

<b>NAME:</b>	Prithvi Singh
<b>UID:</b>	2022301014
<b>SUBJECT</b>	DAA
<b>EXPERIMENT NO:</b>	09
<b>AIM:</b>	To implement vertex cover Problem
<b>Algorithm:</b>	<p>Approx-Vertex-Cover (<math>G = (V, E)</math>)</p> <pre> {     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; } </pre>
<b>Code:</b>	<pre> #include&lt;iostream&gt; #include &lt;list&gt; using namespace std;  class Graph {     int V;     list&lt;int&gt; *adj; public:     Graph(int V);     void addEdge(int v, int w); </pre>

```

    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 << " ";
}

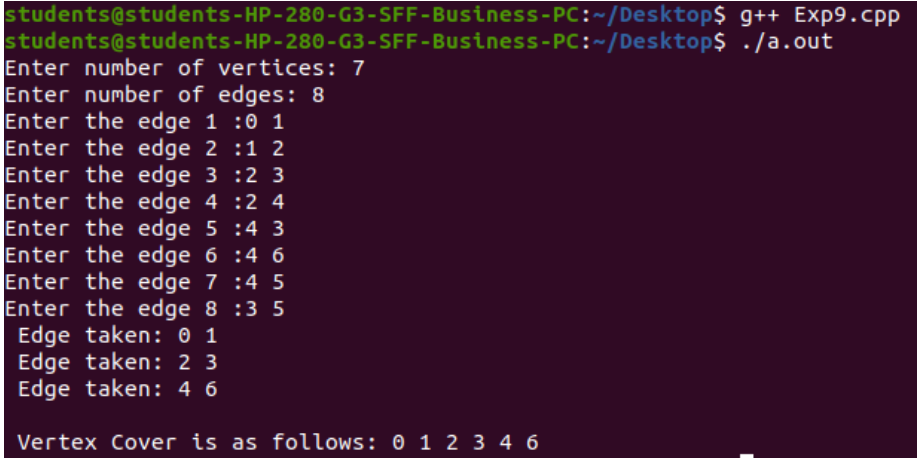
int main()
{
    int v;
    cout<<"Enter number of vertices: ";
    cin>>v;

    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);
    }
}

```

	<pre> g.printVertexCover(); cout&lt;&lt;endl; return 0; } </pre>
<b>Output:</b>	 <pre> students@students-HP-280-G3-SFF-Business-PC:~/Desktop\$ g++ Exp9.cpp students@students-HP-280-G3-SFF-Business-PC:~/Desktop\$ ./a.out Enter number of vertices: 7 Enter number of edges: 8 Enter the edge 1 :0 1 Enter the edge 2 :1 2 Enter the edge 3 :2 3 Enter the edge 4 :2 4 Enter the edge 5 :4 3 Enter the edge 6 :4 6 Enter the edge 7 :4 5 Enter the edge 8 :3 5 Edge taken: 0 1 Edge taken: 2 3 Edge taken: 4 6  Vertex Cover is as follows: 0 1 2 3 4 6 </pre>
<b>Conclusion:</b>	Thus we have implemented Vertex cover Problem.