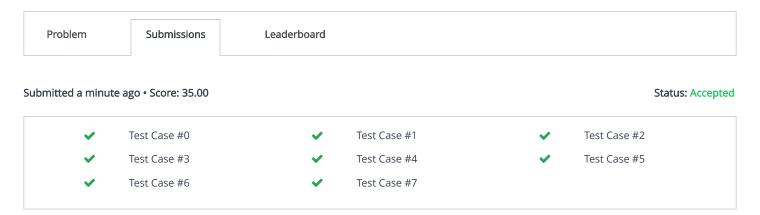
All Contests > HW6_MST > HW6) Investors

HW6) Investors



Submitted Code

```
Language: Python 3
                                                                                              Open in editor
1 def getNum(letter):
       return ord(letter) - ord("A")
2
3
4 def getLetter(num):
5
       return chr(num + ord("A"))
6
7 def intersection(L1, L2):
       L3 = []
8
       for e in L1:
9
10
           if e in L2:
11
               L3.append(e)
12
       return L3
13
14 def union(L1, L2):
       L3 = L1.copy()
15
16
       for e in L2:
           if e not in L1:
17
18
               L3.append(e)
19
       return L3
20
21 def AdjacencyMatrix(V, edges):
       graph = [[False]*V for _ in range(V) ]
22
23
       for edge in edges:
24
25
           X,Y = (edge)
           i, j = getNum(X),getNum(Y)
26
27
28
           graph[i][j] = True
29
           graph[j][i] = True
30
       return graph
31
32 def contacts(graph, investor):
```

```
34
       size = len(graph)
35
       family = [investor]
36
       stack = [investor]
37
       visited = [False]*size
38
       visited[investor] = True
39
       while stack:
40
41
           node = stack.pop()
42
43
           visited = [False]*size
           for i in family:
44
45
               visited[i] = True
46
47
48
           for i in range(size):
49
               if not visited[i]:
50
                    visited[i] = True
51
                    if graph[node][i]==True:
52
                        family.append(i)
                        stack.append(i)
53
54
       return family
55
56
57 def check_connection(graph, investor1, investor2):
58
59
       size = len(graph)
60
       contacts1 = contacts(graph, investor1)
61
62
       contacts2 = contacts(graph, investor2)
63
64
       inter = intersection(contacts1, contacts2)
65
66
       totalcontacts = union(contacts1, contacts2)
67
       totalcontacts.sort()
68
       totalcontactsletters = [getLetter(c) for c in totalcontacts]
       word = ""
69
       for c in totalcontactsletters:
70
71
           word = word + c
72
73
       if not inter:
74
75
           graph[investor1][investor2] = True
76
           graph[investor2][investor1] = True
77
78
           print("T", word)
79
80
       else:
           print("F", word)
81
82
83
84
85 #input treatment
86 ##graph definition
87
88 graph_numbers = input()
89 graph_numbers = tuple(map(int,graph_numbers.split()))
90 V, E = graph_numbers
91
92 edges = []
93 for i in range(E):
       edge = input()
94
95
       edge = tuple(map(str,edge.split()))
96
       edges.append(edge)
97
98 graph = AdjacencyMatrix(V, edges)
99
```

```
100 ##queries
101 num_queries = int(input())
102 for i in range(num_queries):
103
       q = input()
104
        q = tuple(map(str,q.split()))
105
        investor1, investor2 = q
106
        check_connection(graph, getNum(investor1), getNum(investor2))
107
108
109
110
111
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |