1 Introduction

Les automates cellulaires sont des objets mathématiques étudiés en informatique fondamentale. On peut les voir comme des grilles régulières de cellules, où chaque cellule a un état à chaque temps. L'ensemble des états possibles est fini. L'état d'une cellule c au temps t+1 dépend uniquement du voisinage de c au temps t. À chaque pas de temps, les mêmes règles sont appliquées simultanément sur toutes les cellules de la grille, ce qui crée une nouvelle génération de cellules.

2 Un premier exemple d'automate

Afin de rendre cette définition plus concrète, intéressons-nous à un automate simple. Les cellules de cet automate sont dans une grille à une seule dimension et ne peuvent prendre que deux états : 0 ou 1. Ainsi, le voisinage de chaque cellule est constitué d'elle-même et de ses deux cellules adjacentes. Les règles permettant de générer une nouvelle génération de cellules sont définies dans la table 1.

Voisinage (t)	111	110	101	100	011	010	001	000
État central ($t+1$)	0	0	0	1	1	1	1	0

Table 1 – Règles de l'automate simple

Si on représente les cellules dans l'état 0 par des espaces et celles dans l'état 1 par des 'x', qu'on calcule et affiche les 4 premières générations issues de l'état initial 000010000 (une par ligne, en considérant que les cellules en dehors de la grille sont des 0), on obtient :

```
000010000:     x
000111000:    xxx
001100100:   xx  x
011011110:  xx xxxx
110010001: xx  x   x
```

2.1 Premier programme

Dans un fichier **automate1.c**, implémentez l'automate simple que l'on vient de décrire. On va pour cela fixer le nombres de cellules à $n = 41$ et stocker la grille dans un tableau de $n+2$ cases (les valeurs extrêmes représentent les cases autour de l'espace d'étude et doivent toujours avoir 0 pour valeur). À l'état initial, toutes les cellules doivent être à 0 sauf la cellule centrale. Demandez à l'utilisateur le nombre p de pas de temps à calculer, puis calculez et affichez les p premières générations de cellules. Note : afin de mettre facilement la grille à jour, vous pouvez utiliser une variable temporaire (un second tableau).

3 L'automate de Conway

L'automate de Conway, ou *jeu de la vie*, est un automate cellulaire célèbre conçu par John Conway en 1970. Dans celui-ci, la grille est à deux dimensions. Ainsi, le voisinage d'une cellule est la cellule elle-même et ses 8 cellules voisines. Il y a deux états possibles : 0 (on dit que la cellule est morte) et 1 (on dit que la cellule est vivante). Les règles de cet automate sont les suivantes :

— Une cellule morte possédant exactement trois voisines vivantes devient vivante.

— Une cellule vivante possédant deux ou trois voisines vivantes reste vivante ; sinon, elle meurt.

3.1 Deuxième programme

Dans un fichier **automate2.c**, implémentez l'automate de Conway que l'on vient de décrire. Pour cela, on fixe la largeur de la grille à $w = 20$ et sa hauteur à $h = 15$. On va stocker la grille dans le tableau suivant :

```
const int w = 20;
const int h = 15;
int grid[h+2][w+2]; // contour de zéros
// grid[1][1] sera affichée en haut à gauche
// w croît <=> on va vers la droite
// h croît <=> on va vers le bas
```

À l'état initial, les cellules doivent avoir une valeur aléatoire (une chance sur deux d'être vivante). Vous pouvez pour cela utiliser les fonctions définies dans hasard.h ou la bibliothèque standard. Demandez à l'utilisateur le nombre p de pas de temps à calculer, puis calculez et affichez les p premières générations de cellules. Afin de ralentir la simulation, vous pouvez vous servir de la fonction sleep (man 3 sleep). Afin de nettoyer l'écran entre chaque affichage de la grille, vous pouvez faire l'appel système suivant :

```
system("clear"); // "cls" sous Windows
```