

|  |   |
|--|---|
| REPUBLIQUE TUNISIENNE<br>MINISTERE DE L'EDUCATION<br>◆◆◆<br><b>EXAMEN DU BACCALAUREAT</b><br><b>SESSION DE JUIN 2014</b> | Epreuve : <b>ALGORITHMIQUE ET PROGRAMMATION</b> |
|  | Durée : 3 H                                     |
|  | Coefficient : 2,25                              |
| <b>Section : Sciences de l'informatique</b>  | <b>Session principale</b>                       |

**Exercice 1 : (2,25 points)**

Soient les algorithmes et les tableaux des déclarations suivants :

- 0) Début Jeux
- 1) g1.balle ← vrai, g2.balle ← faux, g3.balle ← faux
- 2) N ← aléatoire(10) + 1
- 3) Pour i de 1 à N Faire
  - Proc jeu1**(g1, g2)
  - Proc jeu2**(g2, g3)
  - Proc jeu1**(g3, g1)
- Fin Pour
- 4) Si (g1.balle)
  - Alors Ecrire ("Balle sous gobelet 1")
  - Sinon Si (g2.balle)
    - Alors Ecrire ("Balle sous gobelet 2")
    - Sinon Ecrire ("Balle sous gobelet 3")
  - FinSi
- 5) Fin jeux

T.D.N.T

| Type                     |
|--------------------------|
| Gobelet = enregistrement |
| balle : booléen          |
| couleur : chaîne [10]    |
| Fin                      |

- 0) DEF PROC jeu1 (Var x, y : Gobelet)
- 1) b ← x.balle
  - x.balle ← y.balle
  - y.balle ← b
- 2) Fin jeu1

T.D.O.G.

| Objet      | Type/Nature |
|------------|-------------|
| g1, g2, g3 | Gobelet     |
| n, i       | Entier      |
| jeu1, jeu2 | procedure   |

- 0) DEF PROC jeu2 (x, y : Gobelet)
- 1) b ← x.balle
  - x.balle ← y.balle
  - y.balle ← b
- 2) Fin jeu2

T.D.O.L.

| Objet | Type/Nature |
|-------|-------------|
| b     | booléen     |

T.D.O.L.

| Objet | Type/Nature |
|-------|-------------|
| b     | booléen     |

Valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** si elle est fausse.

1- Après exécution du programme **Jeux** ci-dessus pour  $N=1$ , le message affiché sera :

"Balle sous gobelet 1"       "Balle sous gobelet 2"       "Balle sous gobelet 3"

2- Soit **f** un fichier de booléens.

a- Pour remplir un champ de la variable **g3**, on peut utiliser l'instruction :

Lire (g3.balle)       Lire (g3.couleur)       Lire (f, g3.balle)

b- Pour afficher le contenu de la variable **g1**, on peut utiliser l'instruction :

Ecrire (g1)       Ecrire (g1.balle, g1.couleur)       Ecrire (f, g1.balle)

## Exercice 2 : (2,75 points)

Soit la suite  $U$  définie par :

$$\begin{cases} U_0 \text{ est un entier positif pris au hasard (avec } 3 \leq U_0 < 40), \\ U_n = U_{n-1} \text{ Div } 2, \text{ si } U_{n-1} \text{ est pair, sinon } U_n = 3 * U_{n-1} + 1 \text{ (pour } n > 0). \end{cases}$$

Cette suite aboutit au cycle redondant formé par les trois termes **4, 2 et 1** à partir d'un certain rang.

**Exemple:**

Pour  $U_0 = 3$

$$U_1=10 \quad U_2=5 \quad U_3=16 \quad U_4=8 \quad U_5=4 \quad U_6=2 \quad U_7=1 \quad U_8=4 \quad U_9=2 \quad U_{10}=1 \quad U_{11}=4 \quad U_{12}=2 \quad U_{13}=1 \quad \dots$$

Donc, la suite  $U$  entre dans le cycle redondant **4, 2 et 1** à partir du 6<sup>ème</sup> terme (rang = 6).

**Travail demandé :**

Ecrire un algorithme d'un module récursif, permettant de déterminer le rang à partir duquel la suite  $U$  aboutit au cycle redondant **4, 2 et 1**.

## Exercice 3 : (4 points)

On se propose de calculer le **Plus Grand Commun Diviseur (PGCD)** de  $N$  entiers positifs, déjà stockés dans la 1<sup>ère</sup> ligne d'une matrice carrée  $M$ . Pour ce faire :

- remplir les cases des  $(N-1)$  autres lignes, de sorte que la valeur d'une case  $M[L,C]$  est égale au PGCD des contenus de  $M[L-1,C]$  et  $M[L-1,C+1]$ .
- la case  $M[N, 1]$  contiendra le PGCD des  $N$  entiers.

**Exemple :** pour  $N = 4$  :

|   |    |    |    |    |  |
|---|----|----|----|----|--|
| 1 | 60 | 48 | 16 | 34 | ← La 1 <sup>ère</sup> ligne déjà remplie |
| 2 | 12 | 16 | 2  |    | ← PGCD (16, 34)                          |
| 3 | 4  | 2  |    |    |  |
| 4 | 2  |    |    |    | ← PGCD (60, 48, 16, 34)                  |

**Travail demandé :**

Ecrire un algorithme d'un module permettant de calculer le PGCD de  $N$  entiers positifs en utilisant le procédé décrit ci-dessus.

## Problème : (11 points)

Une molécule est un regroupement d'au moins deux atomes qui sont unis par des liens chimiques et elle est représentée par une formule chimique. Exemple :  $H_2O$ .

Une formule chimique est une succession de symboles d'atomes, suivi chacun par un entier représentant le nombre d'apparitions (**nbr**) de l'atome dans la molécule.

Chaque atome est symbolisé par la première lettre de son nom en majuscule, suivie éventuellement d'une deuxième lettre en minuscule pour distinguer des atomes ayant des initiales identiques. Ainsi, le **Fluor (F)** se distingue du **Fer (Fe)**, du **Fermium (Fm)** et du **Francium (Fr)**,

Le calcul de la masse molaire moléculaire d'une molécule, notée **M(Molécule)**, sera comme suit :

- pour chaque atome de la molécule, calculer le produit (**nbr x A(atome)**) où **A(atome)** est un réel représentant la masse atomique de l'atome,
- calculer la somme des produits obtenus.

### *Exemple :*

*Pour la molécule dichromate de potassium ( $K_2Cr_2O_7$ ) qui est constituée de 2 atomes de potassium (K), 2 atomes de chrome (Cr) et 7 atomes d'oxygène (O), sa masse molaire moléculaire  $M(K_2Cr_2O_7)$  est égale à  $2 \cdot A(K) + 2 \cdot A(Cr) + 7 \cdot A(O)$ .*

*Puisque  $A(K) = 39,1 \text{ g/mol}$ ,  $A(Cr) = 52 \text{ g/mol}$  et  $A(O) = 16 \text{ g/mol}$ ,*

*alors  $M(K_2Cr_2O_7)$  sera égale à  $2 \cdot 39,1 + 2 \cdot 52 + 7 \cdot 16 = 294,2 \text{ g/mol}$*

En disposant d'un fichier texte "**Molecules.txt**" dont chaque ligne contient le **nom** d'une molécule suivi de sa **formule chimique**, séparés par le caractère astérisque "\*", écrire un programme permettant de :

- remplir un fichier "**Atomes.dat**" par les données relatives à N atomes ( $N \leq 50$ ), où chacun est représenté par son **symbole** et sa **masse atomique**,
- stocker dans un fichier "**Resultats.dat**" le **nom** et la **masse molaire moléculaire** de chaque molécule figurant dans le fichier "**Molecules.txt**".

### **Travail demandé :**

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