CSES seleneal1996 — ◐ ⇨

CSES Problem Set

Monsters

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Submission details

Task:	Monsters
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	»
#2	ACCEPTED	0.01 s	»
#3	ACCEPTED	0.01 s	»
#4	ACCEPTED	0.01 s	»
#5	ACCEPTED	0.04 s	»
#6	ACCEPTED	0.04 s	»
#7	ACCEPTED	0.01 s	»
#8	ACCEPTED	0.04 s	»
#9	ACCEPTED	0.01 s	»
#10	ACCEPTED	0.01 s	»
#11	ACCEPTED	0.01 s	»
#12	ACCEPTED	0.01 s	»
#13	ACCEPTED	0.01 s	»
#14	ACCEPTED	0.01 s	»
#15	ACCEPTED	0.01 s	»
#16	ACCEPTED	0.01 s	»
#17	ACCEPTED	0.04 s	»
#18	ACCEPTED	0.01 s	»
#19	ACCEPTED	0.03 s	»
#20	ACCEPTED	0.01 s	»
#21	ACCEPTED	0.01 s	»
#22	ACCEPTED	0.01 s	»
#23	ACCEPTED	0.01 s	»
#24	ACCEPTED	0.01 s	»
#25	ACCEPTED	0.01 s	»

Graph Algorithms

...

Message Route	–
Building Teams	–
Round Trip	–
Monsters	✓
Shortest Routes I	–
Shortest Routes II	–
High Score	–
Flight Discount	–

...

Your submissions

2021-11-26 07:27:52	✓
2021-11-26 07:25:09	✗

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
 2 #include <bits/stdc++.h>
 3 int fx[] = {-1, 1, 0, 0};
 4 int fy[] = {0, 0, 1, -1};
 5 char characters[] = {'U', 'D', 'R', 'L'};
 6 int A[1005][1005];
 7 class Monster{
 8 public:
 9   void Solve(){
10     int n,m;
11     std::cin>>n>>m;
12     std::vector<std::string> grid(n);
13     for (auto &i: grid)
14       std::cin>>i;
15       std::queue<std::pair<int,int>> q;
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];
38             y -= fy[d];
39             d = A[x][y];
40           }
41           reverse(rpta.begin(), rpta.end());
42           std::cout<<rpta.size()<<"\n";
43           std::cout<<rpta;
44           return;
45         }
46 
47         for (int i = 0; i < 4; i++)
```

```
    {
     int xx = x + fx[i],yy = y + fy[i];
     if (xx<0 || xx>=n || yy<0 || yy>=m
      continue;
     else
     {
      grid[xx][yy] = grid[x][y];
      if (grid[xx][yy] == 'A')
       A[xx][yy] = i;
      q.push({xx,yy});
     }
    }
   }
   std::cout<<"NO";
 }
};
int main()
{
 std::ios_base::sync_with_stdio(false);
 std::cin.tie(0);
 Monster S1= Monster();
 int t=1;
 for (int i = 1; i <= t; i++)
 {
  S1.Solve();
  std::cout<<'\n';
 }
 return 0;
}
```

Share code to others

Test details ▲

Test 1

Verdict: ACCEPTED

input

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

correct output

```
YES
7
LDDDDDD
```

user output

```
YES
7
LDDDDDD
```

Test 2

Verdict: ACCEPTED

input

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

correct output

```
NO
```

user output

```
NO
```

Test 3

Verdict: ACCEPTED

input

```
8 8
########
#.....A#
#.######
#......#
...
```

correct output

```
YES
16
LLLLLDDRRRRRDDDD
```

user output

```
YES
16
LLLLLDDRRRRRDDDD
```

Test 4

Verdict: ACCEPTED

input

```
8 8
########
#.....A#
#.######
#......#
...
```

correct output

```
NO
```

user output

```
NO
```

Test 5

Verdict: ACCEPTED

input

```
1000 1000
###MMMMMMMMMMMMMMMMMMMMMMMMMMM...
```

correct output

```
YES
999
LDDDDDDDDDDDDDDDDDDDDDDDDDDDDD...
```

user output

```
YES
999
LDDDDDDDDDDDDDDDDDDDDDDDDDDDDD...
```

Test 6

Verdict: ACCEPTED

input

```
1000 1000
###MMMMMMMMMMMMMMMMMMMMMMMMMMM...
```

correct output

```
NO
```

user output

```
NO
```

Test 7

Verdict: ACCEPTED

input

```
15 15
###############
#M#...#...#...#
#A#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
...
```

correct output

```
YES
96
DDDDDDDDDDDRRUUUUUUUUUUUURRDDD...
```

user output

```
YES
96
DDDDDDDDDDDRRUUUUUUUUUUUURRDDD...
```

Test 8

Verdict: ACCEPTED

input

```
999 999
##############################...
```

correct output

```
YES
498000
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD...
```

user output

```
YES
498000
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD...
```

Test 9

Verdict: ACCEPTED

input

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

correct output

```
YES
1
```

```
L
```

user output

```
YES
1
L
```

Test 10

Verdict: ACCEPTED

input

```
5 1000
##############################...
```

correct output

```
YES
599
URRRRRRRRRRRRRRRRRRRRRRRRRRRRR...
```

user output

```
YES
599
URRRRRRRRRRRRRRRRRRRRRRRRRRRRR...
```

Test 11

Verdict: ACCEPTED

input

```
1 1
A
```

correct output

```
YES
0
```

user output

```
YES
0
```

Test 12

Verdict: ACCEPTED

input

```
4 4
```

```
#.#M
#.##
#A..
####
```

correct output

```
YES
2
UU
```

user output

```
YES
2
UU
```

Test 13

Verdict: ACCEPTED

input

```
3 3
MMM
MAM
MMM
```

correct output

```
NO
```

user output

```
NO
```

Test 14

Verdict: ACCEPTED

input

```
4 4
####
..A#
#.##
#M##
```

correct output

```
YES
2
LL
```

user output

```
YES
```

```
2
LL
```

Test 15

Verdict: ACCEPTED

input

```
4 4
####
#.A#
#M..
####
```

correct output

```
NO
```

user output

```
NO
```

Test 16

Verdict: ACCEPTED

input

```
1 3
##A
```

correct output

```
YES
0
```

user output

```
YES
0
```

Test 17

Verdict: ACCEPTED

input

```
1000 1000
M.M.M.M.M.M.M.M.M.M.M.M.M.M.M....
```

correct output

```
NO
```

user output

```
NO
```

Test 18

Verdict: ACCEPTED

input

```
14 7
#######
#.....#
#.###.#
#.###.#
...
```

correct output

```
YES
7
URRRRDR
```

user output

```
YES
7
URRRRDR
```

Test 19

Verdict: ACCEPTED

input

```
1000 1000
#....................#...........
```

correct output

```
NO
```

user output

```
NO
```

Test 20

Verdict: ACCEPTED

input

```
5 4
.#.M
#MA.
```

```
.MM.
.M.#
...
```

correct output

```
NO
```

user output

```
NO
```

Test 21

Verdict: ACCEPTED

input

```
10 6
######
#...##
#.#.##
#.#.##
...
```

correct output

```
NO
```

user output

```
NO
```

Test 22

Verdict: ACCEPTED

input

```
3 3
.#.
#A#
.#.
```

correct output

```
NO
```

user output

```
NO
```

Test 23

Verdict: ACCEPTED

input

```
6 5
#####
#A..#
#...#
#.###
...
```

correct output

```
YES
4
DDDD
```

user output

```
YES
4
DDDD
```

Test 24

Verdict: ACCEPTED

input

```
3 3
.##
MA#
###
```

correct output

```
NO
```

user output

```
NO
```

Test 25

Verdict: ACCEPTED

input

```
2 2
##
#A
```

correct output

```
YES
0
```

user output

```
YES
0
```