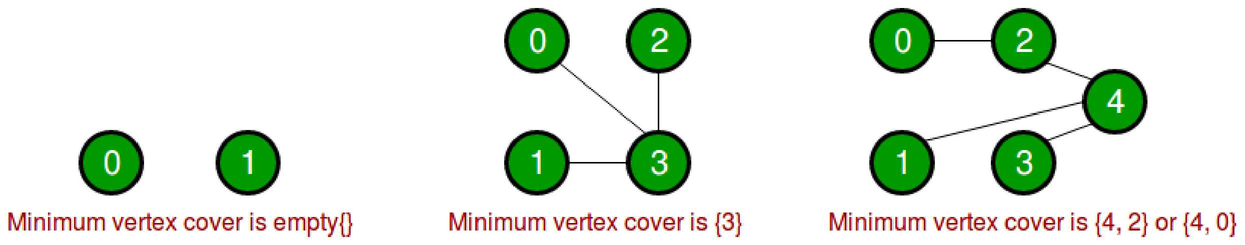
Experiment No.	**9**
Aim	**To implement Vertex Cover Problem using Approximation Algorithm.**
Name	**Prakhar Gupta**
UID No.	**2021300040**
Class & Division	**COMPS-A(C)**

THEORY :

A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either ‘u’ or ‘v’ is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph. ***Given an undirected graph, the vertex cover problem is to find minimum size vertex cover***.
The following are some examples.


0
1
Minimum vertex cover is empty{}
0
2
1
3
Minimum vertex cover is {3}
0
2
4
1
3
Minimum vertex cover is {4, 2} or {4, 0}

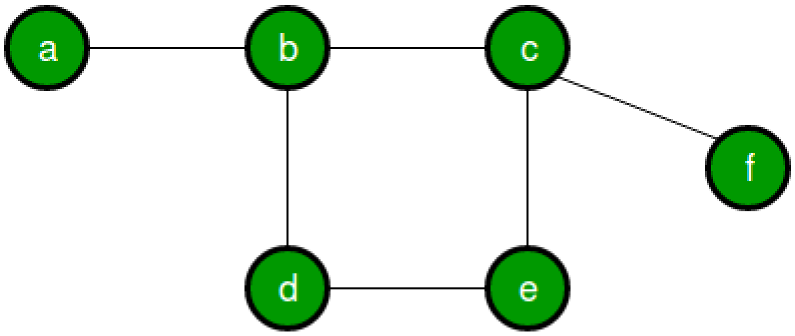
Vertex Cover Problem is a known NP Complete problem, i.e., there is no polynomial-time solution for this unless P = NP. There are approximate polynomial-time algorithms to solve the problem though. Following is a simple approximate algorithm adapted from CLRS book.
Naive Approach:
Consider all the subset of vertices one by one and find out whether it covers all edges of the graph. For eg. in a graph consisting only 3 vertices the set

consisting of the combination of vertices are:{0,1,2,{0,1},{0,2},{1,2},{0,1,2}} . Using each element of this set check whether these vertices cover all the edges of the graph. Hence update the optimal answer. And hence print the subset having minimum number of vertices which also covers all the edges of the graph.

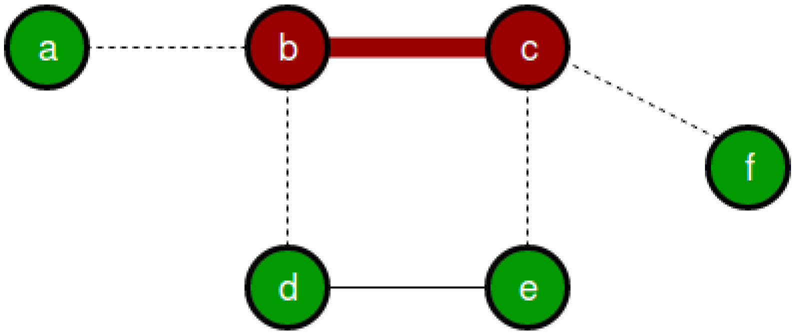
Approximate Algorithm for Vertex Cover:

```
1) Initialize the result as {}

2) Consider a set of all edges in given graph.  Let the set be E.

3) Do following while E is not empty

...a) Pick an arbitrary edge (u, v) from set E and add 'u' and 'v' to
result

...b) Remove all edges from E which are either incident on u or v.

4) Return result
```

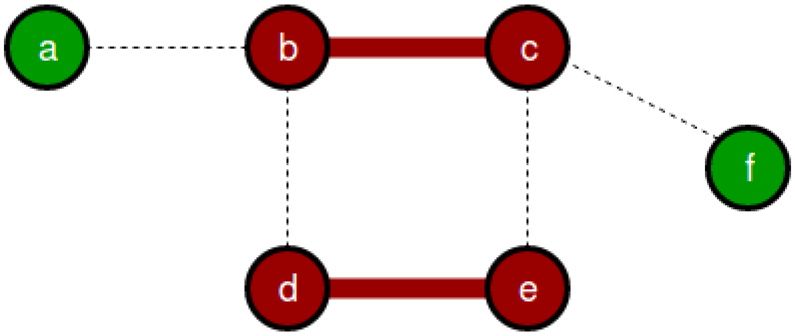
Below diagram to show the execution of the above approximate algorithm:



(a)



(b)



(c)

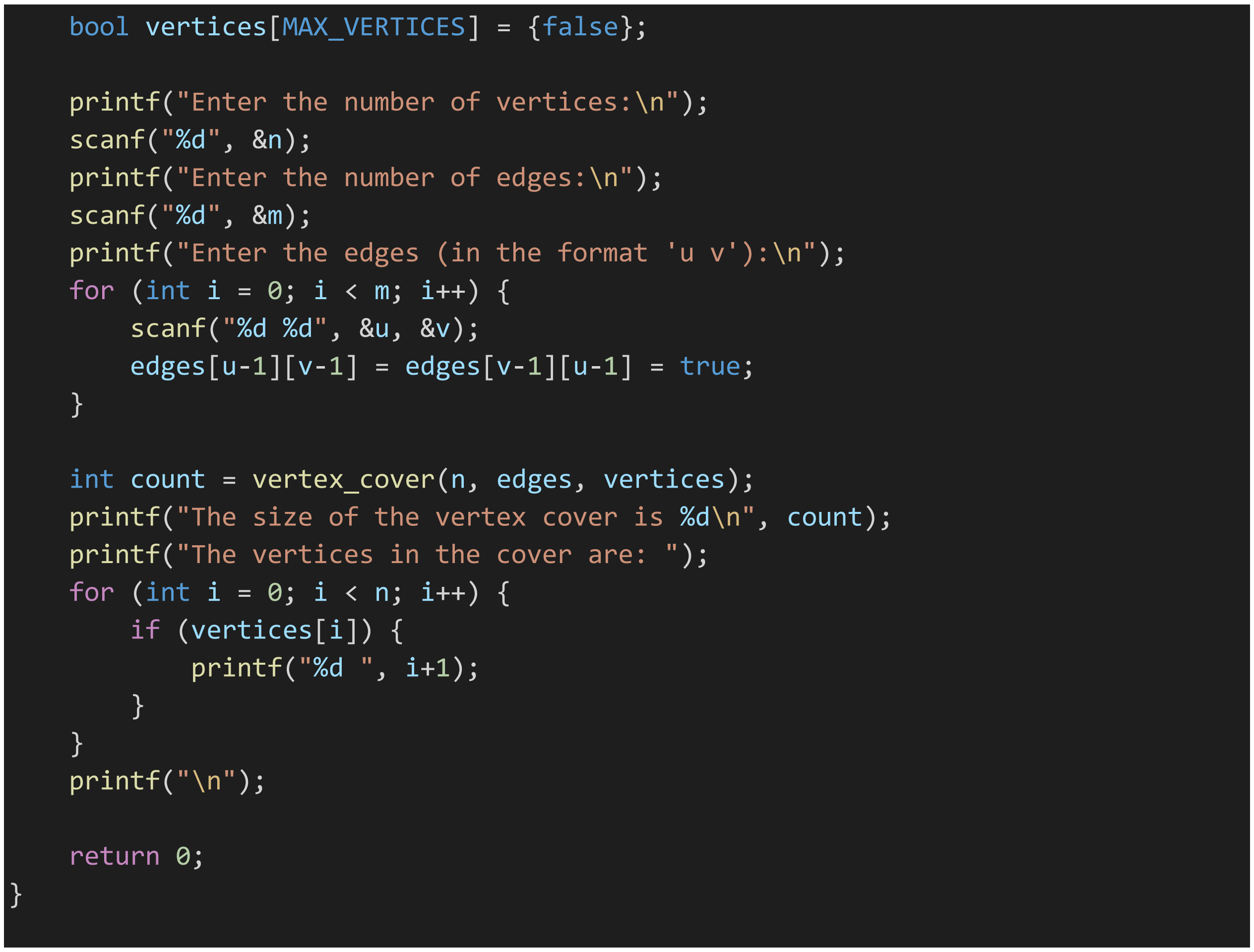
Minimum Vertex Cover is {b, c, d} or {b, c, e}

How well the above algorithm perform?
It can be proved that the above approximate algorithm never finds a vertex cover whose size is more than twice the size of the minimum possible vertex cover (Refer this for proof)
Implementation:

The following are C++ and Java implementations of the above approximate algorithm.

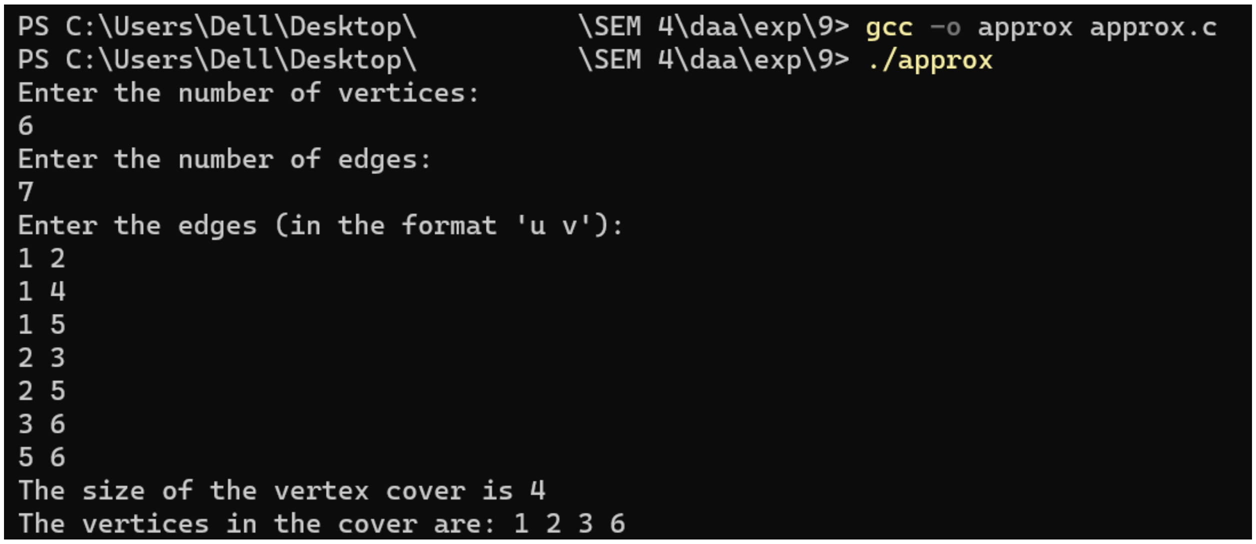
CODE:

```
#include <stdio.h>
#include <stdbool.h>

#define MAX_VERTICES 1000

bool edges[MAX_VERTICES][MAX_VERTICES] = {false};

int vertex_cover(int n, bool edges[MAX_VERTICES][MAX_VERTICES], bool
vertices[MAX_VERTICES]) {

    for (int i = 0; i < n; i++) {
        vertices[i] = false;
    }


    for (int i = 0; i < n; i++) {
        if (!vertices[i]) {
            for (int j = i; j < n; j++) {
                if (edges[i][j] && !vertices[i] && !vertices[j]) {
                    vertices[i] = vertices[j] = true;
                }
            }
        }
    }


    int count = 0;
    for (int i = 0; i < n; i++) {
        if (vertices[i]) {
            count++;
        }
    }

    return count;
}

int main()
{
    int n, m, u, v; //n=vertices m=edges
```



```c
    bool vertices[MAX_VERTICES] = {false};

    printf("Enter the number of vertices:\n");
    scanf("%d", &n);
    printf("Enter the number of edges:\n");
    scanf("%d", &m);
    printf("Enter the edges (in the format 'u v'):\n");
    for (int i = 0; i < m; i++) {
        scanf("%d %d", &u, &v);
        edges[u-1][v-1] = edges[v-1][u-1] = true;
    }

    int count = vertex_cover(n, edges, vertices);
    printf("The size of the vertex cover is %d\n", count);
    printf("The vertices in the cover are: ");
    for (int i = 0; i < n; i++) {
        if (vertices[i]) {
            printf("%d ", i+1);
        }
    }
    printf("\n");

    return 0;
}
```

Result:



```
PS C:\Users\Dell\Desktop\          \SEM 4\daa\exp\9> gcc -o approx approx.c
PS C:\Users\Dell\Desktop\          \SEM 4\daa\exp\9> ./approx
Enter the number of vertices:
6
Enter the number of edges:
7
Enter the edges (in the format 'u v'):
1 2
1 4
1 5
2 3
2 5
3 6
5 6
The size of the vertex cover is 4
The vertices in the cover are: 1 2 3 6
```

Conclusion : Understood the approximation algo by solving vertex cover problem