  ->  2
3  ->  2  ->  4  ->  1  ->  3
4  ->  1  ->  3  ->  2  ->  4

Number of complete of partial cycles: 9
partial path
1  ->  2  ->  4  ->  1
1  ->  3  ->  2  ->  1
2  ->  1  ->  3  ->  2
2  ->  4  ->  1  ->  2
2  ->  4  ->  3  ->  2
3  ->  2  ->  1  ->  3
3  ->  2  ->  4  ->  3
4  ->  1  ->  2  ->  4
4  ->  3  ->  2  ->  4
```

Code:

```
/*
@author: ShikharSahu
*/

#include <iostream>
#include <string>
```

```cpp
#include <vector>
#include <algorithm>
#include <sstream>
#include <queue>
#include <deque>
#include <bitset>
#include <iterator>
#include <list>
#include <stack>
#include <map>
#include <set>
#include <unordered_map>
#include <unordered_set>
#include <functional>
#include <numeric>
#include <utility>
#include <limits>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <assert.h>
typedef long long int ll;
using namespace std;
#define MOD 1e9+7;

set<vector<int>> complete, partial;

void printPath(vector<int> path){
    path.pop_back();
    for (int e : path){
        cout << e << "  ->  ";
    }
    cout << path.front();
    cout << endl ;
}
void helper (int current, vector<int>& allConnected, vector<pair<int,vector<in
t> > >& graph, int initial, unordered_set<int>& visited, vector<int>& path){
    if (path.size()>=2){
        if( find(allConnected.begin(), allConnected.end(), initial) != allConn
ected.end() ){
            if( path.size() == graph.size() ){
                path.push_back(initial);
                complete.insert(path);
                path.pop_back();
                // printPath(path);
            // print path
```

```
            }
            path.push_back(initial);
            partial.insert(path);
            // printPath(path);
            path.pop_back();
        // add to set of paths
        }
    }
    for(int element : allConnected){
        if(visited.count(element)==0){
            visited.insert(element);
            path.push_back(element);
            helper(element, graph[element-
1].second, graph, initial, visited, path);
            path.pop_back();
            visited.erase(element);
        }
    }
}
void printMatrix(vector<vector<bool> > graph){
    int n = graph.size();
    for (int i =0 ; i< n; i++){
        for (int j = 0 ; j < n ; j ++){
            cout << (int) graph[i][j] << "   ";
        }
        cout << endl;
    }
}
vector<vector<bool> > getMatrix(int n){
    vector<vector<bool> > graph(n,vector<bool>(n,false));
    for (int i =0 ; i< n; i++){
        for (int j = 0 ; j < n ; j ++){
            if(i!=j){
                if(rand()%3 == 0){
                    graph[i][j] = true;
                }
            }
        }
    }
    return graph;
}
vector<pair<int,vector<int> > > getList(vector<vector<bool> > mat){
    vector<pair<int,vector<int> > > graph;
    for (int i = 0 ; i < mat.size(); i++){
        vector<int> temp;
        for(int j = 0; j <mat.size(); j++){
            if(mat[i][j]){
                temp.push_back(j+1);
```

```
            }
        }
        graph.push_back(make_pair(i+1,temp));
    }
    return graph;
}

void solve(){
    // vector<pair<int,vector<int> > > graph(4);
    // graph[0] = {1,{2,3}};
    // graph[1] = {2,{1,4}};
    // graph[2] = {3,{1,2}};
    // graph[3] = {4,{1,3}};
    srand(time(0));
    int size = 5 + rand() % 3;
    auto mat  = getMatrix(size);
    cout << "graph is :\n";
    printMatrix(mat);
    auto graph = getList(mat);

    vector<int> path;
    unordered_set<int> visited;
    for (auto node: graph){

        visited.insert(node.first);
        path.push_back(node.first);
        helper(node.first, node.second, graph, node.first, visited, path);
        path.pop_back();
        visited.erase(node.first);
    }

    cout << "Number of complete of complete cycles: " << complete.size() <<end
l;
    cout << "complete path\n";
    for (auto path : complete){
        printPath(path);
    }

    cout << endl;
    set<vector<int>> perfectPartials;
    for (auto path : partial){
        if(path.size()>3){
            perfectPartials.insert(path);
        }
    }
    cout << "Number of complete of partial cycles: " <<  perfectPartials.size(
) <<endl;
    cout << "partial path\n";
```

```
    for (auto path : perfectPartials){
        printPath(path);
    }

}



int main(){
    ll T=5;
    while(T--){
        solve();
    }
}
```

Output:

```
graph is :
0   1   0   0   1   0
0   0   1   0   1   0
0   0   0   1   1   0
1   1   1   0   0   0
1   1   0   1   0   1
0   0   1   1   0   0
Number of complete of complete cycles: 12
complete path
1  ->  2  ->  3  ->  5  ->  6  ->  4  ->  1
1  ->  2  ->  5  ->  6  ->  3  ->  4  ->  1
2  ->  3  ->  5  ->  6  ->  4  ->  1  ->  2
2  ->  5  ->  6  ->  3  ->  4  ->  1  ->  2
3  ->  4  ->  1  ->  2  ->  5  ->  6  ->  3
3  ->  5  ->  6  ->  4  ->  1  ->  2  ->  3
4  ->  1  ->  2  ->  3  ->  5  ->  6  ->  4
4  ->  1  ->  2  ->  5  ->  6  ->  3  ->  4
5  ->  6  ->  3  ->  4  ->  1  ->  2  ->  5
5  ->  6  ->  4  ->  1  ->  2  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  2  ->  5  ->  6
```

6 -> 4 -> 1 -> 2 -> 3 -> 5 -> 6

Number of complete of partial cycles: 79

partial path

1 -> 2 -> 3 -> 4 -> 1

1 -> 2 -> 3 -> 5 -> 1

1 -> 2 -> 3 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 1

1 -> 2 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 6 -> 4 -> 1

1 -> 5 -> 2 -> 3 -> 4 -> 1

1 -> 5 -> 4 -> 1

1 -> 5 -> 6 -> 3 -> 4 -> 1

1 -> 5 -> 6 -> 4 -> 1

2 -> 3 -> 4 -> 1 -> 2

2 -> 3 -> 4 -> 1 -> 5 -> 2

2 -> 3 -> 4 -> 2

2 -> 3 -> 5 -> 1 -> 2

2 -> 3 -> 5 -> 2

2 -> 3 -> 5 -> 4 -> 1 -> 2

2 -> 3 -> 5 -> 4 -> 2

2 -> 3 -> 5 -> 6 -> 4 -> 2

2 -> 5 -> 1 -> 2

2 -> 5 -> 4 -> 1 -> 2

2 -> 5 -> 4 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 2

2 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 4 -> 2

3 -> 4 -> 1 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 6 -> 3

3 -> 4 -> 2 -> 3

3 -> 4 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 1 -> 2 -> 3

3 -> 5 -> 2 -> 3

3 -> 5 -> 4 -> 1 -> 2 -> 3

3 -> 5 -> 4 -> 2 -> 3

3 -> 5 -> 4 -> 3

3 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 2 -> 3

3 -> 5 -> 6 -> 4 -> 3

4 -> 1 -> 2 -> 3 -> 4

4 -> 1 -> 2 -> 3 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 4

4 -> 1 -> 5 -> 2 -> 3 -> 4

4 -> 1 -> 5 -> 4

4 -> 1 -> 5 -> 6 -> 3 -> 4

4 -> 1 -> 5 -> 6 -> 4

4 -> 2 -> 3 -> 4

4 -> 2 -> 3 -> 5 -> 4

4 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 2 -> 5 -> 4

4 -> 2 -> 5 -> 6 -> 3 -> 4

4 -> 2 -> 5 -> 6 -> 4

4 -> 3 -> 5 -> 4

4 -> 3 -> 5 -> 6 -> 4

5 -> 1 -> 2 -> 3 -> 5

5 -> 1 -> 2 -> 5

5 -> 2 -> 3 -> 4 -> 1 -> 5

5 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 5

5 -> 4 -> 1 -> 5

```
5  ->  4  ->  2  ->  3  ->  5
5  ->  4  ->  2  ->  5
5  ->  4  ->  3  ->  5
5  ->  6  ->  3  ->  4  ->  1  ->  5
5  ->  6  ->  3  ->  4  ->  2  ->  5
5  ->  6  ->  3  ->  5
5  ->  6  ->  4  ->  1  ->  2  ->  5
5  ->  6  ->  4  ->  1  ->  5
5  ->  6  ->  4  ->  2  ->  3  ->  5
5  ->  6  ->  4  ->  2  ->  5
5  ->  6  ->  4  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  5  ->  6
6  ->  3  ->  4  ->  2  ->  5  ->  6
6  ->  3  ->  5  ->  6
6  ->  4  ->  1  ->  2  ->  5  ->  6
6  ->  4  ->  1  ->  5  ->  6
6  ->  4  ->  2  ->  3  ->  5  ->  6
6  ->  4  ->  2  ->  5  ->  6
6  ->  4  ->  3  ->  5  ->  6
graph is :
0   1   0   0   1   0
0   0   1   0   1   0
0   0   0   1   1   0
1   1   1   0   0   0
1   1   0   1   0   1
0   0   1   1   0   0
Number of complete of complete cycles: 12
complete path
1  ->  2  ->  3  ->  5  ->  6  ->  4  ->  1
1  ->  2  ->  5  ->  6  ->  3  ->  4  ->  1
2  ->  3  ->  5  ->  6  ->  4  ->  1  ->  2
2  ->  5  ->  6  ->  3  ->  4  ->  1  ->  2
```

3 -> 4 -> 1 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 1 -> 2 -> 3

4 -> 1 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 3 -> 4

5 -> 6 -> 3 -> 4 -> 1 -> 2 -> 5

5 -> 6 -> 4 -> 1 -> 2 -> 3 -> 5

6 -> 3 -> 4 -> 1 -> 2 -> 5 -> 6

6 -> 4 -> 1 -> 2 -> 3 -> 5 -> 6

Number of complete of partial cycles: 79

partial path

1 -> 2 -> 3 -> 4 -> 1

1 -> 2 -> 3 -> 5 -> 1

1 -> 2 -> 3 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 1

1 -> 2 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 6 -> 4 -> 1

1 -> 5 -> 2 -> 3 -> 4 -> 1

1 -> 5 -> 4 -> 1

1 -> 5 -> 6 -> 3 -> 4 -> 1

1 -> 5 -> 6 -> 4 -> 1

2 -> 3 -> 4 -> 1 -> 2

2 -> 3 -> 4 -> 1 -> 5 -> 2

2 -> 3 -> 4 -> 2

2 -> 3 -> 5 -> 1 -> 2

2 -> 3 -> 5 -> 2

2 -> 3 -> 5 -> 4 -> 1 -> 2

2 -> 3 -> 5 -> 4 -> 2

2 -> 3 -> 5 -> 6 -> 4 -> 2

2 -> 5 -> 1 -> 2

2 -> 5 -> 4 -> 1 -> 2

2 -> 5 -> 4 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 2

2 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 4 -> 2

3 -> 4 -> 1 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 6 -> 3

3 -> 4 -> 2 -> 3

3 -> 4 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 1 -> 2 -> 3

3 -> 5 -> 2 -> 3

3 -> 5 -> 4 -> 1 -> 2 -> 3

3 -> 5 -> 4 -> 2 -> 3

3 -> 5 -> 4 -> 3

3 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 2 -> 3

3 -> 5 -> 6 -> 4 -> 3

4 -> 1 -> 2 -> 3 -> 4

4 -> 1 -> 2 -> 3 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 4

4 -> 1 -> 5 -> 2 -> 3 -> 4

4 -> 1 -> 5 -> 4

4 -> 1 -> 5 -> 6 -> 3 -> 4

4 -> 1 -> 5 -> 6 -> 4

4 -> 2 -> 3 -> 4

4 -> 2 -> 3 -> 5 -> 4

4 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 2 -> 5 -> 4

4 -> 2 -> 5 -> 6 -> 3 -> 4

4 -> 2 -> 5 -> 6 -> 4

4 -> 3 -> 5 -> 4

4 -> 3 -> 5 -> 6 -> 4

```
5  ->  1  ->  2  ->  3  ->  5
5  ->  1  ->  2  ->  5
5  ->  2  ->  3  ->  4  ->  1  ->  5
5  ->  2  ->  3  ->  5
5  ->  4  ->  1  ->  2  ->  3  ->  5
5  ->  4  ->  1  ->  2  ->  5
5  ->  4  ->  1  ->  5
5  ->  4  ->  2  ->  3  ->  5
5  ->  4  ->  2  ->  5
5  ->  4  ->  3  ->  5
5  ->  6  ->  3  ->  4  ->  1  ->  5
5  ->  6  ->  3  ->  4  ->  2  ->  5
5  ->  6  ->  3  ->  5
5  ->  6  ->  4  ->  1  ->  2  ->  5
5  ->  6  ->  4  ->  1  ->  5
5  ->  6  ->  4  ->  2  ->  3  ->  5
5  ->  6  ->  4  ->  2  ->  5
5  ->  6  ->  4  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  5  ->  6
6  ->  3  ->  4  ->  2  ->  5  ->  6
6  ->  3  ->  5  ->  6
6  ->  4  ->  1  ->  2  ->  5  ->  6
6  ->  4  ->  1  ->  5  ->  6
6  ->  4  ->  2  ->  3  ->  5  ->  6
6  ->  4  ->  2  ->  5  ->  6
6  ->  4  ->  3  ->  5  ->  6
graph is :
0   1   0   0   1   0
0   0   1   0   1   0
0   0   0   1   1   0
1   1   1   0   0   0
1   1   0   1   0   1
```

0 0 1 1 0 0

Number of complete of complete cycles: 12

complete path

1 -> 2 -> 3 -> 5 -> 6 -> 4 -> 1

1 -> 2 -> 5 -> 6 -> 3 -> 4 -> 1

2 -> 3 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 1 -> 2

3 -> 4 -> 1 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 1 -> 2 -> 3

4 -> 1 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 3 -> 4

5 -> 6 -> 3 -> 4 -> 1 -> 2 -> 5

5 -> 6 -> 4 -> 1 -> 2 -> 3 -> 5

6 -> 3 -> 4 -> 1 -> 2 -> 5 -> 6

6 -> 4 -> 1 -> 2 -> 3 -> 5 -> 6

Number of complete of partial cycles: 79

partial path

1 -> 2 -> 3 -> 4 -> 1

1 -> 2 -> 3 -> 5 -> 1

1 -> 2 -> 3 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 1

1 -> 2 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 6 -> 4 -> 1

1 -> 5 -> 2 -> 3 -> 4 -> 1

1 -> 5 -> 4 -> 1

1 -> 5 -> 6 -> 3 -> 4 -> 1

1 -> 5 -> 6 -> 4 -> 1

2 -> 3 -> 4 -> 1 -> 2

2 -> 3 -> 4 -> 1 -> 5 -> 2

2 -> 3 -> 4 -> 2

2 -> 3 -> 5 -> 1 -> 2

2 -> 3 -> 5 -> 2

2 -> 3 -> 5 -> 4 -> 1 -> 2

2 -> 3 -> 5 -> 4 -> 2

2 -> 3 -> 5 -> 6 -> 4 -> 2

2 -> 5 -> 1 -> 2

2 -> 5 -> 4 -> 1 -> 2

2 -> 5 -> 4 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 2

2 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 4 -> 2

3 -> 4 -> 1 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 6 -> 3

3 -> 4 -> 2 -> 3

3 -> 4 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 1 -> 2 -> 3

3 -> 5 -> 2 -> 3

3 -> 5 -> 4 -> 1 -> 2 -> 3

3 -> 5 -> 4 -> 2 -> 3

3 -> 5 -> 4 -> 3

3 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 2 -> 3

3 -> 5 -> 6 -> 4 -> 3

4 -> 1 -> 2 -> 3 -> 4

4 -> 1 -> 2 -> 3 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 4

4 -> 1 -> 5 -> 2 -> 3 -> 4

4 -> 1 -> 5 -> 4

4 -> 1 -> 5 -> 6 -> 3 -> 4

4 -> 1 -> 5 -> 6 -> 4

4 -> 2 -> 3 -> 4

4 -> 2 -> 3 -> 5 -> 4

4 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 2 -> 5 -> 4

4 -> 2 -> 5 -> 6 -> 3 -> 4

4 -> 2 -> 5 -> 6 -> 4

4 -> 3 -> 5 -> 4

4 -> 3 -> 5 -> 6 -> 4

5 -> 1 -> 2 -> 3 -> 5

5 -> 1 -> 2 -> 5

5 -> 2 -> 3 -> 4 -> 1 -> 5

5 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 5

5 -> 4 -> 1 -> 5

5 -> 4 -> 2 -> 3 -> 5

5 -> 4 -> 2 -> 5

5 -> 4 -> 3 -> 5

5 -> 6 -> 3 -> 4 -> 1 -> 5

5 -> 6 -> 3 -> 4 -> 2 -> 5

5 -> 6 -> 3 -> 5

5 -> 6 -> 4 -> 1 -> 2 -> 5

5 -> 6 -> 4 -> 1 -> 5

5 -> 6 -> 4 -> 2 -> 3 -> 5

5 -> 6 -> 4 -> 2 -> 5

5 -> 6 -> 4 -> 3 -> 5

6 -> 3 -> 4 -> 1 -> 5 -> 6

6 -> 3 -> 4 -> 2 -> 5 -> 6

6 -> 3 -> 5 -> 6

6 -> 4 -> 1 -> 2 -> 5 -> 6

6 -> 4 -> 1 -> 5 -> 6

6 -> 4 -> 2 -> 3 -> 5 -> 6

6 -> 4 -> 2 -> 5 -> 6

```
6  ->  4  ->  3  ->  5  ->  6
graph is :
0   0   0   0   0   1   0
0   0   1   0   0   0   1
0   0   0   0   0   0   0
0   1   0   0   0   0   1
1   0   1   1   0   1   0
0   0   0   1   0   0   0
0   0   0   0   0   0   0
Number of complete of complete cycles: 12
complete path
1  ->  2  ->  3  ->  5  ->  6  ->  4  ->  1
1  ->  2  ->  5  ->  6  ->  3  ->  4  ->  1
2  ->  3  ->  5  ->  6  ->  4  ->  1  ->  2
2  ->  5  ->  6  ->  3  ->  4  ->  1  ->  2
3  ->  4  ->  1  ->  2  ->  5  ->  6  ->  3
3  ->  5  ->  6  ->  4  ->  1  ->  2  ->  3
4  ->  1  ->  2  ->  3  ->  5  ->  6  ->  4
4  ->  1  ->  2  ->  5  ->  6  ->  3  ->  4
5  ->  6  ->  3  ->  4  ->  1  ->  2  ->  5
5  ->  6  ->  4  ->  1  ->  2  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  2  ->  5  ->  6
6  ->  4  ->  1  ->  2  ->  3  ->  5  ->  6

Number of complete of partial cycles: 79
partial path
1  ->  2  ->  3  ->  4  ->  1
1  ->  2  ->  3  ->  5  ->  1
1  ->  2  ->  3  ->  5  ->  4  ->  1
1  ->  2  ->  5  ->  1
1  ->  2  ->  5  ->  4  ->  1
1  ->  2  ->  5  ->  6  ->  4  ->  1
```

1 -> 5 -> 2 -> 3 -> 4 -> 1

1 -> 5 -> 4 -> 1

1 -> 5 -> 6 -> 3 -> 4 -> 1

1 -> 5 -> 6 -> 4 -> 1

2 -> 3 -> 4 -> 1 -> 2

2 -> 3 -> 4 -> 1 -> 5 -> 2

2 -> 3 -> 4 -> 2

2 -> 3 -> 5 -> 1 -> 2

2 -> 3 -> 5 -> 2

2 -> 3 -> 5 -> 4 -> 1 -> 2

2 -> 3 -> 5 -> 4 -> 2

2 -> 3 -> 5 -> 6 -> 4 -> 2

2 -> 5 -> 1 -> 2

2 -> 5 -> 4 -> 1 -> 2

2 -> 5 -> 4 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 2

2 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 4 -> 2

3 -> 4 -> 1 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 6 -> 3

3 -> 4 -> 2 -> 3

3 -> 4 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 1 -> 2 -> 3

3 -> 5 -> 2 -> 3

3 -> 5 -> 4 -> 1 -> 2 -> 3

3 -> 5 -> 4 -> 2 -> 3

3 -> 5 -> 4 -> 3

3 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 2 -> 3

3 -> 5 -> 6 -> 4 -> 3

4 -> 1 -> 2 -> 3 -> 4

4 -> 1 -> 2 -> 3 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 4

4 -> 1 -> 5 -> 2 -> 3 -> 4

4 -> 1 -> 5 -> 4

4 -> 1 -> 5 -> 6 -> 3 -> 4

4 -> 1 -> 5 -> 6 -> 4

4 -> 2 -> 3 -> 4

4 -> 2 -> 3 -> 5 -> 4

4 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 2 -> 5 -> 4

4 -> 2 -> 5 -> 6 -> 3 -> 4

4 -> 2 -> 5 -> 6 -> 4

4 -> 3 -> 5 -> 4

4 -> 3 -> 5 -> 6 -> 4

5 -> 1 -> 2 -> 3 -> 5

5 -> 1 -> 2 -> 5

5 -> 2 -> 3 -> 4 -> 1 -> 5

5 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 5

5 -> 4 -> 1 -> 5

5 -> 4 -> 2 -> 3 -> 5

5 -> 4 -> 2 -> 5

5 -> 4 -> 3 -> 5

5 -> 6 -> 3 -> 4 -> 1 -> 5

5 -> 6 -> 3 -> 4 -> 2 -> 5

5 -> 6 -> 3 -> 5

5 -> 6 -> 4 -> 1 -> 2 -> 5

5 -> 6 -> 4 -> 1 -> 5

5 -> 6 -> 4 -> 2 -> 3 -> 5

5 -> 6 -> 4 -> 2 -> 5

```
5  ->  6  ->  4  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  5  ->  6
6  ->  3  ->  4  ->  2  ->  5  ->  6
6  ->  3  ->  5  ->  6
6  ->  4  ->  1  ->  2  ->  5  ->  6
6  ->  4  ->  1  ->  5  ->  6
6  ->  4  ->  2  ->  3  ->  5  ->  6
6  ->  4  ->  2  ->  5  ->  6
6  ->  4  ->  3  ->  5  ->  6
graph is :
0   0   0   0   0   1   0
0   0   1   0   0   0   1
0   0   0   0   0   0   0
0   1   0   0   0   0   1
1   0   1   1   0   1   0
0   0   0   1   0   0   0
0   0   0   0   0   0   0
Number of complete of complete cycles: 12
complete path
1  ->  2  ->  3  ->  5  ->  6  ->  4  ->  1
1  ->  2  ->  5  ->  6  ->  3  ->  4  ->  1
2  ->  3  ->  5  ->  6  ->  4  ->  1  ->  2
2  ->  5  ->  6  ->  3  ->  4  ->  1  ->  2
3  ->  4  ->  1  ->  2  ->  5  ->  6  ->  3
3  ->  5  ->  6  ->  4  ->  1  ->  2  ->  3
4  ->  1  ->  2  ->  3  ->  5  ->  6  ->  4
4  ->  1  ->  2  ->  5  ->  6  ->  3  ->  4
5  ->  6  ->  3  ->  4  ->  1  ->  2  ->  5
5  ->  6  ->  4  ->  1  ->  2  ->  3  ->  5
6  ->  3  ->  4  ->  1  ->  2  ->  5  ->  6
6  ->  4  ->  1  ->  2  ->  3  ->  5  ->  6
```

Number of complete of partial cycles: 79

partial path

1 -> 2 -> 3 -> 4 -> 1

1 -> 2 -> 3 -> 5 -> 1

1 -> 2 -> 3 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 1

1 -> 2 -> 5 -> 4 -> 1

1 -> 2 -> 5 -> 6 -> 4 -> 1

1 -> 5 -> 2 -> 3 -> 4 -> 1

1 -> 5 -> 4 -> 1

1 -> 5 -> 6 -> 3 -> 4 -> 1

1 -> 5 -> 6 -> 4 -> 1

2 -> 3 -> 4 -> 1 -> 2

2 -> 3 -> 4 -> 1 -> 5 -> 2

2 -> 3 -> 4 -> 2

2 -> 3 -> 5 -> 1 -> 2

2 -> 3 -> 5 -> 2

2 -> 3 -> 5 -> 4 -> 1 -> 2

2 -> 3 -> 5 -> 4 -> 2

2 -> 3 -> 5 -> 6 -> 4 -> 2

2 -> 5 -> 1 -> 2

2 -> 5 -> 4 -> 1 -> 2

2 -> 5 -> 4 -> 2

2 -> 5 -> 6 -> 3 -> 4 -> 2

2 -> 5 -> 6 -> 4 -> 1 -> 2

2 -> 5 -> 6 -> 4 -> 2

3 -> 4 -> 1 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 2 -> 3

3 -> 4 -> 1 -> 5 -> 6 -> 3

3 -> 4 -> 2 -> 3

3 -> 4 -> 2 -> 5 -> 6 -> 3

3 -> 5 -> 1 -> 2 -> 3

3 -> 5 -> 2 -> 3

3 -> 5 -> 4 -> 1 -> 2 -> 3

3 -> 5 -> 4 -> 2 -> 3

3 -> 5 -> 4 -> 3

3 -> 5 -> 6 -> 3

3 -> 5 -> 6 -> 4 -> 2 -> 3

3 -> 5 -> 6 -> 4 -> 3

4 -> 1 -> 2 -> 3 -> 4

4 -> 1 -> 2 -> 3 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 4

4 -> 1 -> 2 -> 5 -> 6 -> 4

4 -> 1 -> 5 -> 2 -> 3 -> 4

4 -> 1 -> 5 -> 4

4 -> 1 -> 5 -> 6 -> 3 -> 4

4 -> 1 -> 5 -> 6 -> 4

4 -> 2 -> 3 -> 4

4 -> 2 -> 3 -> 5 -> 4

4 -> 2 -> 3 -> 5 -> 6 -> 4

4 -> 2 -> 5 -> 4

4 -> 2 -> 5 -> 6 -> 3 -> 4

4 -> 2 -> 5 -> 6 -> 4

4 -> 3 -> 5 -> 4

4 -> 3 -> 5 -> 6 -> 4

5 -> 1 -> 2 -> 3 -> 5

5 -> 1 -> 2 -> 5

5 -> 2 -> 3 -> 4 -> 1 -> 5

5 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 3 -> 5

5 -> 4 -> 1 -> 2 -> 5

5 -> 4 -> 1 -> 5

5 -> 4 -> 2 -> 3 -> 5

5 -> 4 -> 2 -> 5

5 -> 4 -> 3 -> 5

5 -> 6 -> 3 -> 4 -> 1 -> 5

5 -> 6 -> 3 -> 4 -> 2 -> 5

5 -> 6 -> 3 -> 5

5 -> 6 -> 4 -> 1 -> 2 -> 5

5 -> 6 -> 4 -> 1 -> 5

5 -> 6 -> 4 -> 2 -> 3 -> 5

5 -> 6 -> 4 -> 2 -> 5

5 -> 6 -> 4 -> 3 -> 5

6 -> 3 -> 4 -> 1 -> 5 -> 6

6 -> 3 -> 4 -> 2 -> 5 -> 6

6 -> 3 -> 5 -> 6

6 -> 4 -> 1 -> 2 -> 5 -> 6

6 -> 4 -> 1 -> 5 -> 6

6 -> 4 -> 2 -> 3 -> 5 -> 6

6 -> 4 -> 2 -> 5 -> 6

6 -> 4 -> 3 -> 5 -> 6