3 - Cycle

Problem Level: Medium

Problem Description:

Given a graph with N vertices (numbered from 0 to N-1) and M undirected edges, then count the distinct 3-cycles in the graph. A 3-cycle PQR is a cycle in which (P,Q), (Q,R) and (R,P) are connected by an edge.

Sample Input 1:
```
3 3
0 1
1 2
2 0
```

Sample Output 1:
```
1
```

Approach to be followed:

To find the number of three cycles in a graph, we need to run three loops. The first loop will iterate over every vertex**(a)** of the graph. The second loop will find all the adjacent vertices**(b)** of the vertex and the third loop will run over the adjacent vertices**(b)** and will find the adjacent vertices **(c)** of **b** vertex . Now, check if there is an edge between the **c** and **a.** If yes, then increment the count of 3-cycles by 1.

Steps:

1. Create an adjacency matrix as we have to traverse over all the adjacent vertices in the graph.
2. Create a variable **count** to store the number of 3-cycles.
3. Run nested loops over all the vertices of the graph then going through their adjacent vertices and again going over all the adjacent vertices of every adjacent vertex .
4. Check if there exists an edge between the start vertex and the vertex from the third loop.
5. If yes, increment the **count** by 1.

6. Return **Count**

Pseudo Code:

```
function getCycles(graph, n):

    Count = 0

    Loop from i = 0 till i = n-2

        Loop from j = i+1 till n-1

            Loop from k = j+1 till n

     // check if there is an edge between i,j and j,k and k,i

                if (graph[i][j] is true and graph[j][k] is true and
graph[k][i] is true)

                    Count = count + 1

    return Count
```

Time Complexity : O(N³) where N is the number of vertices in the input graph.