

## TP 2 : matrices

Ce TP, non noté, est à rendre **au plus tard le 19 octobre**, après avoir indiqué votre nom en commentaire, en l'envoyant par mël à [dominique.py@univ-lemans.fr](mailto:dominique.py@univ-lemans.fr)

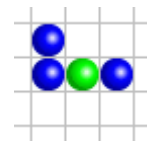
### Énoncé

Un automate cellulaire est constitué d'un ensemble de cellules qui évoluent par générations successives. On représente un automate cellulaire par une grille  $N \times N$  dont chaque case est occupée soit par une cellule vivante, soit par une cellule morte. Chaque case possède huit cases voisines (les quatre côtés plus les quatre coins). Les règles d'évolution sont les suivantes :

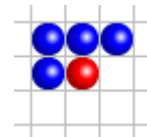
- Si la cellule est vivante et entourée par deux ou trois cellules vivantes, elle reste en vie à la génération suivante, sinon elle meurt.
- Si la cellule est morte et entourée par exactement trois cellules vivantes, elle naît à la génération suivante.

### Exemples

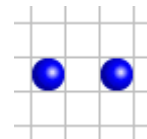
La cellule centrale, qui est morte, possède exactement trois voisins : elle sera vivante à la génération suivante



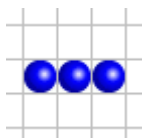
La cellule centrale, qui est vivante, possède quatre voisins : à la génération suivante, elle meurt



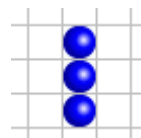
La cellule centrale est morte et ne possède que deux voisins : elle reste morte à la génération suivante



Ainsi, la configuration



donne à la génération suivante la configuration



qui redonne ensuite la première

Écrire un programme qui initialise un automate cellulaire à partir d'un fichier texte puis affiche les  $X$  premières générations de l'automate.

## Mise en œuvre

- Le fichier d'initialisation, supposé sans erreurs, contient une séquence d'entiers qui représentent les coordonnées des cellules vivantes de la première génération.

Par exemple, le fichier :

1 8 2 3 6 4

représente une grille contenant trois cellules vivantes de coordonnées (1,8), (2, 3) et (6, 4).

- Le nom du fichier d'initialisation et le nombre X de générations à afficher seront saisis au clavier par l'utilisateur.

- Pour calculer les générations successives, on utilise deux matrices : l'une pour représenter l'automate cellulaire, l'autre pour calculer le nombre de voisins de chaque cellule. À chaque génération, le calcul se fait en deux temps :

1) chaque cellule de la seconde matrice est initialisée avec le nombre de voisins de la cellule correspondante de la première matrice,

2) la génération suivante est calculée grâce aux valeurs de la seconde matrice et la première matrice est mise à jour.

Exemple (avec N=8) :

	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7
0		●							0	1	0	1	0	0	1	1	1
1							●		1	1	1	2	1	1	1	0	1
2				●					2	1	1	2	0	1	1	1	1
3		●							3	1	1	4	3	3	1	1	0
4			●	●		●			4	1	2	2	2	3	1	1	0
5					●				5	0	1	2	3	2	2	1	0
6									6	0	0	0	1	1	2	1	1
7							●		7	0	0	0	0	0	1	0	1

Matrice de l'automate cellulaire

Matrice du nombre de voisins

Pour améliorer l'affichage, on peut utiliser, sous Linux :

- l'instruction **system("clear")** ; qui efface l'écran (avant d'afficher la génération suivante),
- l'instruction **sleep(x)** ; qui fait une pause dans l'exécution pendant  $x$  secondes.

La fonction prédéfinie **system** est dans la bibliothèque **stdlib.h** qui doit être appelée dans le programme (avec `#include<stdlib.h>`). De même, la fonction prédéfinie **sleep** est dans la bibliothèque **unistd.h** qui doit être appelée dans le programme (avec `#include<unistd.h>`).