

|   |   |
|---|---|
| REPUBLIQUE TUNISIENNE<br>MINISTERE DE L'EDUCATION<br><b>EXAMEN DU BACCALAUREAT</b><br><b>SESSION DE JUIN 2013</b> | Epreuve : <b>ALGORITHMIQUE ET PROGRAMMATION</b> |
|   | Durée : <b>3 h</b>                              |
|   | Coefficient : <b>2,25</b>                       |
| Section : <b>Sciences de l'Informatique</b>   | <b>SESSION PRINCIPALE</b>                       |

Le sujet comporte 3 pages.

### Exercice 1 (2,5 points)

Soit l'algorithme de la fonction **Inconnue** suivante :

```

0) DEF FN Inconnue (ch : Chaîne) : .....
1) Si Pos ("+", ch) = 0
    Alors
        Valeur (ch, n, e)
        Inconnue ← n
    Sinon
        Valeur (Sous_chaine (ch, 1, Pos ("+", ch)-1), n, e)
        Efface (ch, 1, Pos ("+", ch))
        Inconnue ← n + FN Inconnue (ch)
    FinSi
2) Fin Inconnue

```

#### Travail demandé

- Déterminer le type de retour de la fonction **Inconnue**.
- Donner la trace d'exécution ainsi que le résultat retourné par la fonction **Inconnue** pour **ch = "5+3+14+7"**
- Déduire le rôle de la fonction **Inconnue**.

### Exercice 2 (2,75 points)

On veut simuler la croissance d'une population de virus comme suit :

- à la 1<sup>ère</sup> heure, le virus est jeune,
- à la 2<sup>ème</sup> heure, le virus jeune devient adulte,
- à la 3<sup>ème</sup> heure, le virus adulte génère un virus jeune,
- on suppose que les virus ne meurent pas.

#### Travail demandé

- Etant donné qu'au début de l'expérience, la population de virus est constituée d'un unique virus jeune, donner les valeurs des virus jeunes et des virus adultes pour les trois dernières lignes du tableau suivant :

| Heures \ Virus | Jeunes | Adultes |
|----------------|--------|---------|
| h = 1          | 1      | 0       |
| h = 2          | 0      | 1       |
| h = 3          | 1      | 1       |
| h = 4          | 1      | 2       |
| h = 5          | 2      | 3       |
| h = 6          | 3      | 5       |
| h = 7          | .....  | .....   |
| h = 8          | .....  | .....   |
| h = 9          | .....  | .....   |

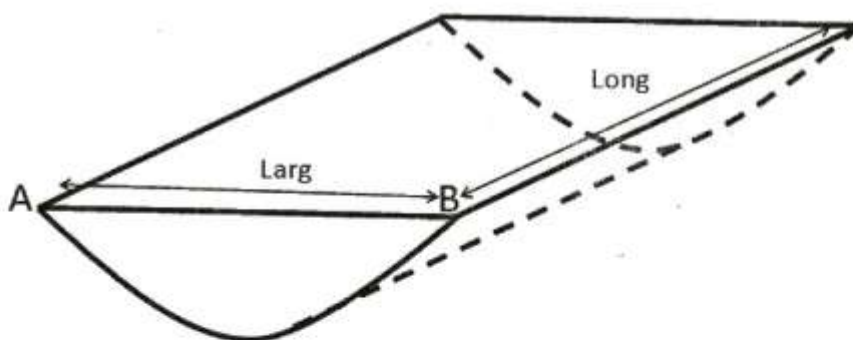
2. Sachant que le nombre total de virus par heure est calculé par une suite  $U$  définie comme suit :

$$\begin{cases} U_1 = 1 \\ U_2 = 1 \\ U_h = U_{h-1} + U_{h-2} \text{ avec } U_h \text{ le nombre total de virus après } h \text{ heures (avec } h > 2) \end{cases}$$

Ecrire un algorithme d'un module qui permet de déterminer l'heure  $h$  à partir de laquelle le nombre de virus dépasse une valeur  $k$ .

### **Exercice 3 (2,25 points)**

Dans le but de creuser une rivière de largeur **Larg** mètres, d'une rive **A** à une rive **B**, et de longueur **Long** mètres, une société de travaux publics veut déterminer le volume approximatif de sable à enlever. La forme de la partie enlevée est obtenue par la représentation graphique de la fonction  $f$  définie par  $f(x) = -3 * \sin(x)$ , comme illustré ci-dessous :



#### **Travail demandé**

Ecrire un algorithme d'un module qui permet de calculer une valeur approchée du volume de sable à enlever entre les deux rives **A** et **B** pour creuser la rivière, pour un nombre de subdivisions  $N$ .

### **Problème (12,5 points)**

Etant donné un fichier texte nommé "F\_IPV4.txt" contenant dans chaque ligne une adresse **IPV4**.

On se propose de vérifier la validité des adresses **IPV4** stockées dans ce fichier, de déterminer la classe à laquelle appartient chacune des adresses valides, de les faire migrer vers le système **IPV6** et de stocker dans un fichier d'enregistrements nommé "F\_IPV6.dat" chaque adresse **IPV4** valide ainsi que la classe à laquelle elle appartient et son équivalent en **IPV6**.

Pour ce faire, on dispose des informations suivantes :

1. Une adresse **IPV4** valide est codée sur quatre octets (32 bits) et représentée sous la forme  $W.X.Y.Z$  avec  $W, X, Y$  et  $Z$  sont quatre entiers naturels appartenant chacun à l'intervalle  $[0, 255]$  et séparés par le caractère "."

**NB.** Pour vérifier la validité d'une adresse **IPV4**, le candidat est appelé uniquement à vérifier si  $W, X, Y$  et  $Z$  sont dans l'intervalle  $[0, 255]$ .

2. Chaque adresse **IPV4** valide appartient à une classe :
  - ✓ Classe **A**, si la valeur du premier bit à gauche de la représentation en binaire de W est **0**.
  - ✓ Classe **B**, si la valeur des deux premiers bits à gauche de la représentation en binaire de W est **10**.
  - ✓ Classe **C**, si la valeur des trois premiers bits à gauche de la représentation en binaire de W est **110**.
  - ✓ Classe **D**, si la valeur des quatre premiers bits à gauche de la représentation en binaire de W est **1110**.
  - ✓ Classe **E**, si la valeur des quatre premiers bits à gauche de la représentation en binaire de W est **1111**.
3. Une adresse **IPV6** est codée sur 16 octets (128 bits). Pour faire migrer une adresse **IPV4** valide vers le système **IPV6**, on va s'intéresser uniquement au bloc de 32 bits dans l'adresse **IPV6** qui représente la conversion en hexadécimal de l'adresse **IPV4**.  
Pour ce faire, on convertit chacun des nombres W, X, Y et Z en hexadécimal, puis, les concaténer en insérant le caractère ":" au milieu du résultat obtenu.

### Exemple

L'adresse 155.105.50.68 est **valide** et elle appartient à la classe **B** car la valeur des deux premiers bits à gauche de la représentation en binaire de 155 qui est **10011011** est **10**.

L'équivalent du nombre décimal 155 en hexadécimal est 9B

L'équivalent du nombre décimal 105 en hexadécimal est 69

L'équivalent du nombre décimal 50 en hexadécimal est 32

L'équivalent du nombre décimal 69 en hexadécimal est 45

Donc, le bloc de 32 bits dans l'adresse **IPV6** qui représente la conversion en hexadécimal de l'adresse **IPV4** est 9B69:3245

### NB.

- Le candidat n'est pas appelé à remplir le fichier "**F\_IPV4.txt**".
- Tous les fichiers utilisés sont enregistrés sur la racine C.

### Travail demandé

1. Analyser le problème en le décomposant en modules.
2. Analyser chacun des modules envisagés.