

# Algorithmique avancée et programmation C

Durée : 1h30

Documents autorisés : **AUCUN**

## Remarques :

- Vos réponses aux exercices 2 et 3 seront sur des copies doubles différentes
- Veuillez lire attentivement les questions avant de répondre ;
- Rendez une copie propre ;
- N'utilisez pas de crayon à papier sur votre copie ;
- Le pourcentage indiqué pour chaque exercice indique l'importance, approximative, de l'exercice dans le calcul de la note finale.

## 1 QCM sur le C (25%)

Répondez au qcm ci joint (à rendre avec votre copie). La note de chaque question peut varier de  $-1$  à  $+1$ .

Soit  $B$  le nombre de bonnes réponses à une question et soit  $M$  le nombre de mauvaises réponses à cette même question ( $B+M$  est égale au nombre de réponses à la question). Chaque bonne réponse cochée rapporte  $1/B$  points et chaque mauvaise réponse  $-1/M$  points.

## 2 *Shadows of the Knigh* (35%)

Cet exercice s'inspire du puzzle « Shadows of the Knigh - Episode 1 » de la plateforme Codingame.

### 2.1 Contexte

« Batman va rechercher les otages d'un bâtiment donné, en sautant de fenêtre en fenêtre à l'aide de son grappin. Le but de Batman est d'arriver sur la fenêtre de la pièce où les otages se trouvent afin de désamorcer la bombe du Joker. Malheureusement il n'a qu'un nombre de sauts limités avant que la bombe n'explose... »

Avant chaque saut, le détecteur fournira à Batman la direction des bombes par rapport à sa position actuelle :

**U** la bombe est située au dessus de Batman ;

**UR** la bombe est située au dessus et à droite de Batman ;

**R** la bombe est située à droite de Batman ;

**DR** la bombe est située en dessous et à droite de Batman ;

**D** la bombe est située en dessous de Batman ;

**DL** la bombe est située en dessous et à gauche de Batman ;

**L** la bombe est située à gauche de Batman ;

**UL** la bombe est située au dessus et à gauche de Batman. »

Votre mission consiste à donner un algorithme que suivra Batman pour secourir au plus vite les otages.

### 2.2 Analyse

On suppose posséder le type *Direction* :

— **Type**  $Direction = \{U,UR,R,DR,D,DL,L,UL\}$

Proposez le TAD *Batiment* (sans les parties Axiomes et Sémantiques) sachant que le bâtiment possède *hauteur*  $\times$  *largeur* pièces avec une fenêtre par pièce (cf. figure 1). Chaque pièce est identifiée par ses coordonnées. La pièce la plus en haut à gauche a pour coordonnées  $(0,0)$ . On peut savoir si une pièce donnée contient ou pas la bombe recherchée.

### 2.3 Conception préliminaire

1. Proposez les signatures des fonctions et procédures des opérations du TAD *Batiment* ;
2. Proposez la signature de la fonction *detecteur* qui fournit à Batman la direction des bombes par rapport à sa position actuelle.

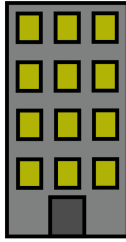


FIGURE 1 – Un exemple de bâtiment de hauteur 4 et de largeur 3

## 2.4 L'algorithme suivi par Batman

Pour aller au plus vite, Batman va utiliser un algorithme en  $O(\log_2(n))$  de la fonction suivante :  
**procédure** trouverBombe (**E** bat : Bâtiment ; xBatman, yBatman : **Naturel**, S xBombe, yBombe : **Naturel**)

1. Que représente  $n$  dans ce contexte ?
2. Donnez les préconditions de cette procédure.
3. Donnez l'algorithme itératif de cette procédure.

## 3 Fractale : courbes de Koch (40 %)

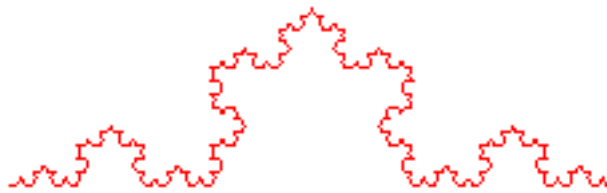


FIGURE 2 – Courbe de Koch de niveau 4

La courbe de Koch est une fractale (un dessin récursif) dont le détail varie en fonction d'un niveau donné  $n$ . La figure 2 (issue de <http://www.mathcurve.com/fractals/koch/koch.shtml>) présente la courbe de Koch de niveau 4. En fait, dessiner la courbe de Koch de niveau  $n$  entre deux points  $A(x_a, y_a)$  et  $E(x_e, y_e)$  revient à :

- dessiner le segment de droite  $[A, E]$  si  $n = 0$
- dessiner 4 courbes de Koch de niveau  $n - 1$  entre les points A,B,C,D et E (la première entre A et B, la deuxième entre B et C, la troisième entre C et D, et la dernière entre D et E) telles que (Cf. figure 3) :
  - $\overrightarrow{AB} = \frac{1}{3}\overrightarrow{AE}$ ,
  - $\overrightarrow{ED} = \frac{1}{3}\overrightarrow{EA}$ ,
  - les points B,C et D forment un triangle équilatéral.

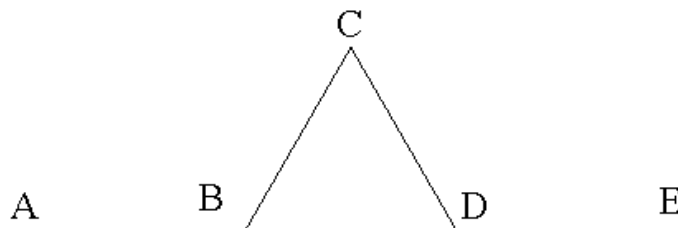


FIGURE 3 – Position des points B,C et D

La figure 4 présente quelques courbes de Koch.

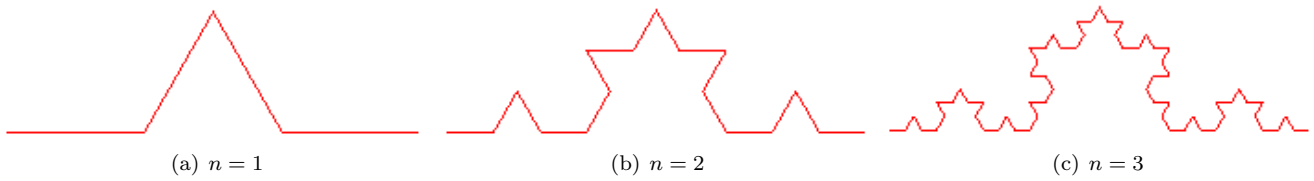


FIGURE 4 – Quelques courbes de Koch

### 3.1 Analyse

On suppose posséder le TAD *Point2D* vu en cours et le TAD *Graphique* suivant :

**Nom:** Graphique

**Utilise:** Point2D

**Opérations:** graphique:  $\rightarrow$  Graphique

ajouterSegment: Graphique  $\times$  Point2D  $\times$  Point2D  $\rightarrow$  Graphique

Donnez l'analyse descendante de l'opération *dessinerCourbeKoch* qui permet de dessiner une courbe de Koch de niveau  $n$  entre deux points sur un graphique.

### 3.2 Conception préliminaire

Donnez les signatures des fonctions et procédures de votre analyse descendante.

### 3.3 Conception détaillée

Donnez le corps de la fonction ou de la procédure correspondante à l'opération *dessinerCourbeKoch*.