CSES Problem Set

Monsters

Submission details

Task:	Monsters
Sender:	Rodry
Submission time:	2021-11-28 10:23:45
Language:	C++11
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	0.05 s	»
#6	ACCEPTED	0.05 s	»
#7	ACCEPTED	0.01 s	»
#8	ACCEPTED	0.07 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	»

Graph Algorithms

...

Message Route –

Building Teams –

Round Trip –

Monsters ✓

Shortest Routes I –

Shortest Routes II –

High Score –

Flight Discount –

...

Your submissions

2021-11-28 10:23:45 ✓

2021-11-28 10:23:09 ✗

2021-11-28 10:21:27 ✗

2021-11-28 09:48:10 ✗

2021-11-28 09:45:43 ✗

test	verdict	time	
#15	ACCEPTED	0.01 s	»
#16	ACCEPTED	0.01 s	»
#17	ACCEPTED	0.05 s	»
#18	ACCEPTED	0.01 s	»
#19	ACCEPTED	0.06 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	»

Code ▲

```
#include<bits/stdc++.h>

#define INF 999999
#define MAX 2000
using namespace std;

queue<pair<int, int>> cola;
int cam[MAX][MAX];
pair<int, int> puntos[MAX][MAX], aux;
string resp;
bool flag = false;
bool flag2 = false;

void busqueda_anchura(int n, int m);

void aniadir(int n, int m){
        busqueda_anchura(n, m);
        flag2 = true;
        puntos[aux.first][aux.second] = pair<int, int>(0,0);
        cam[aux.first][aux.second] = 0;
        cola.push(aux);
        busqueda_anchura(n, m);

}

void res(){
```

```
        //Se muestra al rev►s
        if(flag){
                reverse(resp.begin(), resp.end());
                cout<<resp<< endl;
        }
        else {
                cout<<"NO"<<endl;
        }
}


void back_track(pair<int, int> nodo_par);

void busqueda_anchura(int n, int m){
        while(!cola.empty()){
                pair<int, int> aux = cola.front(), next;
                cola.pop();


                next = aux;
                next.first++;


                int pl = cam[aux.first][aux.second];
                if(pl+1<cam[next.first][next.second]){
                        cam[next.first][next.second]  = pl+1;
                        cola.push(next);
                        puntos[next.first][next.second] = aux;
                }


                next = aux;
                next.first--;


                pl = cam[aux.first][aux.second];
                if(pl+1<cam[next.first][next.second]){
                        cam[next.first][next.second]  = pl+1;
                        cola.push(next);
                        puntos[next.first][next.second] = aux;
                }


                next = aux; next.second++;
                pl = cam[aux.first][aux.second];
                if(pl+1<cam[next.first][next.second]){
```

```
                        cam[next.first][next.second]  = pl+1;
                        cola.push(next);
                        puntos[next.first][next.second] = aux;
                }


                next = aux;
                next.second--;
                pl = cam[aux.first][aux.second];
                if(pl+1<cam[next.first][next.second]){
                        cam[next.first][next.second]  = pl+1;
                        cola.push(next);
                        puntos[next.first][next.second] = aux;
                }


                if(flag2 && (aux.first == 1 || aux.second == 1 || aux.first == r

                        cout<<"YES"<< endl;
                        cout<<cam[aux.first][aux.second]<<endl;

                        back_track(aux);
                        flag = true;
                        return;
                }

        }
}
int main() {

        int n, m;
        string input;
        cin>>n>>m;
        for(int i = 1; i <= n; i++){

                cin >> input;
                for(int j = 1; j <= m; j++){
                        cam[i][j] = INF;
                        if(input[j-1]== '#') {
                                cam[i][j] = 0;
                        }
                        if(input[j-1]== 'M') {
                                cola.push(pair<int, int>(i,j));
                                cam[i][j]  = 0;
                        }
                        if(input[j-1]== 'A') {
```

```
                                aux.first = i;
                                aux.second = j;
                        }
                }
        }

        aniadir(n,m);

        res();
}


void back_track(pair<int, int> nodo_par){
        pair<int, int> origen;
        origen = puntos[nodo_par.first][nodo_par.second];
        if(origen ==  pair<int, int>(0,0))
                return;

        if(origen.first == nodo_par.first-1)
                resp.push_back('D');
        if(origen.first == nodo_par.first+1)
                resp.push_back('U');
        if(origen.second==nodo_par.second-1)
                resp.push_back('R');
        if(origen.second == nodo_par.second+1)
                resp.push_back('L');

        back_track(origen);

}
```

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
###MMMMMMMMMMMMMMMMMMMMMMMMMMMM...
```

correct output

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

user output

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

Test 6

Verdict: ACCEPTED

input

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

correct output

```
NO
```

user output

```
NO
```

Test 7

Verdict: ACCEPTED

input

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

correct output

```
YES
96
DDDDDDDDDDDRRUUUUUUUUUUUUURRDDD...
```

user output

```
YES
96
DDDDDDDDDDDRRUUUUUUUUUUUUURRDDD...
```

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
URRRRRRRRRRRRRRRRRRRRRRRRRRRR...
```

user output

```
YES
```

```
599
URRRRRRRRRRRRRRRRRRRRRRRRRRRR...
```

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.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
```