

Colegiul Național „Roman-Vodă”  
Roman

Jocul Minesweeper  
Lucrare pentru obținerea atestatului profesional la informatică

Candidat,  
Begu Maria-Florina

Coordonator,  
Profesor Moldovanu Florin

2019

## Cuprins

1. Argument	pg. 3
2. Prezentarea jocului	pg. 4
3. Prezentarea programului	pg. 9
4. Rularea jocului	pg. 16
5. Bibliografie	pg. 20

## Argument

Minesweeper este un joc cunoscut de majoritatea oamenilor. Deși regulile jocului nu sunt atât de cunoscute, multe persoane se bucură să îl joace din curiozitate, mai ales în lipsa conexiunii la internet.

Am ales această temă deoarece este unul dintre jocurile copilăriei mele. Prima dată când l-am descoperit, nu am înțeles foarte bine regulile, dar totuși era o activitate interesantă, nouă. Mai târziu, am învățat și modul în care se joacă, regulile sale, fapt care a dus la apariția unei dependențe față de el atunci când nu aveam conexiune la internet, când eram la țară. Mi s-a părut atât de interesant prin modul în care antrenează atenția, în care uneori trebuie să te bazezi și pe noroc, dar în majoritatea timpului pe deducție. De aceea, a ajuns să mă fascineze cum prin simplitatea lui, acest joc reușește să fie foarte interesant și foarte atractiv pentru oricine încearcă să îl joace.

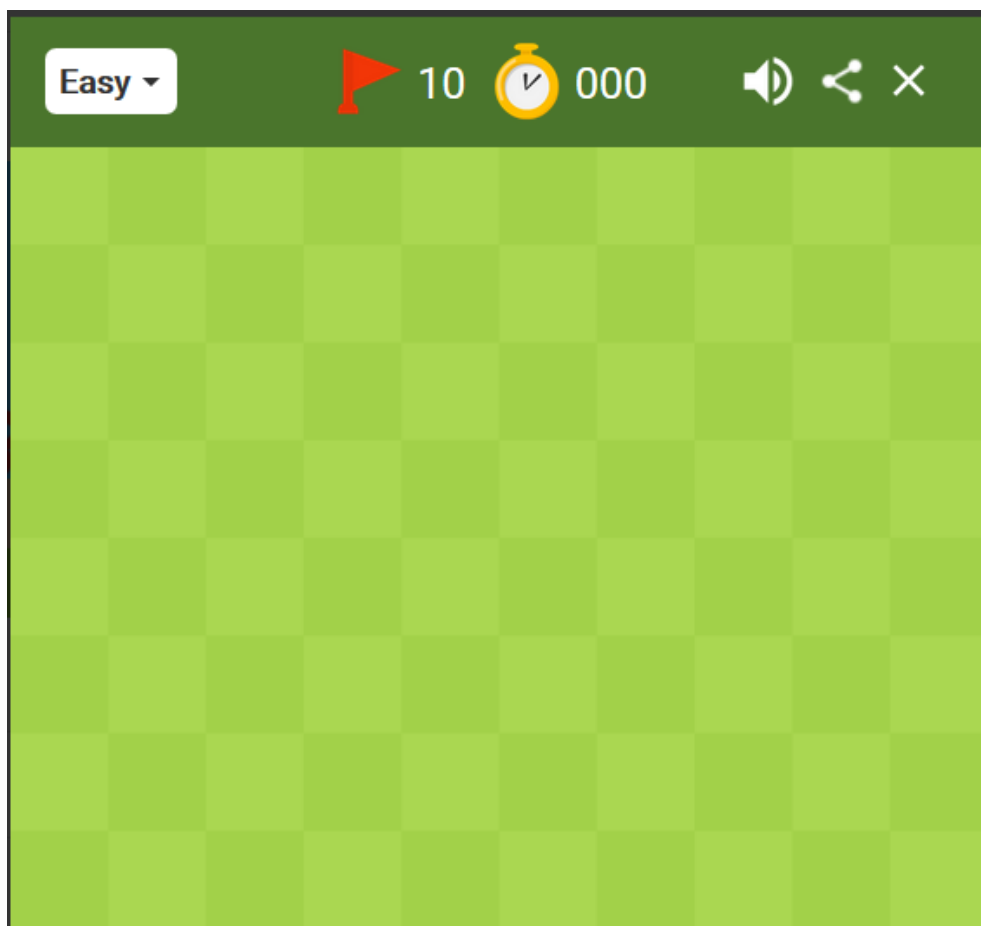
## Prezentarea jocului

Minesweeper este un joc video în care obiectivul principal este „curățarea” unei table pătrate ce conține bombe ascunse, fără a detona niciuna dintre acestea. Astfel, pentru a reuși să completezi jocul, trebuie să utilizezi o serie de indicii care îți oferă informații despre numărul bombelor aflate în vecinătate. Jocul a fost creat în anii 1960 și de-a lungul timpului s-au creat o multitudine de versiuni ale acestuia, disponibile pe diverse platforme.

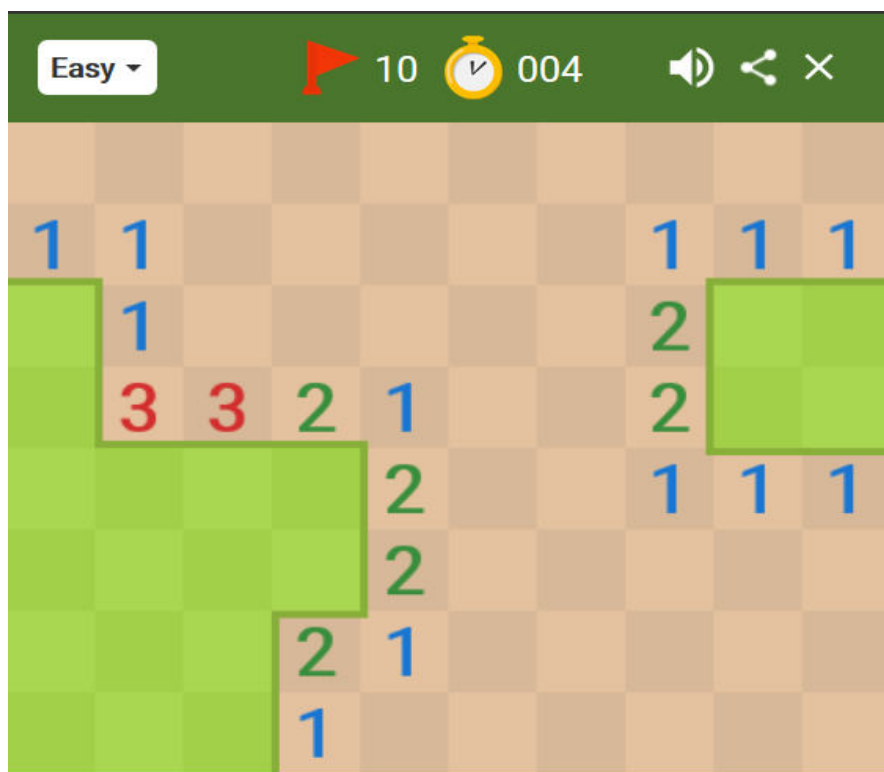
### Desfășurarea jocului

(pentru o mai bună înțelegere a jocului au fost utilizate imagini ale jocului Minesweeper realizat de google)

Inițial, tabla este acoperită și jucătorul trebuie să apese aleatoriu pe una din pătrățele.



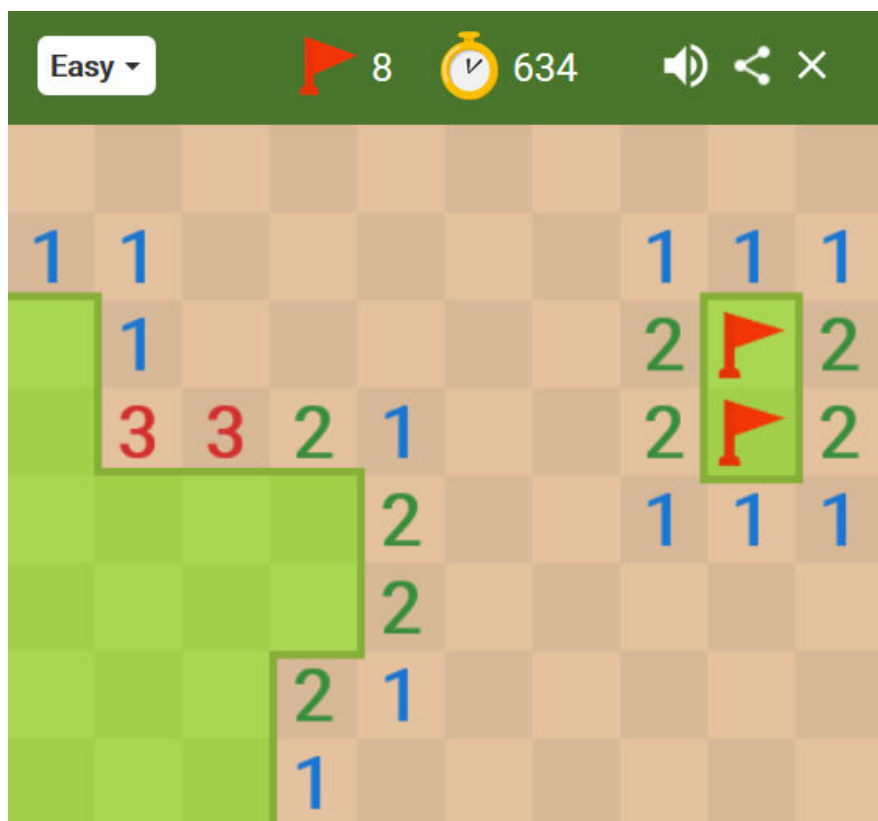
Dacă nu este o bombă, va apărea un spațiu gol, la marginile căruia vor fi prezente diferite numere ce vor oferi informații despre numărul bombelor din vecinătate.



Acolo unde suntem siguri de existența unei bombe, marcăm pătrățelul cu un steguleț.



Pe urmă se continuă procedeul până când sunt marcate toate bombele, restul căsuțelor fiind eliberate.









## Prezentarea programului

Programul realizat constă în realizarea jocului utilizând limbajul C++. Astfel, am creat o versiune ce poate fi jucată utilizând consola. În construirea programului am utilizat două matrici (matricea `a[11][11]` de tip `int` în care au fost păstrate indiciile cu privire la numărul bombelor din vecinătate, pozițiile bombelor care au fost marcate cu -1 și matricea `b[11][11]` de tip `char` care reprezintă matricea ce va apărea în consolă la fiecare mutare a jucătorului), doi vectori de coordonate pentru a ușura procesul de numărare a vecinilor ce ar putea fi bombe, precum și un vector de coordonate în care am memorat pozițiile bombelor.

```
#include <iostream>
#include <time.h>
#include <stdlib.h>

using namespace std;

int a[11][11];
char b[11][11];

struct coada
{
    int x,y;
}c[1005];

int dx[]={-1,-1,-1,0,0,1,1,1};
int dy[]={-1,0,1,-1,1,-1,0,1};
```

Funcții utilizate:

### 1. pozbombe

Sunt generate aleatoriu pozițiile celor 10 bombe din joc (acest lucru se realizează la fiecare rulare a programului). Am verificat ca aceste poziții să nu se repete și am marcat în matricea `a[11][11]` cu -1 pozițiile pe care se găsesc bombele.

```
void pozbombe()
{
    int t,i,x,y,j;
    t=time(0);
    srand(t);
    for(i=1;i<=10;++i)
    {
        x=rand()%10+1;
        y=rand()%10+1;
        if(a[x][y]==-1)
            --i;
        else
        {
            a[x][y]=-1;
        }
    }
}
```

## 2. nrbombe

Sunt numărate bombele vecine, pe cele 8 direcții, unei poziții date.

```
int nrbombe(int i, int j)
{
    int nr=0;
    if(a[i-1][j-1]==-1) ++nr;
    if(a[i-1][j]==-1) ++nr;
    if(a[i-1][j+1]==-1) ++nr;
    if(a[i][j-1]==-1) ++nr;
    if(a[i][j+1]==-1) ++nr;
    if(a[i+1][j-1]==-1) ++nr;
    if(a[i+1][j]==-1) ++nr;
    if(a[i+1][j+1]==-1) ++nr;
    return nr;
}
```

## 3. crearea

Este creată matricea a[11][11], punându-se pe fiecare poziție care nu este marcată cu -1 numărul de bombe vecine acesteia.

```
void crearea()
{
    int i,j;
    for(i=1;i<=10;++i)
        for(j=1;j<=10;++j)
            if(a[i][j]!=-1) a[i][j]=nrbombe(i,j);
}
```

## 4. creareb

Este creată matricea b[11][11], punându-se pe fiecare poziție semnul întrebării „?”.

```
void creareb()
{
    int i,j;
    for(i=1;i<=10;++i)
        for(j=1;j<=10;++j)
            b[i][j]=63;
}
```

## 5. afis

Este afișată matricea a[11][11].

```
void afis()
{
    int i, j;
    for(i=1;i<=10;++i)
    {
        for(j=1;j<=10;++j)
            cout<<a[i][j]<<' ';
        cout<<'\\n';
    }
}
```

#### 6. afisb

Este afișată matricea b[11][11], dacă ok=1.

```
void afisb(int ok)
{
    if(ok)
    {
        int i,j;
        cout<<" | ";
        for(j=1;j<=10;++j)
            cout<<j<<" ";
        cout<<"\n"<<"_|_";
        for(j=0;j<=9;++j)
            cout<<"____";
        cout<<"\n"<<" | "<<"\n";
        for(i=1;i<=9;++i)
        {
            cout<<i<<" | ";
            for(j=1;j<=10;++j)
            {
                cout<<b[i][j]<<" ";
            }
            cout<<"\n"<<" | "<<"\n";
        }
        cout<<"10| ";
        for(j=1;j<=10;++j)
            cout<<b[10][j]<<" ";
        cout<<"\n";
    }
}
```

#### 7. interior

Verifică dacă o poziție este în interiorul matricii.

```
int interior(int x, int y)
{
    return(x>=1&& x<=10&& y>=1&& y<=10);
}
```

#### 8. elib

Eliberează căsuțele în urma alegerii unei poziții de către jucător, dacă aceasta nu este bombă. Această funcție are la bază Algoritmul lui Lee care este modificat parțial conform cu cerințele jocului. Astfel, dacă un pătrățel este marcat cu 0, ceea ce ne indică faptul că nu are vecini bombe, acesta va fi afișat și va fi pusă în coadă, urmând ca și vecinii săi să fie eliberați. Altfel, dacă un pătrățel are vecini bombe, atunci acesta va fi afișat, dar nu va mai fi pus în coadă, deoarece vecinii săi nu trebuie afișați.

```

void elib(int x, int y)
{
    int p,u,i,xv,yv;
    p=u=1;
    c[p].x=x;
    c[p].y=y;
    b[x][y]=a[x][y]+'0';
    a[x][y]=-2;
    while(p<=u)
    {
        x=c[p].x;
        y=c[p++].y;
        for(i=0;i<=7;++i)
        {
            xv=x+dx[i];
            yv=y+dy[i];
            if(interior(xv,yv))
            {
                if(a[xv][yv]==0)
                {
                    c[++u].x=xv;
                    c[u].y=yv;
                    b[xv][yv]=a[xv][yv]+'0';
                    a[xv][yv]=-2;
                }
                else if(a[xv][yv]>0)
                {
                    b[xv][yv]=a[xv][yv]+'0';
                    a[xv][yv]=-2;
                }
            }
        }
    }
}

```

#### 9.decizie

Afișează jucătorului deciziile pe care le poate lua și în funcție de acestea, subprogramul va apela celelate subprograme sau va afișa mesajul „Era o bomba. Ati pierdut! o(>~<)o ” în care acesta apasă pe o pătrățică greșită (acest lucru duce la terminarea jocului). Trebuie menționat că jucătorul trebuie să fie atent cum poziționează stegulețele, deoarece acestea nu pot fi eliminate ulterior.

```

void decizie(int &ok,int &okw)
{
    cout<<'\\n'<<"a. Marcare bomba (a) ";
    cout<<'\\n'<<"b. Eliberare casuta (b) ";
    cout<<'\\n'<<"c. Renuntare la joc (orice alta tasta)";
    cout<<'\\n'<<"Raspunsul dumneavoastra : ";
    char c;
    int i,j;
    cin>>c;
    if(c=='a')
    {
        cout<<'\\n'<<"a. Introducere coordonate bomba (a) ";
        cout<<'\\n'<<"b. Intoarcere la pasul anterior (b) ";
        cout<<'\\n'<<"c. Renuntare la joc (orice alta tasta)";
        cout<<'\\n'<<"Raspunsul dumneavoastra : ";
        char cl;
        cin>>cl;
        if(cl=='a')
        {
            cout<<'\\n'<<" Introducere coordonate bomba : ";
            cin>>i>>j;
            cout<<'\\n';
            b[i][j]='F';
            a[i][j]=-3;
            afisb(ok);
        }
        else if(cl=='b')
        {
            afisb(ok);
            decizie(ok, okw);
        }
        else
        {
            cout<<"EROARE!";
            ok=0;
            okw=0;
        }
    }
    else if(c=='b')
    {
        cout<<'\\n'<<"a. Introducere coordonate casuta goala (a) ";
        cout<<'\\n'<<"b. Intoarcere la pasul anterior (b) ";
        cout<<'\\n'<<"c. Renuntare la joc (orice alta tasta)";
        cout<<'\\n'<<"Raspunsul dumneavoastra : ";
        char c2;
        cin>>c2;
        if(c2=='a')
        {
            cout<<'\\n'<<" Introducere coordonate casuta goala : ";
            cin>>i>>j;
            cout<<'\\n';

```

```

        if(a[i][j]==-1)
        {
            cout<<"Era o bomba. Ati pierdut! o(>~<)o ";
            cout<<"\n";
            ok=0;
            okw=0;
        }
        else if(a[i][j]!=-2)
        {
            elib(i,j);
            ok=1;
            afisb(ok);
        }
    }
    else if(c2=='b')
    {
        afisb(ok);
        decizie(ok, okw);
    }
    else
    {
        cout<<"EROARE!";
        ok=0;
        okw=0;
    }
}

else
{
    cout<<"EROARE!";
    ok=0;
    okw=0;
}
}

```

#### 10. verifbomb

Se verifică dacă în matricea a[11][11] mai sunt prezente bombe.

```
int verifbomb(int &ok)
{
    int i,j;
    ok=0;
    for(i=1;i<=10&&!ok;++i)
        for(j=1;j<=10&&!ok;++j)
            if(a[i][j]==-1) ok=1;
    return ok;
}
```

#### 11. main

Sunt apelate funcțiile 1,3,4,6. De asemenea, aici sunt utilizate două variabile ok și okw care ne ajută să verificăm când și cât timp trebuie apelate funcțiile 9,10. Dacă la final okw este 1, lucru care înseamnă ca nicio bombă nu a fost atinsă și că nu mai sunt bombe prezente în matricea a[11][11] (acest lucru se întâmplă când se întâmplă când ok devine 0 și structura repetitivă se încheie), se afișează mesajul „Bombele au fost eliminate. Ati castigat! \(^o^)/”.

```
int main()
{
    int ok=1,okw=1;
    pozbombe();
    crearea();
    creareb();
    afisb(ok);
    cout<<'\\n';
    while(ok)
    {
        if(okw==1)
            decizie(ok,okw);
        else break;
        if(verifbomb(ok)==0) break;
        cout<<'\\n';
    }
    if(okw) cout<<"Bombele au fost eliminate. Ati castigat! \(^o^)/ ";
    return 0;
}
```

## Rularea jocului

În continuare, vor fi prezentate câteva secvențe de pe parcursul programului rulat care constă în jocul Minesweeper jucat în consolă.

Este important de menționat faptul că o bombă nu poate fi demarcată. Astfel, atunci când marcăm o căsuță trebuie să fim siguri că acolo se află o bombă, altfel, jocul nu va funcționa corect.

```
| 1 2 3 4 5 6 7 8 9 10
|-----
1 | ? ? ? ? ? ? ? ? ? ?
2 | ? ? ? ? ? ? ? ? ? ?
3 | ? ? ? ? ? ? ? ? ? ?
4 | ? ? ? ? ? ? ? ? ? ?
5 | ? ? ? ? ? ? ? ? ? ?
6 | ? ? ? ? ? ? ? ? ? ?
7 | ? ? ? ? ? ? ? ? ? ?
8 | ? ? ? ? ? ? ? ? ? ?
9 | ? ? ? ? ? ? ? ? ? ?
10| ? ? ? ? ? ? ? ? ? ?

a. Marcare bomba (a)
b. Eliberare casuta (b)
c. Renuntare la joc (orice alta tasta)
Raspunsul dumneavoastra :
```



2	?	?	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?	?	?

a. Marcare bomba (a)  
b. Eliberare casuta (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : b

a. Introducere coordonate casuta goala (a)  
b. Intoarcere la pasul anterior (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : a

Introducere coordonate casuta goala : 5 5

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	?	1	0	1	?	?
2	0	1	1	2	1	1	0	1	?	?
3	0	1	?	1	0	0	0	1	?	?
4	0	1	1	1	0	1	1	1	?	?
5	0	0	0	0	0	1	?	?	?	?
6	0	0	0	0	0	1	?	?	?	?
7	1	1	1	0	1	1	?	?	?	?
8	?	?	1	0	1	?	?	?	?	?
9	?	?	2	0	1	2	?	?	?	?
10	?	?	1	0	0	1	?	?	?	?

a. Marcare bomba (a)  
b. Eliberare casuta (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra :

a. Marcare bomba (a)  
b. Eliberare casuta (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : a

a. Introducere coordonate bomba (a)  
b. Intoarcere la pasul anterior (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : a

Introducere coordonate bomba : 3 3

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	?	1	0	1	?	?
2	0	1	1	2	1	1	0	1	?	?
3	0	1	F	1	0	0	0	1	?	?
4	0	1	1	1	0	1	1	1	?	?
5	0	0	0	0	0	1	?	?	?	?
6	0	0	0	0	0	1	?	?	?	?
7	1	1	1	0	1	1	?	?	?	?
8	?	?	1	0	1	?	?	?	?	?
9	?	?	2	0	1	2	?	?	?	?
10	?	?	1	0	0	1	?	?	?	?

a. Marcare bomba (a)  
b. Eliberare casuta (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : a

a. Introducere coordonate bomba (a)  
b. Intoarcere la pasul anterior (b)  
c. Renuntare la joc (orice alta tasta)  
Raspunsul dumneavoastra : a

Introducere coordonate bomba : 9 1

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	F	1	0	1	?	?
2	0	1	1	2	1	1	0	1	?	?
3	0	1	F	1	0	0	0	1	?	?
4	0	1	1	1	0	1	1	1	?	?
5	0	0	0	0	0	1	?	?	?	?
6	0	0	0	0	0	1	?	?	?	?
7	1	1	1	0	1	1	?	?	?	?
8	2	F	1	0	1	?	?	?	?	?
9	F	3	2	0	1	2	?	?	?	?
10	?	?	1	0	0	1	?	?	?	?

```

a. Marcare bomba (a)
b. Eliberare casuta (b)
c. Renuntare la joc (orice alta tasta)
Raspunsul dumneavoastra : b

a. Introducere coordonate casuta goala (a)
b. Intoarcere la pasul anterior (b)
c. Renuntare la joc (orice alta tasta)
Raspunsul dumneavoastra : a

Introducere coordonate casuta goala : 7 7

```

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	F	1	0	1	?	?
2	0	1	1	2	1	1	0	1	?	?
3	0	1	F	1	0	0	0	1	1	1
4	0	1	1	1	0	1	1	1	0	0
5	0	0	0	0	0	1	?	1	0	0
6	0	0	0	0	0	1	1	1	0	0
7	1	1	1	0	1	1	1	0	0	0
8	2	F	1	0	1	F	2	1	1	1
9	F	3	2	0	1	2	?	?	?	?
10	2	F	1	0	0	1	?	?	?	?

```

a. Marcare bomba (a)
b. Eliberare casuta (b)
c. Renuntare la joc (orice alta tasta)
Raspunsul dumneavoastra : a

a. Introducere coordonate bomba (a)
b. Intoarcere la pasul anterior (b)
c. Renuntare la joc (orice alta tasta)
Raspunsul dumneavoastra : a

Introducere coordonate bomba : 9 7

```

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	F	1	0	1	1	1
2	0	1	1	2	1	1	0	1	F	1
3	0	1	F	1	0	0	0	1	1	1
4	0	1	1	1	0	1	1	1	0	0
5	0	0	0	0	0	1	F	1	0	0
6	0	0	0	0	0	1	1	1	0	0
7	1	1	1	0	1	1	1	0	0	0
8	2	F	1	0	1	F	2	1	1	1
9	F	3	2	0	1	2	F	1	1	F
10	2	F	1	0	0	1	1	1	1	1

Bombele au fost eliminate. Ati castigat! (^o^)/

### Bibliografie

1. Minesweeper Google
2. [https://en.wikipedia.org/wiki/Minesweeper\\_\(video\\_game\)](https://en.wikipedia.org/wiki/Minesweeper_(video_game))
3. Emanuela Cerchez, Marinel Șerban, „Programare în limbajul C/C++ pentru liceu”
4. <http://www.cplusplus.com/doc/tutorial/>