



CSES Problem Set

Planets and Kingdoms

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Submission details

_Task:	<u>Planets and Kingdoms</u>
Sender:	seleneal1996
Submission time:	2021-11-26 20:25:54
Language:	C++17
Status:	READY
Result:	ACCEPTED

Test results

test	verdict	time	
#1	ACCEPTED	0.01 s	<u>>></u>
#2	ACCEPTED	0.01 s	<u>>></u>
#3	ACCEPTED	0.01 s	<u>>></u>
#4	ACCEPTED	0.01 s	<u>>></u>
#5	ACCEPTED	0.01 s	<u>>></u>
#6	ACCEPTED	0.15 s	<u>>></u>
#7	ACCEPTED	0.15 s	<u>>></u>
#8	ACCEPTED	0.15 s	<u>>></u>
#9	ACCEPTED	0.15 s	<u>>></u>
#10	ACCEPTED	0.15 s	<u>>></u>

Code -

```
//https://cses.fi/problemset/task/1683/
   #include <bits/stdc++.h>
 3
   class P7{
 4
     public:
 5
     int visit[100005];
 6
     std::stack<int>A;
 7
     std::vector<int> part[100005], sequence[100]
 8
     std::vector<std::vector<int>> adj, adj_rev;
 9
     int k=0;
10
     void first_dfs(int u)
11
12
       if (visit[u])
13
         return;
14
       visit[u]=1;
15
       for(auto var:part[u])
         first_dfs(var);
16
17
       A.push(u);
18
19
     }
```

Graph Algorithms

Your submissions

2021-11-26 20:25:54	✓
2021-11-26 19:58:35	×
2021-11-26 19:55:44	×
2021-11-26 18:38:09	~

```
20
     void second_dfs(int u)
21
22
        if(visit[u])
23
          return;
24
        visit[u]=k;
25
        for(auto var:sequence[u])
26
          second dfs(var);
27
     }
28
29
     void final_solution(){
30
        int n,m;
31
        std::cin>>n>>m;
32
        for (int i=0; i<m; i++){
33
          int a,b;
34
          std::cin>>a>>b;
35
          part[a].push back(b);
36
          sequence[b].push_back(a);
37
38
        for (int i=1; i<n+1; i++){</pre>
39
          if(!visit[i])
40
            first_dfs(i);
41
        }
42
        memset(visit,0,sizeof visit);
43
        while(!A.empty())
44
45
            int x = A.top();
46
            A.pop();
47
            if (!visit[x])
48
49
              k++;
50
              second_dfs(x);
51
            }
52
53
        std::cout << k <<"\n";
54
        for (int i=1; i<n+1; i++)
55
          std::cout <<visit[i]<<"\n";</pre>
56
     }
57
58
   };
59
60 | int main(){
61
        std::ios::sync_with_stdio(0);
62
        std::cin.tie(0);
63
        P7 S1= P7();
64
        int t=1;
65
        while(t--){
66
          S1.final_solution();
67
68
        return 0;
69 }
```

Share code to others

Test details ▲

Test 1

```
input
10 20
4 5
```

```
10 7
6 1
6 5
...
```

```
user output

4
4
3
1
1
....
```

Verdict: ACCEPTED

```
input

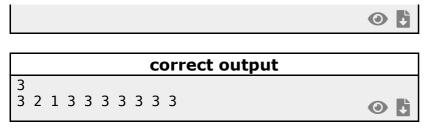
10 20
5 6
1 2
7 9
7 3
...
```

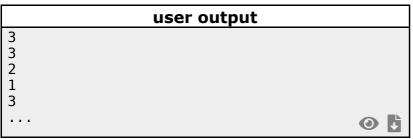
```
2
1
1
2
....
```

Test 3

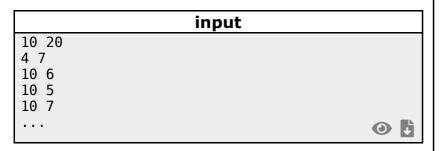
```
input

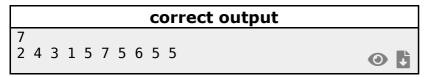
10 20
1 6
10 1
1 10
9 10
...
```

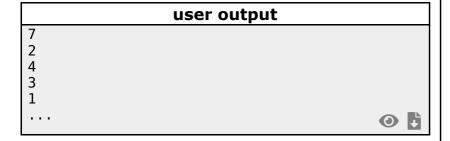




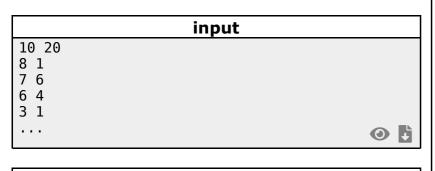
Verdict: ACCEPTED

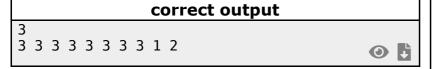


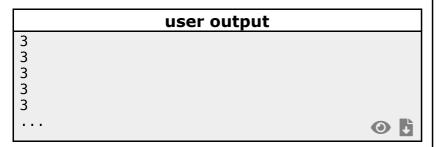




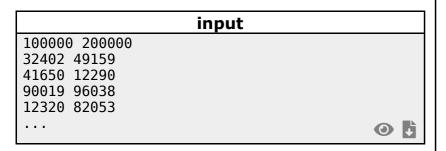
Test 5







Verdict: ACCEPTED



```
43755
26333 26333 40691 34351 26331 ...
```

```
43755
26333
26333
40691
34351
```

Test 7

```
input

100000 200000

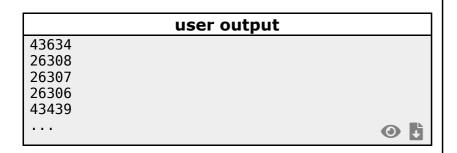
98891 58773

74281 97370

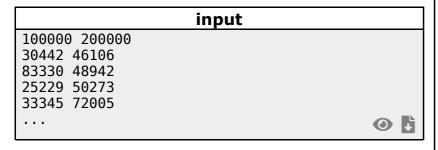
25400 8211

25600 1357
...
```

		C	orrect	outpu	ıt	
43634			_	_		
26308	26307	26306	43439	26305		O



Verdict: ACCEPTED



		C	orrect	outpu	ıt		
43498							
43496	26032	26031	32415	26033		0	+

	user output	
43498		
43496		
26032		
26031		
32415		
		O

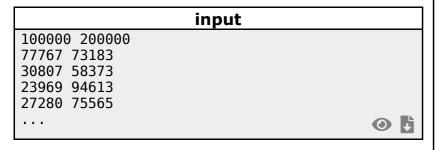
Test 9

i	nput
100000 200000	
45422 77478	
6800 88602	
9724 59882	
20954 36466	
	O

	correct	t output	
44087 26407 26407	26407 26404	26407	0

	user output	
44087		





		C	orrect	outpu	ıt		
43903	_	_					
43903	43902	26255	26256	29892		0	

	user output	
43903		
43903		
43902		
26255		
26256		
		O