EXPERIMENT 4

Objective: Implementation of algorithms for vertex cover and set cover in graphs.
.
Brief Theory:

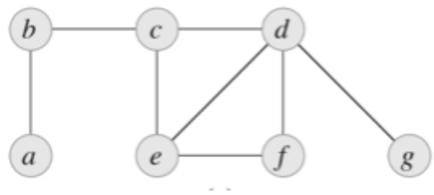
<u>Vertex Cover:</u> A vertex cover is a subset of vertices such that every edge in the graph is incident to at least one vertex in the subset. The minimum vertex cover is the smallest such subset.

```
APPROX-VERTEX-COVER(G)
C = ∅
E' = G.E
while E' ≠ ∅
    let (u, v) be an arbitrary edge of E'
    C = C ∪ {u, v}
    remove from E' edge (u, v) and every edge incident on either u or v
return C
```

<u>Set Cover:</u> The set cover algorithm provides solutions to many real-world resource allocating problems. For instance, consider an airline assigning crew members to each of their airplanes such that they have enough people to fulfill the requirements for the journey. They consider the flight timings, the duration, the pit-stops, availability of the crew to assign them to the flights. This is where the set cover algorithm comes into picture. Given a universal set U, containing few elements which are all divided into subsets. Considering the collection of these subsets as S = $\{S_1, S_2, S_3, S_4 ... S_n\}$, the set cover algorithm finds the minimum number of subsets such that they cover all the elements present in the universal set.

```
GREEDY-SET-COVER(X, F)
U = X
C = ∅
while U ≠ ∅
    select an S ∈ F that maximizes |S ∩ U|
    U = U − S
    C = C ∪ {S}
return C
```

Task:



1) Implement the approximation algorithm to find a vertex cover in the given graph and calculate its size. (Use Adjacency matrix)
2) Create a program where users can input a graph (vertices and edges) and visualize the vertex cover computed by the greedy algorithm. (Use Linked Representation)
3) Create program to output the minimal collection of subsets to cover all elements in U.

A universal set U with n elements (e.g., $U = \{1, 2, 3, 4, 5\}$).

A collection of subsets $S_1, S_2, ..., S_m$, where each $S_i \subseteq U$ (e.g., $S = \{\{1, 2\}, \{2, 3, 4\}, \{4, 5\}\}$).

Apparatus and components required: Computer with C or C++ Compiler and Linux/Windows platform.

Experimental/numerical procedure: Coding, compilation, editing, run and debugging.