ÉPREUVE THÉORIQUE

Algorithmique & programmation

Section : Sciences de l'informatique

Etablissements:

Lycée Cité **ENNOUR** — Lycée **TATAOUINE**
Lycée Cité **EL MAHRAJEN** — Lycée **BROURMET**

COEFF. : 3

Date : 09 –05- 2013

Durée : 3 heures

Partie I :

Exercice 1 : (3 points)

L'une des règles de divisibilité par 19 consiste à ajouter à N augmenté de son chiffre des unités le double du chiffre supprimé et recommencer éventuellement avec le nombre ainsi obtenu jusqu'au moment où l'on peut conclure à la divisibilité (***jusqu'à obtenu une valeur <= 19***).

Exemple:

N=**345686**

```
34568+2*6 = 34580
3458 +2*0 = 3458
345  +2*8 = 361
36   +2*1 = 38
3    +2*8 = 19      qui est divisible par 19
```

Proposer l'algorithme d'une fonction qui vérifie si un nombre N est divisible par 19 ou non.

Exercice 2 : (3 points)

Soit la fonction suivante :

```
0) Def  fn  inconnue (n :entier) : ..................
1) Si (n=0) alors
                inconnue ← ""
     Sinon
          Si (n mod 2 = 0) Alors
              inconnue ← FN inconnue (n div 2) + "0"
          Sinon
              inconnue ← FN inconnue (n div 2) + "1"
          Fin Si
   Fin Si
2) Fin inconnue
```

1. Quel est le type du résultat de cette fonction ?
2. Donnez la trace d'exécution de la fonction « inconnue » pour n = 12
3. Quel est le rôle de cette fonction ?

Exercice 3 : (4 points)

On vous rappel que un **nombre premier** est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs (qui sont alors 1 et lui-même).

Un **quadruplet de premiers** est un couple constitué de **4 nombres jumeaux premiers successifs,** séparés d'une distance de **4** de la forme suivante : **(p , p + 2 , p + 6 , p + 8)**

Exemple :

Les quadruplets de premiers jumeaux sont :

(5, 7, 11, 13), (11, 13, 17, 19), (101, 103, 107, 109)……

En 1919 Vigo Brun (1885–1978) à prouvé que la somme de la série des inverses des nombres premiers jumeaux converge vers une constante (B) dite constante de Brun :

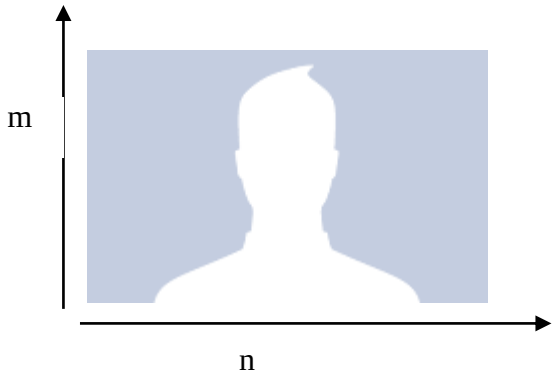
$$B = \left(\frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13}\right) + \left(\frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \frac{1}{19}\right) + \left(\frac{1}{101} + \frac{1}{103} + \frac{1}{107} + \frac{1}{109}\right) + \cdots$$

Travail demandé :

1. Donner l'algorithme de la fonction premier.
2. Donner l'algorithme de la fonction **Calcul_B** qui permet de calculer et retourner une valeur approché de la constante B. Le calcul s'arrête quand la différence entre deux sommes successives devient inférieure ou égale à une erreur « Epsilon » (0<Epsilon<0.0001) .

Partie II

Problème : (10 points)



On vous rappel qu'une image bitmap est représentée par **n*m pixels** (n= le nombre de pixels horizontalement et m est le nombre de pixel verticalement).

Un pixel d'une image est identifié par ses coordonnées dans l'image et sa couleur.

On suppose que le nombre de couleurs est égal à 16 au maximum. Les couleurs des pixels sont codées en hexadécimal par un caractère de "0" à "F" et seront stockées dans une matrice de n*m caractères.

La couleur **la plus fréquente** sera considérée la couleur **de fond de l'image.**

Pour sauvegarder l'image dans un fichier sous un format compressé, on va sauvegarder, une seule fois, la couleur de fond ainsi que les dimensions de l'image en pixels dans le premier enregistrement du fichier. Pour le reste des pixels ils seront sauvegardés par bloc des pixels consécutifs ayants une même couleur et différente de la couleur de fond de l'image.

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

- ✓ Saisir les informations d'une image.
- ✓ Compresser et sauvegarder l'image.

A. **Saisir les informations d'une image** :

Le programme fait la saisie des couleurs des pixels de l'image dans une matrice (MP) de n* m caractères (n est m sont deux entiers positifs inférieurs 400). Le code couleur d'un pixel est exprimé en hexadécimal par un caractère de ''0'' à ''F'').

Exemple : Pour n=8 et m=9 et MP =

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	1	1
3	1	2	1	1	1	1	2	1
4	1	A	2	1	1	2	E	1
5	1	E	E	2	2	E	E	1
6	1	E	2	2	2	2	E	1
7	1	2	1	1	1	1	2	1
8	2	2	2	2	2	2	2	2
9	1	1	1	1	1	1	1	1

MP [2,4] =''A'' représente le pixel de coordonnées 2, 4 et a la couleur 'A' qui vaut 10 en décimal.

MP [5,8] =''2'' représente le pixel de coordonnées (5,8) et a la couleur 1.

B. **Compresser et sauvegarder l'image**:

Le programme sauvegarde le contenu de la matrice dans un fichier selon le principe de compression suivant :

- Seulement les pixels ayant une couleur **différente** de la couleur de fond seront sauvegardés.
- La couleur de fond est la couleur la plus utilisée dans l'image.
- Pour chaque ligne de la matrice on n'enregistre que des blocs des pixels consécutifs qui ont la même couleur. Un bloc de pixels consécutif est défini par le numéro de la ligne occupé (L), le numéro de la colonne du début du bloc (Cd), le numéro de la colonne de la fin du bloc (Cf) et la couleur du bloc (Cb).
- Le premier enregistrement du fichier contiendra dans le champ **Cb** la couleur du fond, dans le champ **L** la largeur de l'image (n) en pixels, dans le champ **Cd** la hauteur de l'image (m) et le champ **Cf** contiendra la valeur 0.

Exemple :
Pour la matrice précédente La couleur de fond sera 1 puisque la couleur 1 est la plus utilisée et le fichier correspondant sera représenté comme suit :

L	Cd	Cf	Cb
8	9	0	1
2	3	6	2
3	2	2	2
3	7	7	2
4	2	2	10
4	3	3	2
4	7	7	14
5	2	3	14
5	6	7	14
6	2	2	14
6	3	6	2
6	7	7	14
7	2	2	2
7	7	7	2
8	1	8	2

Remarque :

- La couleur d'un bloc est enregistrée dans le fichier en décimal.
- Le fichier sera enregistré dans le dossier « c:\images » sous un nom saisie au moment de l'enregistrement par l'utilisateur. Un nom de fichier peut être composé de tout caractère sauf les caractères suivants : / , \, <, >, : , * , ? , " , |

Travail demandé :

- Analyser le problème en le décomposant en modules.
- Analyser chacun des modules envisagés précédemment.