NAME:	Prithvi Singh
UID:	2022301014
SUBJECT	DAA
EXPERIMENT NO:	09
AIM:	To implement vertex cover Problem

Algorithm:	Approx-Vertex-Cover (G = (V, E))
	{
	C = empty-set;
	E'= E;
	While E' is not empty do
	{
	Let (u, v) be any edge in E': (*)
	Add u and v to C;
	Remove from E' all edges incident to
	u or v;
	}
	Return C;
	}

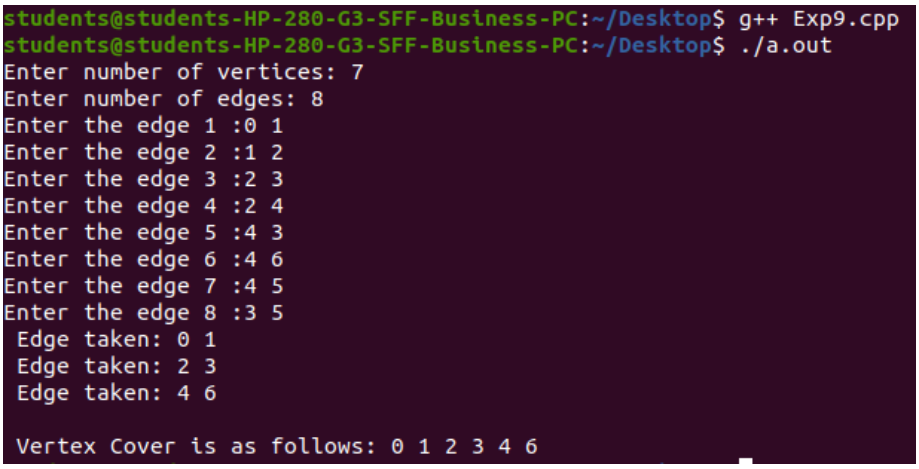
Code:	#include<iostream>
	#include <list>
	using namespace std;
	
	class Graph
	{
	int V;
	list<int> *adj;
	public:
	Graph(int V);
	void addEdge(int v, int w);

```
    void printVertexCover();
};

Graph::Graph(int V)
{
    this->V = V;
    adj = new list<int>[V];
}

void Graph::addEdge(int v, int w)
{
    adj[v].push_back(w);
    adj[w].push_back(v);
}


void Graph::printVertexCover()
{

    bool visited[V];
    for (int i=0; i<V; i++)
        visited[i] = false;

    list<int>::iterator i;

    for (int u=0; u<V; u++)
    {

        if (visited[u] == false)
        {

            for (i= adj[u].begin(); i != adj[u].end(); ++i)
            {
                int v = *i;
                if (visited[v] == false)
                {
```

```
                visited[v] = true;
                visited[u]  = true;
                cout<<" Edge taken: "<< u<< " "<< v<<endl;
                break;
            }
        }
    }
}

cout<<"\n Vertex Cover is as follows: ";
    for (int i=0; i<V; i++)
        if (visited[i])
          cout << i << " ";
}


int main()
{
    int v;
    cout<<"Enter number of vertices: ";
    cin>>v;


    int e;
    cout<<"Enter number of edges: ";
    cin>>e;

    Graph g(v);

    for(int i=0;i<e;i++){
    int a,b;
      cout<<"Enter the edge "<<i+1<<" :";
      cin>>a>>b;
      g.addEdge(a,b);
    }
```

```
    g.printVertexCover();
    cout<<endl;
    return 0;
}
```

Output:



Conclusion: Thus we have implemented Vertex cover Problem.