

```

#include
#include
#include
#include
#include
#include
#include

#include "../include/Cellule.h"
#include "../include/Grille.h"

using namespace std;

Grille::Grille()
{
    srand(time(NULL));
    for(int i = 0; i < N; i++){ cells.push_back(vector(N)); /* ajouts de ligne N cases / } for(int i = 0; i < N;
    i++){ int x = rand() % N; int y = rand() % N; if(cells[x][y].getEtat() != true) cells[x][y] = true; }
    file.open(fichier, fstream::in | fstream::out); if(file){ ecrire(); }else { cerr<<"erreur lors=" de="
    l'ouverture=" du=" fichier"<
    Grille::~~Grille(){
    file.close();
    }*/

    void Grille::ecrire(){
    //ofstream file(fichier.c_str());
    string chaine = "";
    if(file){
    file << "$";
    chaine.append(1, "\n");
    for(int i = 0; i < N; i++){
    for(int j = 0; j < N; j++){
    chaine.append(1, (bool)cells[i][j].getEtat());
    }
    chaine.append(1, "\n");
    }
    file << chaine;
    file << "$";
    }else{
    cerr <<"erreur lors de l'ouverture du fichier"<<endl;
    }
    }

```

```

void Grille::lire(){
// ifstream file(fichier.c_str());
string chaine = "", ligne = "";
char delim;
int k = 0;
if(file){
file >> delim; // récupérer $
while(getline(file, chaine, delim)){
stringstream ss(chaine);
getline(ss,ligne,'\n'); // pour sautetr une ligne
while(getline(ss, ligne, '\n')){
for(int j = 0; j < N; j++){
cells[k][j].setEtat((bool)ligne[j]);
}
k++;
ligne = "";
}
}
}else{
cerr << "erreur lors de l'ouverture du fichier" <<endl;
}
}

```

```

void Grille::affiche(ostream& os) const
{
for(int i = 0; i < N; i++){
for(int j = 0; j < N; j++){
cells[i][j].affiche(os);
os<<" ";
}
os<<endl;
}
}

```

```

ostream& operator<<(ostream& os, Grille& p)
{
p.affiche(os);
return os;
}

```