RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT - SESSION 2022	
	Session de contrôle	**ANCIEN RÉGIME**
	Épreuve : Algorithmique et Programmation	Section : Sciences de l'informatique
	Durée : **3h**	Coefficient de l'épreuve : **2.25**

N° d'inscription | | | | | | |

Le sujet comporte 5 pages numérotées de 1/5 à 5/5

Important :

Chaque solution développée par le candidat sous forme d'une analyse ou d'un algorithme doit être accompagnée d'un tableau de déclaration des objets ayant la forme suivante :

Objet	Type/Nature	Rôle

✎ Exercice 1 (3 points)

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

0) DEF FN Inconnue (Var F : fiche) : entier
1) Si Non (Fin_fichier(F)) Alors
Lire (F, e)
Inconnue ← 1 + FN Inconnue (F)
Sinon
Inconnue ← 0
Fin Si
2) Fin Inconnue

NB : **F** est un fichier d'entiers.

Travail demandé :

a) Donner la valeur retournée par la fonction **Inconnue** pour le contenu du fichier **F** donné ci-dessous.

F
23
345
657
22
789
55

b) Déduire le rôle de cette fonction.

c) Recopier le tableau ci-dessous sur votre copie, puis en se référant à l'algorithme de la fonction **Inconnue** et pour chacune des propositions ci-après, remplir la case correspondante par la lettre de la réponse correcte.

Proposition	**Proposition 1**	**Proposition 2**	**Proposition 3**
Réponse			

Proposition 1 :

Fin_fichier (F) représente
- **a)** l'appel récursif.
- **b)** la condition d'arrêt.
- **c)** l'instruction qui mène vers la condition d'arrêt.

Proposition 2 :

Lire (F, e) représente
- **a)** l'appel récursif.
- **b)** la condition d'arrêt.
- **c)** l'instruction qui mène vers la condition d'arrêt.

Proposition 3 :

Inconnue ← 1 + FN Inconnue (F) représente
- **a)** l'appel récursif.
- **b)** la condition d'arrêt.
- **c)** l'instruction qui mène vers la condition d'arrêt.

✎ Exercice 2 (5,5 points)

Pour évaluer $\mathbf{a^n}$ ($a^n = a * a * a \ldots * a$), avec **a** et **n** deux entiers naturels, on a besoin de **n−1** multiplications. En informatique, l'algorithme d'exponentiation rapide est un algorithme utilisé pour calculer rapidement des grandes puissances entières. Le principe de cet algorithme est basé sur le fait qu'on a :

$a^n = a^{n/2} * a^{n/2}$ lorsque **n** est pair et $a^n = a * a^{(n-1)/2} * a^{(n-1)/2}$ lorsque **n** est impair.

D'où :

$$\mathbf{a^n} = \begin{cases} \mathbf{1} & \text{si } \mathbf{n = 0} \\ \mathbf{a^{n/2} * a^{n/2}} & \text{si } \mathbf{n} \text{ est pair} \\ \mathbf{a * a^{(n-1)/2} * a^{(n-1)/2}} & \text{si } \mathbf{n} \text{ est impair} \end{cases}$$

Travail demandé :

1) Ecrire une fonction récursive **Expo_rapide (a,n)** qui permet de calculer $\mathbf{a^n}$ en utilisant le principe décrit précédemment.
2) En faisant appel à la fonction **Expo_rapide** de la question **1**, écrire une fonction **Exponentielle(x)** qui permet de calculer une valeur approchée de e^x (l'exponentielle d'un entier naturel **x**) à **epsilon** prés (**epsilon** = 10^{-5}), sachant que $e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = \mathbf{1} + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$ avec $n!$ représente la factorielle de n.

   N.B. : La factorielle d'un entier naturel **n** noté **n!** est définie par la formule **n! = n*(n-1)*(n-2)*…*1** avec **0! = 1** et **1! = 1**

3) Soit la fonction f définie par :

$$f(x) = \frac{1}{x^n * e^x}$$

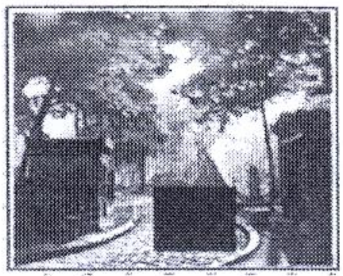
   Afin de vérifier que $\lim_{x \to +\infty} f(x) = 0$, utiliser les modules définies précédemment pour écrire un algorithme d'une procédure **Verif (FLim, n)** qui permet de stocker dans un fichier texte **FLim**, les valeurs de $f(x)$ en commençant par **x = 1** et en faisant varier **x** d'un pas égal à **1**. L'écriture dans le fichier **FLim** s'arrête lorsque $f(x)$ devient inférieure ou égale à $\mathbf{10^{-5}}$.

   N.B. :
   - Chaque valeur $f(x)$ sera stockée dans une ligne du fichier **FLim**.
   - Le candidat n'est pas appelé à saisir l'entier naturel **n**.

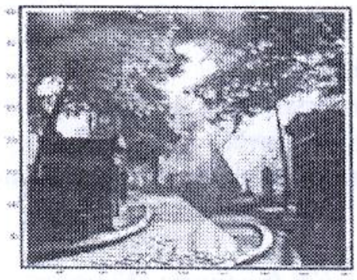
✎ Exercice 3 (5 points)

Le filtre médian est un traitement qui s'applique à une image pour l'améliorer en corrigeant les pixels isolés. Pour un pixel isolé de l'image, le filtre médian calcule sa nouvelle valeur en tenant compte des autres pixels voisins.

L'image avant amélioration



L'image après amélioration



Pour une matrice **M** remplie par les valeurs des pixels d'une image, le principe du filtre médian est le suivant:

Etape 1 : Stocker dans un tableau **T** la valeur du pixel isolé à modifier ainsi que toutes les valeurs des cases voisines de la matrice **M**.

N.B. : **T** contient **9** entiers.

Etape 2 : Trier le tableau **T** par ordre croissant.

Etape 3 : Attribuer la médiane des valeurs du tableau **T** au pixel isolé sachant que la médiane est la valeur "**milieu**" (50 % des valeurs sont plus sombres et 50% sont plus claires).

Exemple :

On se propose d'appliquer le principe du filtre médian à la matrice **M** ci-dessous représentant une partie d'une image.

M

	1	2	3
1	125	221	154
2	**91**	**210**	**90**
3	**77**	**10**	**95**
4	**115**	**151**	**68**

Pour le pixel **P** isolé de coordonnées **(3,2)**, les étapes à appliquer sont :

Etape 1 :

T

1	2	3	4	5	6	7	8	9
91	210	90	77	**10**	95	115	151	68

Etape 2 :

T trié :

1	2	3	4	5	6	7	8	9
10	68	77	90	**91**	95	115	151	210

Etape 3 :

La valeur "**milieu**" est T[5]=**91** qui sera attribuée au pixel P de coordonnées **(3,2)**
D'où la matrice **M** ci-dessous :

	1	2	3
1	125	221	154
2	91	210	90
3	77	**91**	95
4	115	151	68

Travail demandé :

En appliquant le principe du filtre médian décrit précédemment, écrire un algorithme d'une procédure **Corriger_Pixel(M, i, j)** qui permet de corriger un pixel **P** isolé de coordonnés**(i,j)** dans une matrice **M** de **L** lignes et **C** colonnes (avec $1<i<L$ et $1<j<C$).

N.B. :

- **M** contient **L*C** entiers et elle est de type **Mat**.
- Le candidat n'est pas appelé à saisir **M, L, C, i** et **j**.

✎ Exercice 4 (6,5 points)

On se propose de réaliser la conversion d'un entier naturel strictement positif **N** dans une base **B** donnée (avec $2 \leq B \leq 16$).
Pour cela on effectue des divisions euclidiennes par **B**, les restes successifs seront rangés dans un tableau **Restes** à **Rmax** éléments (avec **Rmax = 15**) puis on inverse les éléments du tableau **Restes** et on les concatène tout en convertissant les valeurs supérieures ou égales à **10** (dans le cas où la base **B** >10) en leurs équivalents dans la base **B**.

Travail demandé :

1) Ecrire un algorithme d'une procédure **SAISIE(P, Binf, Bsup)** qui permet de saisir un entier naturel **P** tel que $Binf \leq P \leq Bsup$ (avec Binf et Bsup deux entiers naturels).
2) Ecrire un algorithme d'une procédure **RANGER (N, B, RESTES, NbreR)** qui permet de :
   - ➢ Ranger dans un tableau **RESTES** les restes successifs de la suite des divisions euclidiennes par **B** jusqu'à obtenir un quotient égal à 0 (dans la première division euclidienne on divise **N** par **B**, puis on divise le quotient obtenu par **B**, etc.).
   - ➢ Calculer le nombre des restes **NbreR**.
3) Ecrire un algorithme d'une procédure **RENVERSER(RESTES, NbreR)** qui renverse les **NbreR** éléments rangés dans le tableau **RESTES**.
   (Permuter **RESTES [1]** avec **RESTES [NbreR]**, **RESTES [2]** avec **RESTES [NbreR-1]**, etc.)
4) Ecrire un algorithme d'une fonction **CONVERT(C)** qui permet de retourner le caractère qui correspond à l'entier **C** (avec $0 \leq C \leq 15$).
   Exemples : CONVERT (0) retourne le caractère "0", **CONVERT (9)** retourne le caractère "9", **CONVERT (10)** retourne le caractère "A", **CONVERT (15)** retourne le caractère "F"
5) Ecrire un algorithme d'une procédure **CONCATENATION(RESTES, NbreR)** qui, en utilisant la fonction **CONVERT** et la procédure **RENVERSER**, affiche l'équivalent du nombre N dans la base **B** en concaténant les éléments du tableau **RESTES**.

RÉPUBLIQUE TUNISIENNE	**EXAMEN DU BACCALAURÉAT - SESSION 2022**	
	Session de contrôle	**ANCIEN RÉGIME**
MINISTÈRE DE L'ÉDUCATION	Épreuve : **Algorithmique et Programmation**	Section : **Sciences de l'informatique**
	Durée : **3h**	Coefficient de l'épreuve : **2.25**

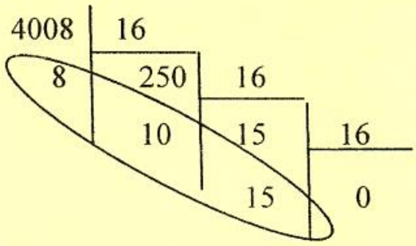
✎✎✎✎✎

N° d'inscription | | | | | | |

6) En faisant appel uniquement aux modules déjà définis, écrire un algorithme d'un programme principal intitulé **CONVERSION** qui permet de saisir un entier naturel **N** (avec **100 ≤ N ≤ 20000**) et une base **B** (avec **2 ≤ B ≤ 16)** et d'afficher le résultat de la conversion du nombre décimal **N** dans la base **B**.

Exemple : Pour **N**=4008 et **B**=16

- On procède à des divisions successives par **B**.



Après appel de la procédure **RANGER**, le tableau **RESTES** sera :

8	10	15

- Après appel de la procédure **RENVERSER**, le tableau **RESTES** sera :

15	10	8

- Après appel de la procédure **CONCATENATION**, le résultat affiché est : "**FA8**".
  En effet :
  CONVERT(15) retourne "**F**", CONVERT(10) retourne "**A**" et CONVERT(8) retourne "**8**".