```
 1 import java .util.*;
 2 class Codechef{
 3   static class ClassRoom
   {
 4    int src;
 5    int dist;
 6    int wgt;
 7
 8   public  ClassRoom(int
   s,int d,int w){
 9     this.src=s;
10     this.dist=d;
11     this.wgt=w;
12
13   }
14 }
15   public static void
   main(String args[]){
16
17     ArrayList <ClassRoom
```

```
17 >[] graph=new ArrayList[
   5];
18     for(int i=0;i<graph.
   length;i++){
19         graph[i]=new
   ArrayList<>();
20
21     } graph[0].add(new
   ClassRoom(0,1,5));
22         graph[0].add(new
   ClassRoom(0,2,5));
23     graph[1].add(new
   ClassRoom(1,0,1));
24     graph[1].add(new
   ClassRoom(1,3,3));
25     graph[2].add(new
   ClassRoom(2,0,5));
26     //graph[2].add(new
   ClassRoom(2,4,1));
27    // graph[2].add(new
```

```
27 ClassRoom(2,4,3));
28      graph[3].add(new
   ClassRoom(3,1,5));
29     // graph[3].add(new
   ClassRoom(3,4,1));
30     //graph[4].add(new
   ClassRoom(4,2,2));
31      //  graph[4].add(new
    ClassRoom(4,3,2));
32          System.out.println
   (bipartite(graph));
33    }
34    public static boolean
   bipartite (ArrayList <
   ClassRoom>[] graph){
35          int cpl[]=new int
    [graph.length];
36          for(int i=0;i<
   graph.length;i++){
37              cpl[i]=-1;
```

```
38          }
39          Queue <Integer>q=
    new LinkedList<>();
40          // int curr=
41          // q.add(curr);
42          for(int i=0;i<
    graph.length;i++){
43              if(cpl[i]==-1
    ){
44                  q.add(i);
45                  while (!q.
    isEmpty()){
46                      int
    curr=q.remove();
47                      for(
    int j=0;j<graph[curr].
    size();j++){
48
    ClassRoom cl=graph[curr
    ].get(j);
```

```
49                                  if
   (cpl[cl.dist]==-1){
50
     int anothercl=cpl[curr
   ]==0?1:0;
51
     cpl[cl.dist]=anothercl
   ;
52
     q.add(cl.dist);
53
                                    }
54
55
   else if(cpl[cl.dist]==
   cpl[curr]){
56
     return false;
57                                  }
58                                }
59                              }
```

```
60              }
61
62              // ClassRoom cl
   =graph[curr].get(i);
63              // q.add(cl.
   dist);
64          }
65      return true;
66
67    }
68 public static void bfs(
   ArrayList <ClassRoom>[]
   graph){
69      Boolean bol[]=new
   Boolean[graph.length];
70      for(int i=0;i<graph.
   length;i++){
71          if(!bol[i]){
72              dfs(graph,i,bol
   );
```

```
73          }
74      }
75
76
77  }
78  public static void dfs(
    ArrayList<ClassRoom>[]
    graph,int j,Boolean bol
    []){
79      Queue<Integer> q=new
     LinkedList<>();
80      q.add(j);
81      while(!q.isEmpty()){
82          int curr =q.remove
    ();
83          bol[curr]=true;
84          if(!bol[curr]){
85              System.out.
    println(curr);
86              for(int i=0;i<
```

```
 86 graph[curr].size();i
    ++){
 87             ClassRoom cl=
    graph[curr].get(i);
 88             q.add(cl.dist
    );
 89
 90
 91           }
 92         }
 93
 94
 95
 96
 97       }
 98
 99
100 }
101
102
```

```
103 }
104
```