

Unité : INF1	Labo no : 8	Jeu de la vie
--------------	-------------	---------------

But

- Présentation du code aussi propre que possible
- Décomposition en bibliothèques et sous-programmes réutilisables
- Utilisation de tableaux C à 2 dimensions

A faire

Le jeu de la vie, inventé par J. H. Conway, utilise une grille rectangulaire de cellules, chacune d'entre elles pouvant contenir 0 ou 1 habitant.

Chaque cellule a jusqu'à huit et nous utiliserons « $occ(k)$ » pour désigner le nombre de cellules voisines de la cellule k qui sont occupées.

L'état d'une nouvelle génération est obtenu à partir de la précédente en appliquant deux règles simples :

- Un habitant situé dans la cellule k survit à la génération suivante si $occ(k) = 2$ ou 3 , et meurt dans le cas contraire.
- Il y aura une "naissance" dans une cellule k vide si $occ(k) = 3$, sinon la cellule restera vide.

Ecrire un programme en C++ qui, partant d'une configuration initiale donnée (cf. Indications), calcule un certain nombre de générations suivantes en application des règles ci-dessus et affiche chacune des configurations trouvées.

L'affichage sera fait en utilisant les symboles

- O cellule vide
- X cellule occupée

Contraintes

- Utiliser des tableaux C, les tailles sont connues à la compilation
- Ne pas utiliser de notation pointeur, comme par exemple ie $*(tab + indice)$
- Représenter les grilles comme dans les exemples ci-dessous
- Prévoir un mode DEBUG selon l'exemple ci-dessous
- Une fonction doit permettre de générer la prochaine génération en mettant à jour le tableau reçu en paramètre
- Une autre fonction doit permettre de générer N générations et s'arrêter automatique, selon un paramètre booléen, s'il aucun changement est détecté

Sans mode DEBUG

GENRATION NO 0									
	0	1	2	3	4	5	6	7	8
0
1
2	X
3	.	.	.	X	X	X	.	.	.
4	.	.	X	X	X	X	X	.	.
5	.	.	.	X	X	X	.	.	.
6	X
7
8

GENRATION NO 1									
	0	1	2	3	4	5	6	7	8
0
1
2	.	.	.	X	X	X	.	.	.
3	.	.	X	.	.	.	X	.	.
4	.	.	X	.	.	.	X	.	.
5	.	.	X	.	.	.	X	.	.
6	.	.	.	X	X	X	.	.	.
7
8

Avec mode DEBUG

GENRATION NO 0									
	0	1	2	3	4	5	6	7	8
0
1
2	X
3	.	.	.	X	X	X	.	.	.
4	.	.	X	X	X	X	X	.	.
5	.	.	.	X	X	X	.	.	.
6	X
7
8

```

tab[2][3] = 3 => naissance
tab[2][4] = 3 => survie
tab[2][5] = 3 => naissance
tab[3][2] = 3 => naissance
tab[3][3] = 5 => mort
tab[3][4] = 6 => mort
tab[3][5] = 5 => mort
tab[3][6] = 3 => naissance
tab[4][2] = 3 => survie
tab[4][3] = 6 => mort
tab[4][4] = 8 => mort
tab[4][5] = 6 => mort
tab[4][6] = 3 => survie
tab[5][2] = 3 => naissance
tab[5][3] = 5 => mort
tab[5][4] = 6 => mort
tab[5][5] = 5 => mort
tab[5][6] = 3 => naissance
tab[6][3] = 3 => naissance
tab[6][4] = 3 => survie
tab[6][5] = 3 => naissance

```

GENRATION NO 1									
	0	1	2	3	4	5	6	7	8
0
1
2	.	.	.	X	X	X	.	.	.
3	.	.	X	.	.	.	X	.	.
4	.	.	X	.	.	.	X	.	.
5	.	.	X	.	.	.	X	.	.
6	.	.	.	X	X	X	.	.	.
7
8

```

tab[1][4] = 3 => naissance
tab[2][3] = 2 => survie
tab[2][4] = 2 => survie
tab[2][5] = 2 => survie
tab[3][2] = 2 => survie
tab[3][4] = 3 => naissance
tab[3][6] = 2 => survie
tab[4][1] = 3 => naissance
tab[4][2] = 2 => survie
tab[4][3] = 3 => naissance
tab[4][5] = 3 => naissance
tab[4][6] = 2 => survie
tab[4][7] = 3 => naissance
tab[5][2] = 2 => survie
tab[5][4] = 3 => naissance
tab[5][6] = 2 => survie
tab[6][3] = 2 => survie
tab[6][4] = 2 => survie
tab[6][5] = 2 => survie
tab[7][4] = 3 => naissance

```