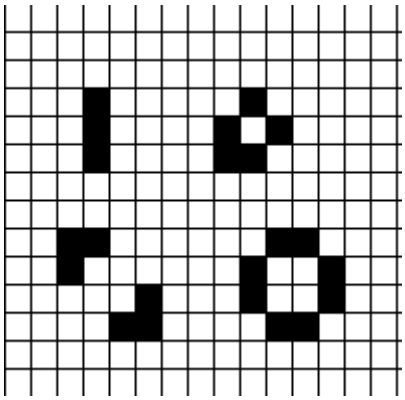
programmation
examen septembre

Le monde du jeu de la vie de Conway est un plan infini quadrillé dont chaque case est occupée par une cellule morte ou vivante. Chaque case possède huit voisines et, d'une génération à l'autre, des naissances et des décès s'y déroulent mécaniquement selon la règle simple suivante :

- si une case est occupée par une cellule morte et que trois de ses voisines sont occupées par des cellules vivantes, alors la cellule renaît ;
- si une case est occupée par une cellule vivante la survie n'y est possible que si deux ou trois cases voisines sont occupées par des cellules vivantes ; dans tous les autres cas, la cellule meurt.

Nous nous intéressons à la simulation de ce jeu dans un plan fini.



Question 0. (2 points)

Quel est l'état du monde ci-contre à la génération suivante.

Question 1. (2 points)

Ecrire la définition du constructeur et des fonctions *setVivant* et *estVivant* de la classe *Cellule*.

```
class Cellule {
  private :
    bool vivant;
  public :
    Cellule() ;
    void setVivant(bool b);
    bool estVivant();
};
```

Le monde du jeu de la vie est représenté par un tableau de *nbLignes* lignes et *nbColonnes* colonnes. Il y a deux façons d'envisager le voisinage des cellules des bords du monde :

- Les cellules du bord ont soit 5 voisins soit 3 voisins pour les coins. (*torique* = *false*)
- Les cellules du bord ont toutes 8 voisins : on considère la première ligne et la dernière lignes voisines, et la première colonne et la dernière colonne voisines. (*torique* = *true*)

```
class Monde {
  private :
     bool torique;
     int lignes, colonnes ;
     Cellule **rep;

  public :
    Monde (int l, int c) ;
    Monde (int l, int c, bool torique);
    ~Monde() ;

    int getLignes();
    int getColonnes();
    void genererAlea();

    int nbVoisinsVivants( int i, int j) ;
    int nbVoisinsVivantsTorique( int i, int j);

    void generationSuivante()
} ;
```

Question 2. (4 points)

Ecrire la définition des constructeurs et du destructeur de la classe *Monde.*

Question 3. (2 points)

Ecrire la définition des fonctions membres *getLignes()* et *getColonnes().*

Question 3. (4 points)

Ecrire la définition des fonctions membres *nbVoisinsVivants(i, j)* et *nbVoisinsVivantsTorique(i, j)* qui renvoient le nombre de voisins vivants de la cellule située en (*i, j*).

Question 4. (3 points)

Ecrire la définition de la fonction membre *genererAlea()* qui remplit le monde de façon aléatoire : en moyenne une cellule sur 3 sera vivante.

Question 5. (3 points)

Ecrire la définition de la fonction membre *generationSuivante()* qui calcule le monde à la génération suivante.