

EXAMEN DU BACCALAUREAT
SESSION DE JUIN 2011

SESSION
PRINCIPALE

SECTION : Sciences de l'Informatique

EPREUVE THEORIQUE : Algorithmique et programmation DUREE : 3h COEFFICIENT : 2.25

Le sujet comporte 4 pages numérotées de 1/4 à 4/4

Première partie : (10 points)

Exercice n°1 : (4 points)

Soit l'algorithme suivant de la fonction **inconnue** :

Fonction Inconnue (b, n : entier) :

Début

ch ← ""

Répéter

r ← n Mod b

n ← n Div b

Selon r **Faire**

0..9 : Convch (r, s)

10 : s ← "A"

11 : s ← "B"

12 : s ← "C"

13 : s ← "D"

14 : s ← "E"

15 : s ← "F"

FinSelon

ch ← s + ch

Jusqu'à (n = 0)

Inconnue ← ch

Fin Inconnue

1. Donner le résultat de cette fonction pour chacun des couples (b, n) suivants : (2,13) et (16,163).
2. Donner le type de la fonction **Inconnue** ainsi que le tableau de déclaration de ses objets.
3. En déduire le rôle de cette fonction.

Exercice n°2 : (3 points)

Soit la suite U définie par :

$$\begin{cases} U_0 = 0, U_1 = -9 \\ U_n = 6*U_{n-1} - 9*U_{n-2} \text{ pour tout entier naturel } n \geq 2. \end{cases}$$

1. Ecrire l'algorithme d'une fonction récursive qui permet de retourner le n^{ème} terme de la suite U.
2. Donner l'ordre de récurrence de la fonction proposée. Justifier la réponse.

Exercice n°3 : (3 points)

Soit N un entier naturel de grande taille avec un nombre de chiffres le composant compris entre **20** et **200**. N est divisible par **8**, si *l'un des cas suivants* est vérifié :

1^{er} cas : Le chiffre des centaines est **pair** et le nombre formé par les 2 derniers chiffres (les plus à droite) est multiple de **8**.

2^{ème} cas : Le chiffre des centaines est **impair** et le nombre formé par les 2 derniers chiffres (les plus à droite) diminué de 4 est multiple de **8**.

Exemple 1 : Si $N = 1245896578541236593224$, alors :

- son chiffre des **centaines** est **pair** puisqu'il est égal à **2**
- le nombre formé par les **2 derniers chiffres** (les plus à droite), est **multiple de 8** puisqu'il est égal à **24** qui est multiple de **8**.
- donc, le premier cas est vérifié, et N est multiple de **8**.

Exemple 2 : Si $N = 1245896578541236593120$, alors :

- son chiffre des **centaines** est **impair** puisqu'il est égal à **1**
- le nombre formé par les **2 derniers chiffres** (les plus à droite) **diminué de 4**, est **multiple de 8** puisqu'il est égal à **20** et que **16** ($20 - 4$) est multiple de **8**.
- donc, le deuxième cas est vérifié, et N est multiple de **8**.

Exemple 3 : Si $N = 1245896578541236593221$, alors :

- son chiffre des **centaines** est **pair** puisqu'il est égal à **2**
- le nombre formé par les **2 derniers chiffres** (les plus à droite), **n'est pas multiple de 8** puisqu'il est égal à **21**.
- donc, N n'est pas multiple de **8**.

Travail demandé :

Ecrire l'**analyse** d'un module intitulé **Div_huit**, permettant de vérifier la divisibilité d'un entier N par **8** selon le principe décrit précédemment.

NB : Le candidat n'est pas appelé à développer le module de saisie de N

Deuxième partie : (10 points)

Soit **Espace** une matrice carrée d'ordre **N** remplie par des **0** et des **1**. On se propose de chercher les zones de concentration du chiffre **1** qui peuvent exister dans la matrice **Espace**. Pour déterminer les zones de concentration, on suit les étapes suivantes :

1. On remplit d'une manière aléatoire (au hasard) la matrice carrée **Espace**, par des **0** et des **1**.
2. On fixe **DN**, un diviseur de **N**, afin de partager la matrice **Espace** en carrés de dimensions égales à **DN*DN**, comme l'illustre l'exemple suivant :

Si **N = 9**, et **DN = 3** alors le nombre de carrés sera égal à $\left(\frac{N}{DN}\right)^2 = 9$ et la subdivision de la matrice

Espace en **9** carrés se fera de la manière suivante :

	1	2	3	4	5	6	7	8	9
1	1	0	1	1	0	1	0	0	1
2	0	1	0	0	0	1	1	0	0
3	1	0	0	1	1	0	0	0	1
4	1	0	1	1	1	1	0	1	0
5	1	0	0	1	1	1	1	1	0
6	0	0	1	0	1	1	0	0	1
7	1	0	1	0	1	0	1	1	0
8	0	1	1	1	0	1	1	0	0
9	0	1	1	0	1	0	0	1	1

3. On saisit le degré de concentration minimum **Deg_Min**, pour lequel on va chercher les zones de concentration du chiffre **1**. (**Deg_Min** \in [1,(**DN*DN**)])
4. On détermine toutes les zones de concentration du chiffre **1** dans la matrice **Espace**. Les zones de concentration correspondent aux carrés de la matrice **Espace** qui contiennent un nombre d'occurrences du chiffre **1** supérieur ou égal à **Deg_Min**.

On se propose d'écrire un programme qui permet de :

- Saisir **N**, la taille de la matrice **Espace** et la remplir de **0** et de **1**, d'une manière aléatoire.
- Saisir **DN**, un diviseur de **N**.
- Saisir **Deg_Min**, le degré de concentration minimum (**Deg_Min** \in [1,(**DN*DN**)]).
- Déterminer et afficher le nombre de zones de concentration du chiffre **1**.
- Afficher les caractéristiques de chaque zone de concentration en précisant à **chaque fois** les deux informations suivantes :
 1. Les coordonnées (**ligne, colonne**) de la **1^{ère}** case de la zone de concentration (**en haut à gauche**).
 2. Le nombre d'occurrences du chiffre **1** figurant dans cette zone.

Pour la matrice **Espace** de l'exemple précédent et pour **Deg_Min** égal à 5, on obtient les résultats suivants :

Le nombre de zones de concentration du chiffre 1, est égal à 4.

Les zones de concentration se définissent ainsi :

Zone n°1 : ligne : 1, colonne : 4. Le nombre de 1 dans cette zone est : 5.

Zone n°2 : ligne : 4, colonne : 4. Le nombre de 1 dans cette zone est : 8.

Zone n°3 : ligne : 7, colonne : 1. Le nombre de 1 dans cette zone est : 6.

Zone n°4 : ligne : 7, colonne : 7. Le nombre de 1 dans cette zone est : 5.

En effet, les zones de concentration étant les subdivisions de la matrice **Espace** où le nombre d'occurrences du chiffre 1 est supérieur ou égal à **Deg_Min**, on peut illustrer ces résultats de la manière suivante :

	1	2	3	4	5	6	7	8	9
1	1	0	1	1	0	1	0	0	1
2	0	1	0	0	0	1	1	0	0
3	1	0	0	1	1	0	0	0	1
4	1	0	1	1	1	1	0	1	0
5	1	0	0	1	1	1	1	1	0
6	0	0	1	0	1	1	0	0	1
7	1	0	1	0	1	0	1	1	0
8	0	1	1	1	0	1	1	0	0
9	0	1	1	0	1	0	0	1	1

Travail demandé :

1. Analyser le problème en le décomposant en modules et en déduire l'algorithme du programme principal.
2. Analyser chacun des modules envisagés précédemment et en déduire les algorithmes correspondants.