

Sep 25, 12 13:50

pacman.cpp

Page 1/2

```

/* FILE: pacman.cpp last change: 23-July-2012 author: Romeo Rizzi
 * This program is a solver for problem 2 (pacman) in 23-07-2012 exam in Algorit
hms
 */

// #include <iostream> // NOTA: ho commentato questa riga per essere sicuro di
non aver lasciato operazioni di input/output di debug nella fretta di consegnare
, se compila cosi' non ci sono ed io non perdo stupidamente i miei punti-esame.
#include <cstdlib>
#include <fstream>
#include <cassert>

using namespace std;

const int MAX_M = 1000;
const int MAX_N = 1000;
int m, n;
char cell[MAX_M+2][MAX_N+2]; // +2 perche' intendo bordare tutta la gr
iglia in input con una cornice di celle muro
long long int num[MAX_M+2][MAX_N+2]; // per evitare di dover trattare distinta
mente le celle ai margini della griglia // (puo' essere visto come un esempio di
uso di sentinelle in programmazione)
long long int num_sol;

/* template <class genType>
void displayMatrix(genType mat[MAX_M+2][MAX_N+2]) {
    for(int i = 1; i <= m; i++) {
        for(int j = 1; j <= n; j++)
            cout << mat[i][j] << " ";
        cout << endl;
    } cout << endl;
}*/

char aux[MAX_M+2][MAX_N+2];
void spreadF() { // i fantasmi trapassano anche i muri, e dove i fantasmi poi ar
rivano portano la morte
    for(int i = 1; i <= m; i++)
        for(int j = 1; j <= n; j++)
            aux[i][j] = cell[i][j];
    for(int i = 1; i <= m; i++)
        for(int j = 1; j <= n; j++)
            if( (aux[i][j] == 'F') | (aux[i+1][j] == 'F') | (aux[i-1][j] == 'F') | (au
x[i][j+1] == 'F') | (aux[i][j-1] == 'F') ) {
                cell[i][j] = 'F'; // anche se era muro o pillolazzo
                num[i][j] = 0; // la muerte
            }
}

long long int aux2[MAX_M+2][MAX_N+2];
void spreadNum() { // si pensi al singolo percorso safe che cresce, e di far s
parire i fantasmi ed essere vincolati solo dai muri
    for(int i = 1; i <= m; i++)
        for(int j = 1; j <= n; j++)
            aux2[i][j] = num[i][j];
    for(int i = 1; i <= m; i++)
        for(int j = 1; j <= n; j++)
            if( cell[i][j] != 'W' )
                num[i][j] += aux2[i + 1][j] + aux2[i][j + 1] + aux2[i - 1][j] + aux2[i][
j - 1];
}

int main() {
    ifstream fin("input.txt"); assert(fin);
    fin >> m >> n;
    for(int i = 0; i <= m+1; i++)
        for(int j = 0; j <= n+1; j++)
            cell[i][j] = 'W'; // mettiamo ovunque muro, in particolare sul contorn

```

Sep 25, 12 13:50

pacman.cpp

Page 2/2

```

o (dove servira' come sentinella)
for(int i = 1; i <= m; i++)
    for(int j = 1; j <= n; j++) {
        fin >> cell[i][j]; while( cell[i][j] == ' ' ) fin >> cell[i][j];
        if( cell[i][j] == 'M' ) num[i][j] = 1; // ho un percorso safe di lunghe
zza 0 che porta in [i][j]
        else num[i][j] = 0;
    }
fin.close();
// displayMatrix(cell); displayMatrix(num);

// INVARIANTI:
num_sol = 0; int len = 0; // num_sol = numero di percorsi safe di lunghezza a
l piu' len che arrivano a un qualche pillolazzo
// num[i][j] = numero di percorsi safe di lunghezza
esattamente len che arrivano in [i][j]

bool still_makes_sense = true;
while( still_makes_sense ) {
    spreadNum(); // displayMatrix(num);
    spreadF(); // displayMatrix(cell);
    still_makes_sense = false;
    for(int i = 1; i <= m; i++)
        for(int j = 1; j <= n; j++)
            if( cell[i][j] == 'P' ) {
                num_sol += num[i][j];
                still_makes_sense = true;
            }
    len++;
}

ofstream fout("output.txt"); assert(fout);
fout << num_sol << endl;
fout.close();
return 0;
}

```