Section : .... N° d'inscription : .... Série : ....

Nom et prénom : ....

Date et lieu de naissance : ....

Signature des surveillants

....

....

*Épreuve : **Algorithmique et Programmation** - Section : **Sciences de l'informatique** - Session de contrôle 2022*

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
La page 1/4 est à remplir par le candidat et à rendre avec sa copie

✎ Exercice 1 (3,5 points)

Soient le tableau de déclaration des nouveaux types (**TDNT**) et le tableau de déclaration des objets (**TDO**) suivants :

TDNT

Types
Renseignements = Enregistrement
Nom : Chaîne
Num : Octet
Etat_Civil : Caractère
Fin Renseignements
Eleve = Fichier de Renseignements
Fent = Fichier d'entiers
Tsec = Tableau de 50 Renseignements

TDO

Objet	Type/Nature
F	Texte
E	Renseignements
P	Booléen
B	Octet
M	Réel
Felv	Eleve
Fe	Fent
T	Tsec

Dans le tableau ci-dessous, valider chacune des instructions en mettant dans la case correspondante de la 2ème colonne la lettre **V** si l'instruction est valide ou la lettre **F** dans le cas contraire. Justifier la réponse si l'instruction est invalide.

Instruction	Valide/Invalide	Justification
B ← Fin_Fichier(F)	....	....
Ecrire (F, E.Eta_Civil)	....	....
P ← E.Num > 10	....	....
Ecrire (Felv, E.Nom)	....	....
Ecrire (Fe, M)	....	....
T[2] ← E.Num	....	....

N° d'inscription

Exercice 2 (5,5 points)

Pour évaluer 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 :

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

Travail demandé :

1) Ecrire une fonction récursive **Expo_rapide (a,n)** qui permet de calculer 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 $e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$ 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 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 (4,5 points)

En mathématique, une matrice carrée **M** de dimension **n x n** est dite stochastique (ou encore matrice de **Markov**) lorsque chaque élément de la matrice est un réel de l'intervalle **[0, 1]** et la somme des éléments de chaque ligne est égale à **1**.

Un tableau **T** de **n** réels est dit stable pour une matrice stochastique **M** lorsque le tableau **P** résultat du produit de **T** et **M** vérifie **P = T (T x M = P = T)**, sachant que le tableau **P** est obtenu comme suit :

$$P[j] = \sum_{i=0}^{n-1} M[i,j] * T[i] \qquad avec\ 0 \leq j \leq n-1$$

Exemple : Pour la matrice carrée **M** de dimension **3x3** et le tableau **T** de **3** éléments suivants :

M

	0	1	2
0	0,5	0,3	0,2
1	0,2	0,8	0
2	0,3	0,3	0,4

T

0	1	2
3	6	1

- **M** est une matrice stochastique puisque les éléments de **M** sont tous des réels de l'intervalle **[0,1]** et la somme des éléments de chaque ligne est égale à **1**.
  En effet :
  La somme des éléments de la $1^{ère}$ ligne est égale à M[0,0] + M[0,1] + M[0,2] = 0,5+0,3+0,2 = **1**
  La somme des éléments de la $2^{ème}$ ligne est égale à M[1,0] + M[1,1] + M[1,2] = 0,2+0,8+0 = **1**
  La somme des éléments de la $3^{ème}$ ligne est égale à M[2,0] + M[2,1] + M[2,2] = 0,3+0,3+0,4 = **1**
- **P**, le tableau résultat du produit **M** x **T** est :

P

0	1	2
3	6	1

  En effet :
  P[0] = M[0,0]*T[0] + M[1,0]*T[1] + M[2,0]*T[2] = 0,5*3+0,2*6+0,3*1 = **3** qui est égal à **T[0]**
  P[1] = M[0,1]*T[0] + M[1,1]*T[1] + M[2,1]*T[2] = 0,3*3+0,8*6+0,3*1 = **6** qui est égal à **T[1]**
  P[2] = M[0,2]*T[0] + M[1,2]*T[1] + M[2,2]*T[2] = 0,2*3+0*6+0,4*1 = **1** qui est égal à **T[2]**
- **T** est dit stable pour **M** car **M** est stochastique et le produit **P** de **T** et **M** est égal à **T**.

Travail demandé :

1) Déclarer un type pour chacune des variables **M** et **T**.
2) Ecrire un algorithme d'une fonction **Stochastique(M,n)** qui permet de vérifier si la matrice carrée **M** de dimension **n x n** est stochastique.
   N.B. : Le candidat n'est pas appelé à saisir **M** et **n**.
3) Ecrire un algorithme d'une fonction **M_Stable(M,n,T)** qui permet de vérifier si un tableau **T** de **n** réels est stable pour la matrice carrée **M** de dimension **n x n**.
   N.B. : Le candidat n'est pas appelé à saisir **M**, **n** et **T** et on supposera que la matrice **M** est stochastique.

✎ 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 ≤ B ≤ 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**.

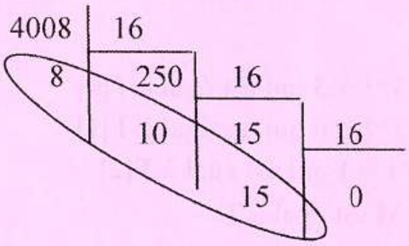
Travail demandé :

1) Ecrire un algorithme d'une procédure **SAISIE(P, Binf, Bsup)** qui permet de saisir un entier naturel **P** tel que **Binf ≤ P ≤ 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 [0]** avec **RESTES [NbreR-1]**, **RESTES [1]** avec **RESTES [NbreR-2]**, 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**.
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 $100 \leq N \leq 20000$) et une base **B** (avec $2 \leq B \leq 16$) 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"**.