

<b>RÉPUBLIQUE TUNISIENNE</b>  <b>MINISTÈRE DE L'ÉDUCATION</b>	<b>EXAMEN DU BACCALAURÉAT - SESSION 2022</b>	
	<b>Session principale</b>	<b>NOUVEAU RÉGIME</b>
	Épreuve : <b>Algorithmique et Programmation</b>	Section : <b>Sciences de l'informatique</b>
	Durée : <b>3h</b>	Coefficient de l'épreuve : <b>2</b>



N° d'inscription

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

### Exercice 1 (2 points)

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

**Fonction Inconnu** (.....) : .....

**DEBUT**

**Si** Long(ch) = 0 **Alors**

**Retourner** 0

**Sinon**

**Si** ch[Long(ch)-1] ∈ ["0".."9"] **Alors**

d ← Valeur (ch[Long(ch)-1])

**Retourner** d + Inconnu (Sous\_chaine(ch, 0, Long(ch)-1))

**Sinon**

**Retourner** Inconnu (Sous\_chaine(ch, 0, Long(ch)-1))

**Finsi**

**Finsi**

**FIN**

### Travail demandé :

- 1- Réécrire sur votre copie d'examen l'entête de la fonction **Inconnu** en complétant la déclaration des paramètres et le type de retour.
- 2- Dresser le tableau de déclaration des objets locaux de la fonction **Inconnu**.
- 3- Quel est le résultat retourné par la fonction **Inconnu** pour **ch = "Bac22G3"**.
- 4- Déduire le rôle de la fonction **Inconnu**.

### Exercice 2 (3,5 points)

Une séquence contigüe dans une matrice carrée **M** de  $N \times N$  entiers, est une séquence formée d'au moins deux éléments successifs se trouvant sur la même ligne et dont leur somme est égale à zéro.

A partir d'une matrice **M** remplie aléatoirement, on se propose de remplir un fichier texte **F** par les positions des séquences contigües, se trouvant dans les lignes de cette matrice, comme suit :

- Dans la première ligne du fichier **F**, écrire le titre "**Les séquences contigües des lignes**".
- Pour chaque séquence contigüe trouvée, écrire ses positions dans une ligne du fichier **F** en les séparant par des espaces sachant que les positions d'une séquence sont :
  - Le numéro de la ligne de la matrice où se trouve la séquence.
  - Le numéro de la colonne de début de la séquence.
  - Le numéro de la colonne de fin de la séquence.



Exemple : Pour  $N = 5$  et la matrice  $M$  suivante :

	0	1	2	3	4
0	20	2	-5	3	2
1	10	3	0	60	2
2	1	2	-3	2	-2
3	30	-5	40	50	2
4	-7	4	2	1	-9

Le contenu du fichier  $F$  sera :

**Les séquences contigües des lignes**

0 1 3

0 2 4

2 0 2

2 0 4

2 3 4

4 0 3

En effet, pour la ligne 0 de la matrice  $M$  on a :  $M[0,1]+M[0,2]+M[0,3]=0$  donc cette séquence est contigüe et se trouve dans la ligne numéro 0, le numéro de sa colonne de début est 1 et le numéro de sa colonne de fin est 3. Dans le fichier  $F$  on écrit la ligne "0 1 3".

#### Travail demandé :

Ecrire un algorithme d'une procédure **Remplir\_F(M, N, F)** permettant de remplir un fichier  $F$  par les séquences contigües se trouvant dans les lignes d'une matrice  $M$  de  $N \times N$  entiers comme décrit précédemment.

**N.B. :**

- Une ligne de la matrice  $M$  peut contenir plusieurs séquences contigües.
- Le candidat n'est pas appelé à :
  - saisir  $N$  et  $M$ .
  - écrire l'instruction d'ouverture du fichier  $F$ .

#### Exercice 3 (3 points)

Soient  $x$  un réel positif et  $U$  une suite définie par :

$$\begin{cases} U_0 = \frac{(1+x)}{2} \\ U_n = \frac{1}{2} \left( U_{n-1} + \frac{x}{U_{n-1}} \right) \quad \text{pour tout } n > 0 \end{cases}$$

Le terme  $U_n$  est une valeur approchée de la racine carrée de  $x$  à **epsilon** près, si  $\left| \frac{U_n - U_{n-1}}{U_{n-1}} \right| < \text{epsilon}$

#### Travail demandé :

- 1- Quel est l'ordre de récurrence de la suite  $U$  ? Justifiez votre réponse.
- 2- Ecrire un algorithme d'une fonction **RacineU(x)** qui retourne une valeur approchée de la racine carrée d'un réel positif  $x$  à **epsilon** près (avec  $\text{epsilon} = 10^{-4}$ ) en utilisant la suite  $U$  définie précédemment.

**N.B. :** Le candidat n'est pas appelé à saisir  $x$ .



#### 📎 Exercice 4 (4,5 points)

Soit  $T$  un tableau de  $N$  entiers trié dans l'ordre croissant. Pour rechercher un entier  $x$  dans le tableau  $T$ , on peut utiliser une méthode de recherche dite "Trichotomique" dont le principe est le suivant :

**Etape 1:** On compare  $x$  avec  $T[p_1]$  et  $T[p_2]$ , sachant que  $p_1 = (2*d+f) \text{ div } 3$  et  $p_2 = (d+2*f) \text{ div } 3$  (où  $d$  et  $f$  sont respectivement l'indice du début et l'indice de la fin du tableau) :

- Si  $x$  est égal à  $T[p_1]$  ou égal à  $T[p_2]$ , la recherche est terminée.
- Si  $x$  est inférieur à  $T[p_1]$ , on refait la recherche dans la partie gauche du tableau (de l'indice  $d$  à l'indice  $p_1-1$ ).  
Sinon Si  $x$  est inférieur à  $T[p_2]$ , on refait la recherche dans la partie du milieu du tableau (de l'indice  $p_1+1$  à l'indice  $p_2-1$ ).  
Sinon ( $x$  est supérieur à  $T[p_2]$ ), on refait la recherche dans la partie droite du tableau (de l'indice  $p_2+1$  à l'indice  $f$ ).

**Etape 2:** On refait **Etape 1** pour la partie sélectionnée du tableau jusqu'à trouver l'élément recherché ( $x$  est égal à  $T[p_1]$  ou égal à  $T[p_2]$ ) ou l'élément recherché n'existe pas ( $d > f$ ).

**Exemple :** Pour  $x = 10$ ,  $N = 8$  et le tableau  $T$  suivant :

0	1	2	3	4	5	6	7
-5	-2	0	4	6	10	15	23

- On a  $d = 0$  et  $f = 7$  donc :

$$p_1 = (2*0+7) \text{ div } 3 = 7 \text{ div } 3 = 2 \text{ et } p_2 = (0+2*7) \text{ div } 3 = 14 \text{ div } 3 = 4$$

$$T[p_1] = T[2] = 0 \neq x$$

$$T[p_2] = T[4] = 6 \neq x$$

$x$  est supérieur à  $T[p_2]$ , donc on refait la recherche dans la partie droite du tableau de l'indice  $p_2+1$  ( $4+1=5$ ) à l'indice 7

- Pour la partie sélectionnée du tableau  $d = 5$  et  $f = 7$  donc :

$$p_1 = (2*5+7) \text{ div } 3 = 17 \text{ div } 3 = 5 \text{ et } p_2 = (5+2*7) \text{ div } 3 = 19 \text{ div } 3 = 6$$

$T[p_1] = T[5] = 10 = x$ , donc la recherche est terminée et  $x$  se trouve dans le tableau.

#### Travail demandé :

Ecrire un algorithme d'une fonction récursive **Rech\_Trich( $T, d, f, x$ )** qui permet de vérifier l'existence d'un entier  $x$  dans un tableau  $T$  d'entiers trié dans l'ordre croissant en utilisant la méthode de recherche Trichotomique décrite précédemment. Les indices des cases du tableau  $T$  commencent de  $d$  jusqu'à  $f$  (avec  $d < f$ ).

**N.B. :** Le candidat n'est pas appelé à saisir  $T, d, f$  et  $x$ .

#### 📎 Exercice 5 (7 points)

Un nombre décimal  $n$  est dit brésilien s'il possède, dans une base  $B$  (avec  $2 \leq B \leq n-2$ ), une représentation qui s'écrit sous la forme de  $p$  chiffres égaux, c'est-à-dire :  $n = \underbrace{(kkk\dots kkk)}_p B$

$p$  chiffres

#### Exemples :

- 7 est un nombre brésilien car  $7 = (111)_2$
- 3124 est un nombre brésilien car  $3124 = (44444)_5$
- 1170 est un nombre brésilien car  $1170 = (2222)_8$
- 20 est un nombre brésilien car  $20 = (22)_9$
- 204 est un nombre brésilien car  $204 = (CC)_{16}$
- 9 n'est pas un nombre brésilien car  $9 = (1001)_2 = (100)_3 = (21)_4 = (14)_5 = (13)_6 = (12)_7$  et aucune de ces écritures n'est brésilienne.



On se propose d'écrire un algorithme d'une procédure **Gen\_Bres** qui permet de créer et de remplir un fichier d'enregistrements nommé "**F\_Brésilien.dat**" par les nombres brésiliens contenus dans un fichier texte existant nommé "**Nombres.txt**", sachant que chaque ligne du fichier texte "**Nombres.txt**" contient un nombre décimal et chaque enregistrement du fichier "**F\_Brésilien.dat**" comportera les champs suivants :

- **N** : Le nombre décimal.
- **B** : Une base dans laquelle le nombre  $N$  s'écrit sous la forme de  $p$  chiffres égaux avec  $2 \leq B \leq 16$ .
- **Rep** : La représentation du nombre décimal  $N$  dans la base  $B$ .

**Travail demandé :**

- 1- Donner une déclaration d'un type pour le fichier d'enregistrement "**F\_Brésilien.dat**" ainsi que celles des types nécessaires à sa déclaration.
- 2- Donner en algorithmique les instructions d'ouverture des deux fichiers "**Nombres.txt**" et "**F\_Brésilien.dat**", sachant que le fichier à créer et à remplir "**F\_Brésilien.dat**" et le fichier source "**Nombres.txt**" se trouvent sur la racine du disque **D**.
- 3- Ecrire un algorithme de la procédure **Gen\_Bres**, sachant que le fichier "**Nombres.txt**" est déjà rempli dans le programme appelant.