

**Le sujet comporte 04 pages**

### Exercice 1 : (3 points)

Soit le tableau de déclaration des nouveaux types suivant :

Type
<b>Personne</b> = Enregistrement <div style="margin-left: 100px;">id, age : entier</div> <div style="margin-left: 100px;">genre : caractère</div> <div style="margin-left: 40px;">Fin Personne</div>
<b>Tab1</b> = tableau de 100 <b>Personne</b>
<b>Tab2</b> = tableau de 200 <b>Personne</b>

Soient **H** et **F** deux tableaux contenant respectivement **n1** et **n2** enregistrements de type **Personne** et triés selon l'ordre croissant du champ **id**.

Et soit l'algorithme de la procédure **Traitement** suivant :

0) Def Proc Traitement (n1,n2 : Octet ; H, F : Tab1 ; Var P : Tab2)

1)  $k \leftarrow 0, i \leftarrow 0, j \leftarrow 0$

### Répéter

$$k \leftarrow k+1$$
$$i \leftarrow i+1$$
$$P[k] \leftarrow H[i]$$
$$k \leftarrow k+1$$
$$j \leftarrow j+1$$
$$P[k] \leftarrow F[i]$$

Jusqu'à ( $i=n1$ ) ou ( $j=n2$ )

Si  $(i=n1)$  Alors

Pour  $c$  de  $j+1$  à  $n2$  Faire

$$k \leftarrow k+1$$
$$P[k] \leftarrow F[c]$$

*FinPour*

**Sinon**

Pour  $c$  de  $i+1$  à  $n1$  Faire

$$k \leftarrow k+1$$
$$P[k] \leftarrow H[c]$$

*FinPour*

*FinSi*

## 2) Fin Traitement



### Travail demandé :

- 1) Donner le contenu du tableau **P** après exécution de la procédure **Traitement** pour **n1=4**, **n2=6** et les valeurs de **H** et **F** suivantes :

<b>H</b>	id = 123	id = 125	id = 363	id = 430
	age = 57	age = 22	age = 35	age = 33
	genre = "M"	genre = "M"	genre = "M"	genre = "M"

<b>F</b>	id = 113	id = 115	id = 263	id = 380	id = 455	id = 663
	age = 57	age = 30	age = 18	age = 55	age = 23	age = 19
	genre = "F"	genre = "F"	genre = "F"	genre = "F"	genre = "F"	genre = "F"

- 2) Dédurre le rôle de la procédure **Traitement**.
- 3) Apporter les modifications nécessaires au contenu de la boucle **Répéter** pour obtenir le tableau **P** trié selon le champ **id**.

### Exercice 2 : (3 points)

Soient **a** et **n** deux entiers naturels non nuls et **F** une fonction définie de la façon suivante :

$$F(a,0) = 1$$

$$F(a,n) = F(a*a, n \text{ div } 2) \quad \text{Si } n \text{ est pair}$$

$$F(a,n) = a * F(a*a, (n-1) \text{ div } 2) \quad \text{Si } n \text{ est impair}$$

- 1) Donner la trace d'exécution de la fonction **F** pour chacun des cas suivants :

1<sup>er</sup> cas : **a = 2** et **n = 2**

2<sup>ème</sup> cas : **a = 2** et **n = 3**

- 2) En déduire le rôle de la fonction **F**.
- 3) Ecrire un **algorithme récursif** de la fonction **F**.

### Exercice 3 : (4 points)

En mathématiques, la constante de Brun (**B**) des nombres premiers jumeaux est la somme de la série des inverses des nombres premiers distants de 2.

$$B = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \left(\frac{1}{17} + \frac{1}{19}\right) + \left(\frac{1}{29} + \frac{1}{31}\right) + \dots$$

On rappelle qu'un nombre est dit premier s'il est divisible uniquement par 1 et par lui-même. Par convention l'entier 1 n'est pas premier.

### Travail demandé :

Ecrire un algorithme d'une fonction **Brun (epsilon)** permettant de calculer, à *epsilon* près, une valeur approchée de la constante de Brun définie précédemment (avec *epsilon* un réel passé en paramètres et dont la valeur est déjà saisie dans le module appelant).

**NB :** Chaque algorithme proposé doit être accompagné d'un tableau de déclaration des objets ayant la forme suivante :

Objet	Type / Nature	Rôle



Problème : (10 points)

La stéganographie est une méthode qui consiste à cacher un texte dans une image numérique bitmap. Cette méthode peut aider à échanger des messages secrets.

Dans une image RVB, chaque pixel est représenté par une chaîne de 6 chiffres hexadécimaux ; les deux premiers représentent l'intensité de la couleur "Rouge", les deux suivants celle de la couleur "Vert" et les deux derniers représentent l'intensité de la couleur "Bleu".

Exemples :

Couleur	Rouge	Vert	Noir	Blanc	Orange	Bleu	Rose
Code RVB	"FF0000"	"00FF00"	"000000"	"FFFFFF"	"ED7F10"	"0000FF"	"FD6C9E"

On se propose d'utiliser cette technique pour crypter un texte. Pour cela on dispose d'un fichier texte à crypter nommé "source.txt" situé sur la racine du disque C et comportant N lignes non vides ( $1 \leq N \leq 40$ ) de longueur maximale 120 caractères chacune.

Le procédé de cryptage est décrit ci-dessous :

- générer une matrice carrée M (40x40) à partir du fichier "source.txt" comme suit :
  - initialiser les cases de la matrice par le code de la couleur blanche "FFFFFF"
  - remplir chaque ligne de la matrice M par une ligne du fichier "source.txt" de la manière suivante :
    - ajouter à la fin de la ligne, si c'est nécessaire, un ou deux espaces pour que sa longueur soit divisible par trois
    - subdiviser la ligne en blocs de trois caractères consécutifs et remplir chaque case de la matrice par la chaîne résultante de la concaténation des équivalents hexadécimaux du code ASCII des trois caractères de chaque bloc
- générer un fichier "code.txt" à partir de la matrice M, où chaque ligne du fichier correspond à la concaténation du contenu d'une colonne de la matrice.

Exemple :

Pour le fichier "source.txt" suivant :

BAC SI 2016
44 %
BAC SC 2016
60 %

On obtient la matrice M suivante :

	1	2	3	4		39	40
1	424143	205349	203230	313620	FFFFFF	FFFFFF	FFFFFF
2	343420	252020	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF
3	424143	205343	203230	313620	FFFFFF	FFFFFF	FFFFFF
4	363020	252020	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF
	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF
	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF
39	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF
40	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF	FFFFFF



En effet :

- la matrice **M** a été initialisée par le code de la couleur blanche "**FFFFFF**"
- la première ligne de la matrice **M** est remplie à partir de la première ligne du fichier "**source.txt**" comme suit :
  - étant donné que la longueur de la première ligne du fichier "**source.txt**" est non divisible par 3, on ajoute un espace à la fin pour obtenir 4 groupements de 3 caractères consécutifs :

B A C    S I    2 0 1 6  
⏟   ⏟   ⏟   ⏟

- le premier élément de la matrice **M[1,1]** est égal à "**424143**" car :
  - le code **ASCII** de "**B**" est **66** et son équivalent hexadécimal est **42**
  - le code **ASCII** de "**A**" est **65** et son équivalent hexadécimal est **41**
  - le code **ASCII** de "**C**" est **67** et son équivalent hexadécimal est **43**
  - la concaténation des 3 équivalents hexadécimaux donne "**424143**" d'où le contenu de **M[1,1] = "424143"**
- le même procédé donne **M[1,2] = "205349"**, **M[1,3] = "203230"** et **M[1,4] = "313620"**
- les autres lignes de la matrice **M** sont remplies selon le même procédé.

D'où, en concaténant les valeurs de chaque colonne de la matrice **M** pour former une ligne du fichier, on obtient le fichier "**code.txt**" ci-dessous :

424143343420424143363020FFFFFF.....FFFFFFFFFFFFFF
205349252020205343252020FFFFFF.....FFFFFFFFFFFFFF
203230FFFFFFFF203230FFFFFFFFFFFFFF.....FFFFFFFFFFFFFF
313620FFFFFFFF313620FFFFFFFFFFFFFF.....FFFFFFFFFFFFFF
.....
FFFFFFFFFFFFFF.....FFFFFFFFFFFFFF

#### Travail demandé :

- 1- Analyser le problème en le décomposant en modules.
- 2- Ecrire un algorithme solution pour chaque module envisagé. Chaque algorithme proposé doit être accompagné d'un tableau de déclaration des objets ayant la forme suivante :

Objet	Type / Nature	Rôle