PSUEDO_CODE-

```
--> color[] holds values for a node as follows:
        > -1, the node hasn't been visited in the dfs
        >  0, the node should be a part of the first set in bipartite(V1)
        >  1, the node should be a part of the second set in bipartite(V2)

    Initially all color values are -1

--> Nodes of the graph are numbered, the source/first node is represented by 0.


dfs(u, color)                               // updates color & returns True if graph is bipartite
{
    for all (u,v) in Edges:                 // scanning through edges of Node u of graph

        if(color[v]==-1)
            color[v] = color[u] xor 1       // Assigning the opposite set of u to unvisited Nodes
            if(dfs(v)==False)
                return False                // Graph found non-bipartite during recursive calls

        else if(color[v]==color[u])
            return False                    // Graph not bipartite - Nodes of edge(u,v) have same color

    return True                             // Graph bipartite - colors properly assigned to the 2 sets
}

bipartite(G)
{
    color[0] = 0                    // First Node should of be first set

    if(dfs(0) == True)              // Prints YES and the 2 sets if graph found bipartite after dfs search
    {
        print("YES")
        for k in V:
            print(color[k]+1)
    }
    else                            // Prints NO if graph found non-bipartite after dfs search
        print("NO")
}
```