



### **CSES Problem Set**

# **Monsters**

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

# **Submission details**

Task:	<u>Monsters</u>	
Sender:	seleneal1996	
Submission time:	2021-11-26 07:27:52	
Language:	C++17	
Status:	READY	
Result:	ACCEPTED	

# **Test results** ▲

test	verdict	time	
#1	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#2	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#3	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#4	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#5	ACCEPTED	0.04 s	<u>&gt;&gt;</u>
#6	ACCEPTED	0.04 s	<u>&gt;&gt;</u>
#7	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#8	ACCEPTED	0.04 s	<u>&gt;&gt;</u>
#9	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#10	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#11	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#12	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#13	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#14	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#15	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#16	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#17	ACCEPTED	0.04 s	<u>&gt;&gt;</u>
#18	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#19	ACCEPTED	0.03 s	<u>&gt;&gt;</u>
#20	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#21	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#22	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#23	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#24	ACCEPTED	0.01 s	<u>&gt;&gt;</u>
#25	ACCEPTED	0.01 s	<u>&gt;&gt;</u>

# **Graph Algorithms**

•••	
Message Route	_
Building Teams	_
Round Trip	_
Monsters	<b>✓</b>
Shortest Routes I	_
Shortest Routes II	_
High Score	_
Flight Discount	_
•••	

#### **Your submissions**

2021-11-26 07:27:52	<b>✓</b>
2021-11-26 07:25:09	X

## Compiler report -

```
input/code.cpp: In member function 'void Monster:
input/code.cpp:13:5: warning: this 'for' clause c
    for (auto &i: grid)
    ^~~

input/code.cpp:15:7: note: ...this statement, but
    std::queue<std::pair<int,int>> q;
    ^~~

input/code.cpp:24:15: warning: 'y' may be used ur
    A[x][y] = -1;
    ~~~~~~~~~~~~

input/code.cpp:24:15: warning: 'x' may be used ur
```

#### Code -

```
1 //https://cses.fi/problemset/task/1194
   #include <bits/stdc++.h>
   int fx[] = {-1, 1, 0, 0};
 4 \mid int fy[] = \{0, 0, 1, -1\};
   char characters[] = {'U', 'D', 'R', 'L'};
   int A[1005][1005];
   class Monster{
 8
   public:
 9
     void Solve(){
        int n,m;
10
11
        std::cin>>n>>m;
12
        std::vector<std::string> grid(n);
13
        for (auto &i: grid)
14
          std::cin>>i;
          std::queue<std::pair<int,int>> q;
15
16
          int x,y;
17
          for (int i = 0; i < n; i++)
18
            for (int j = 0; j < m; j++)
19
              if (grid[i][j] == 'M')
20
                q.push({i,j});
21
              else if (grid[i][j] == 'A')
22
                x = i, y = j;
23
          q.push({x,y});
24
          A[x][y] = -1;
25
          while(!q.empty())
26
27
            auto [x,y] = q.front();
28
            q.pop();
29
            if (grid[x][y] == 'A' && (x == 0 || x)
30
31
              std::cout<<"YES"<<"\n";
32
              std::string rpta;
33
              int d = A[x][y];
34
              while(d != -1)
35
36
                rpta += characters[d];
37
                x -= fx[d];
                y -= fy[d];
38
39
                d = A[x][y];
40
41
              reverse(rpta.begin(), rpta.end());
42
              std::cout<<rpta.size()<<"\n";</pre>
43
              std::cout<<rpta;</pre>
44
              return;
45
46
47
            for (int i = 0; i < 4; i++)
```

```
48
49
              int xx = x + fx[i], yy = y + fy[i];
50
              if (xx<0 || xx>=n || yy<0 || yy>=m
51
                 continue;
52
              else
53
54
                 grid[xx][yy] = grid[x][y];
55
                 if (grid[xx][yy] == 'A')
56
                   A[xx][yy] = i;
57
                 q.push({xx,yy});
58
59
            }
60
61
          std::cout<<"NO";</pre>
62
     }
63
64 int main()
65
66
     std::ios base::sync with stdio(false);
67
     std::cin.tie(0);
     Monster S1= Monster();
68
     int t=1;
69
     for (int i = 1; i <= t; i++)
70
71
72
        S1.Solve();
73
        std::cout<<'\n';</pre>
74
75
     return 0;
76 | }
```

#### Share code to others

#### Test details ▲

#### Test 1

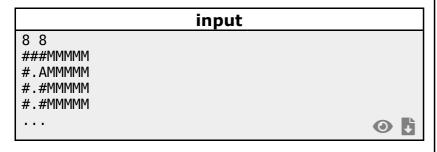
```
input

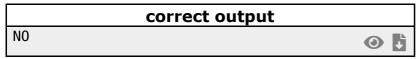
8 8
###MMMMM
#.AMMMMM
#.#MMMMM
#.#MMMMM
#.#MMMMM
```

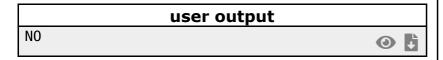
```
YES 7 LDDDDDDD
```

```
YES 7 LDDDDDDD
```

Verdict: ACCEPTED

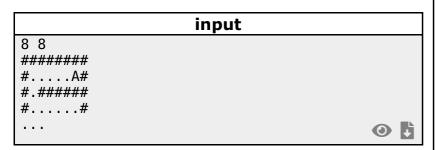




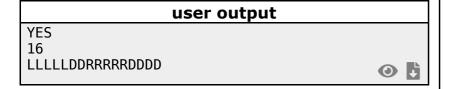


#### Test 3

Verdict: ACCEPTED



YES
16
LLLLLDDRRRRRDDDD



#### Test 4

```
input

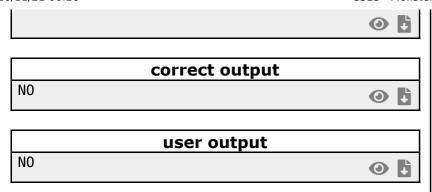
8 8

#######

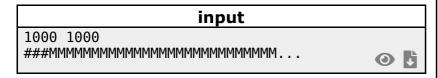
#....A#

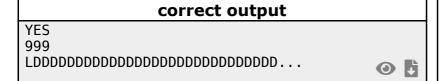
#.#####

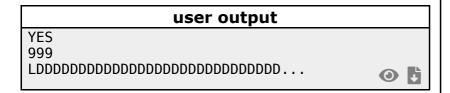
#....#
```



Test 5

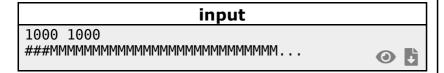


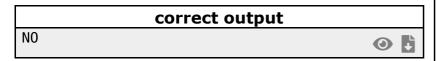


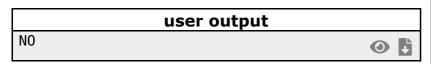


#### Test 6

Verdict: ACCEPTED







#### Test 7

CORRECT OUTPUT

YES
96
DDDDDDDDDDDRRUUUUUUUUUUUURRDDD...

# YES 96 DDDDDDDDDDDRRUUUUUUUUUUURRDDD...

#### Test 8

Verdict: ACCEPTED



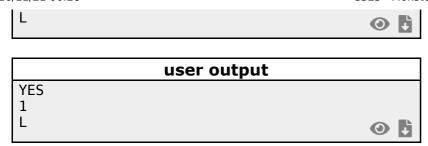
#### Test 9

Verdict: ACCEPTED

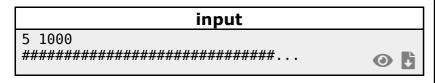
```
input

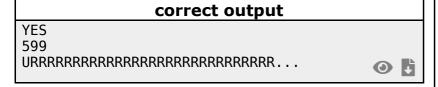
10 3
M#.
M#.
M#.
M#.
...
```

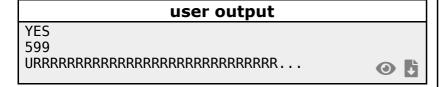
# YES 1



Verdict: ACCEPTED

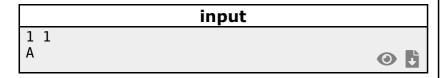






#### Test 11

Verdict: ACCEPTED



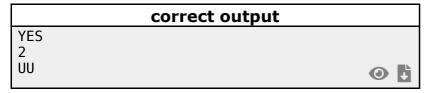
	correct output
YES	
0	<b>◎</b> ₺

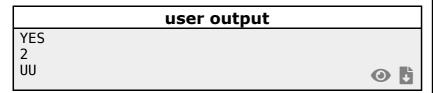
	user output	
YES		
0		<b>O</b>

#### **Test 12**

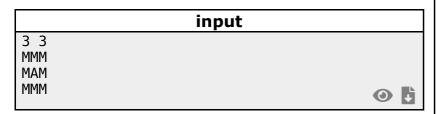


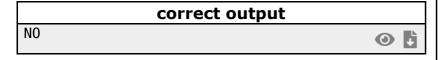


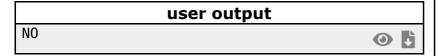




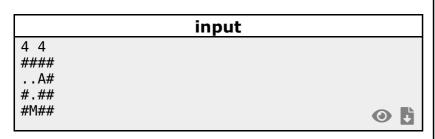
**Test 13** 







#### **Test 14**

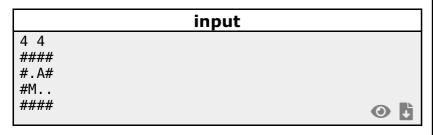


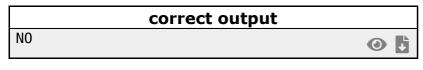
	correct output	
YES		
ĹL		<b>O</b>

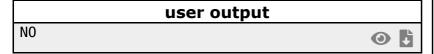
user output		
YES		



Verdict: ACCEPTED

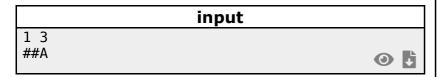


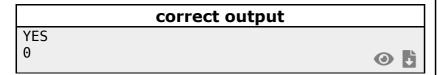


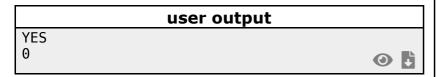


#### Test 16

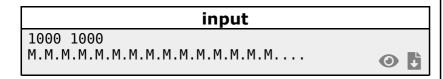
Verdict: ACCEPTED



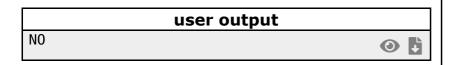




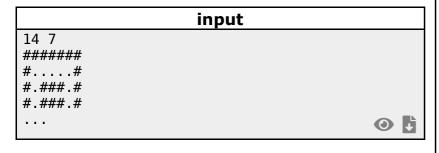
#### **Test 17**

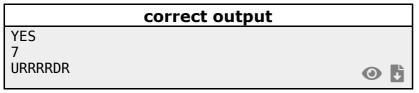


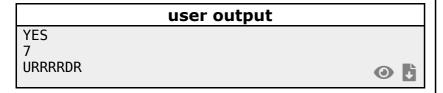
	correct output	
NO		<b>O</b>



Verdict: ACCEPTED

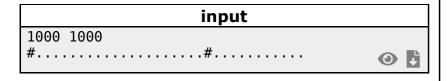


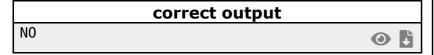




#### **Test 19**

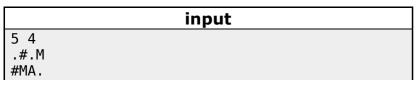
Verdict: ACCEPTED

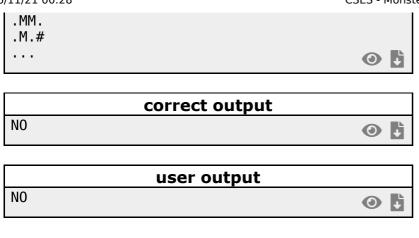




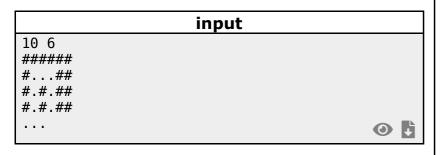
	user output	
NO	<b>@</b>	+

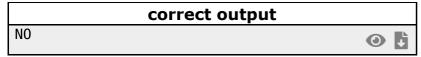
#### Test 20

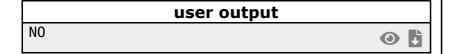




**Test 21** 

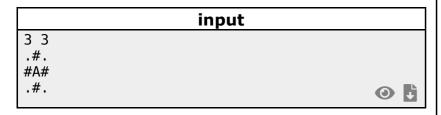


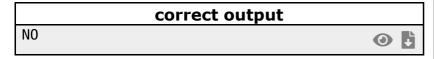


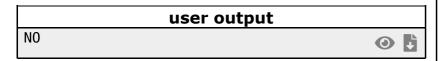


#### Test 22

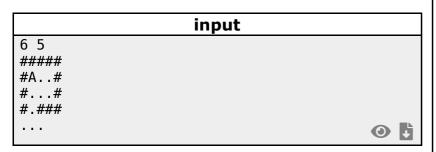
Verdict: ACCEPTED

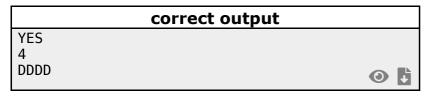


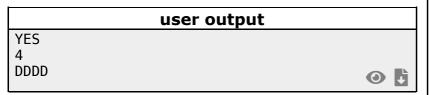




#### **Test 23**

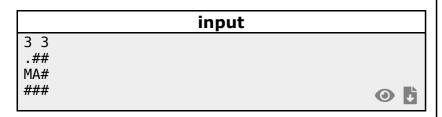


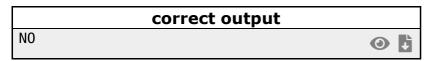




#### Test 24

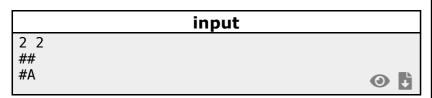
Verdict: ACCEPTED







#### Test 25



	correct output	
YES		
0		<b>O</b>

