

Section : N° d'inscription : Série :

Nom et prénom :

Date et lieu de naissance :

Signatures des surveillants
.....
.....



*Épreuve : Algorithmique et Programmation - Section : Sciences de l'informatique
Session principale 2023*

20

*Le sujet comporte 5 pages numérotées de 1/5 à 5/5.
Les réponses aux questions de l'exercice 1 doivent être rédigées sur les pages 1/5 et 2/5
qui doivent être remises avec la copie*

Exercice 1 : (4,5 points)

Soient les deux structures de données "Date" et "Joueur" décrites comme suit :

- La structure de données "Date" :

Champ	Jour	Mois	Annee
Type	Entier	Chaîne de caractères	Entier

- La structure de données "Joueur" :

Champ	Code	Nom_prenom	Date_naissance	Prime
Type	Entier	Chaîne de caractères	Date	Réel

- 1) En algorithmique, qu'appelle-t-on la structure de données "Date" ?

.....

- 2) Pour la structure de données "Joueur" décrite précédemment, valider chaque proposition en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** dans le cas contraire.

Une variable ayant la structure "Joueur" regroupe un nombre infini d'informations.

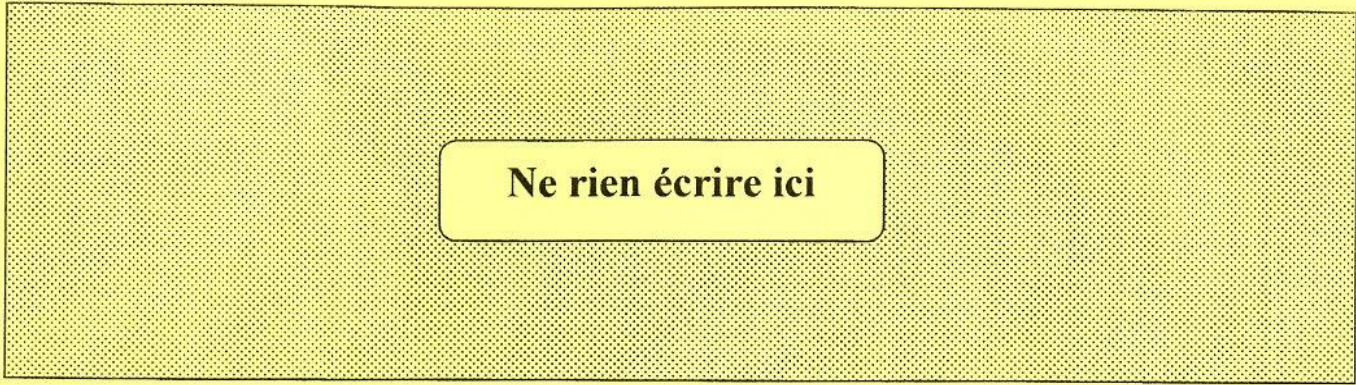
La définition d'un type pour la structure "Joueur" nécessite la définition d'un type pour la structure "Date".

Pour afficher le contenu d'une variable j ayant la structure "Joueur", on utilise l'instruction algorithmique **Ecrire(j)**.

- 3) Dresser le tableau de déclaration des nouveaux types pour les structures de données "Date" et "Joueur".

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

.../4,5



Ne rien écrire ici

- 4) Soit **T** un tableau de **10** éléments de type "**Joueur**".
 - a) Déclarer en algorithmique un type pour le tableau **T**.

.....

.....

.....

.....

- b) Ecrire une procédure nommée **Trier** qui permet de trier le tableau **T** dans l'ordre décroissant des primes.

NB : Le candidat est appelé à utiliser une procédure **Permuter (V1,V2)**, sans la développer, qui permet de permuter le contenu de deux variables **V1** et **V2** de type "**Joueur**".

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

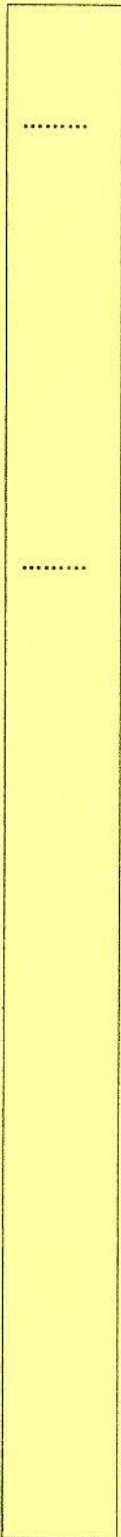
.....

.....

.....

.....

.....



RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session principale 2023
	Épreuve : Algorithmique et Programmation	Section : Sciences de l'informatique
	Durée : 3h	Coefficient de l'épreuve : 2

N° d'inscription



Exercice 2 : (2,5 points)

Soit N un entier positif et soit la trace d'exécution manuelle de la fonction **Inconnue(N)** pour $N = 5142$:

$$\text{Inconnue}(5142) = 2 + \text{Inconnue}(514)$$

$$\quad \hookrightarrow \text{Inconnue}(514) = 4 + \text{Inconnue}(51)$$

$$\quad \quad \hookrightarrow \text{Inconnue}(51) = 1 + \text{Inconnue}(5)$$

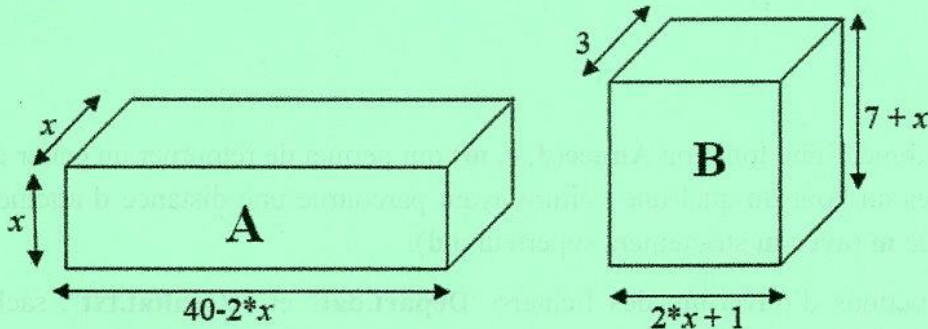
$$\quad \quad \quad \hookrightarrow \text{Inconnue}(5) = 5$$

Travail demandé :

- 1) En se référant à la trace d'exécution manuelle précédente, donner le résultat retourné par la fonction **Inconnue** pour $N = 5142$
- 2) Dédurre le rôle de la fonction **Inconnue(N)**.
- 3) Ecrire un algorithme récursif pour la fonction **Inconnue(N)**.

Exercice 3 : (2,25 points)

Une société se propose de choisir un emballage pour son nouveau produit à partir de deux boîtes **A** et **B** de dimensions différentes illustrées ci-dessous :



Travail demandé :

Sachant que les deux boîtes doivent avoir le même volume, écrire un algorithme d'une fonction qui reçoit en paramètre **epsilon** et le **pas** de variation de x et retourne une valeur approchée de x pour laquelle les volumes des deux boîtes **A** et **B** soient égaux à epsilon près (avec $\text{pas} \leq \text{epsilon}/100$).

NB :

- Le volume de la boîte A est égal à $(40-2*x)*x*x$
- Le volume de la boîte B est égal à $(2*x+1)*(7+x)*3$
- La variation de x commence à partir de 0

Exercice 4 : (5 points)

Une agence de location de voitures se propose de gérer le budget de renouvellement de ses **Nb** voitures. Pour se faire, l'agence calcule pour chaque voiture le nombre d'années au bout duquel elle doit être renouvelée. Ce nombre dépend de la valeur du kilométrage initialement parcouru par la voiture (**d**), du nombre de kilomètres parcourus chaque année (**r**) et de la valeur maximale prévue des kilomètres à parcourir (**m**).

La valeur du kilométrage parcouru par une voiture est initialement égale à **d** et elle augmente chaque année d'une même valeur **r**, donc elle forme une suite arithmétique **U** de raison **r** :

$$\begin{cases} U_1 = d \\ U_n = U_{n-1} + r \quad \text{pour } n \geq 2 \end{cases}$$

Avec :

- **d** : un entier qui représente la valeur du kilométrage initialement parcouru.
- **r** : un entier qui représente le nombre de kilomètres parcourus chaque année.

Exemple : Pour **d = 10000** km, **r = 40000** km et **m = 400000** km, le nombre d'années au bout duquel la voiture doit être renouvelée est égale à **10** ans.

Les informations des **Nb** voitures de l'agence de location sont stockées dans un fichier d'enregistrements "**Depart.dat**" déjà rempli. Chaque enregistrement du fichier contient les champs suivants :

- **immat** : une chaîne de caractères qui représente le numéro d'immatriculation de la voiture.
- **d** : un entier naturel qui représente la valeur du kilométrage initialement parcouru.
- **r** : un entier naturel qui représente le nombre de kilomètres parcourus chaque année.
- **m** : un entier naturel qui représente la valeur maximale prévue des kilomètres à parcourir.

On se propose d'écrire un algorithme d'une procédure nommée **Renouv** qui, à partir du fichier "**Depart.dat**", permet de générer un fichier texte "**Resultat.txt**" contenant dans chaque ligne le numéro d'immatriculation de la voiture suivi du nombre d'années pour atteindre la distance maximale prévue séparés par le caractère espace.

Travail demandé :

- 1) Ecrire un algorithme d'une fonction **Annee(d, r, m)** qui permet de retourner un entier qui représente le nombre d'années au bout duquel une voiture ayant parcourue une distance **d** atteindra une distance maximale prévue **m** (avec **m** strictement supérieur à **d**).
- 2) Ecrire les instructions d'ouverture des fichiers "**Depart.dat**" et "**Resultat.txt**", sachant qu'ils sont stockés sur la racine du disque **D**.
- 3) Ecrire un algorithme de la procédure **Renouv**, pour générer le fichier "**Resultat.txt**", en faisant appel à la fonction **Annee**.

NB : Déclarer les nouveaux types nécessaires.

Exercice 5 : (5,75 points)

Soit M une matrice de $NL \times NC$ entiers (avec $3 \leq NL \leq 15$ et $3 \leq NC \leq 15$) dont les éléments d'une même colonne sont distincts.

On se propose de déterminer la valeur minimale des maximums des colonnes de la matrice M appelée **MiniMax**. Pour ce faire on détermine le maximum de chaque colonne puis on détermine la valeur minimale des maximums obtenus.

Travail demandé :

- 1) Ecrire un algorithme d'une fonction **Max_Colonne**(NL, j, M) qui, pour une colonne j de la matrice M , permet de retourner le numéro de la ligne de son maximum.
- 2) Ecrire un algorithme d'une procédure nommée **Recherche_Minimax** qui permet de déterminer et d'afficher les positions de **MiniMax** dans la matrice M en utilisant le module **Max_Colonne** et en appliquant le procédé suivant :
 - Remplir un tableau d'enregistrements T par les positions des maximums des colonnes de la matrice M . Chaque enregistrement du tableau T correspond au maximum d'une colonne et il est formé des deux champs suivants :
 - **C** : Le numéro de la colonne de la matrice M où se trouve la valeur maximale d'une colonne.
 - **L** : Le numéro de la ligne de la matrice M où se trouve la valeur maximale d'une colonne.
 - Chercher et afficher la valeur du **MiniMax**.
 - Afficher à partir du tableau T , la(les) position(s) du **MiniMax**.

Exemple :

Pour $NL = 5$, $NC = 6$ et la matrice M suivante :

	0	1	2	3	4	5
0	900	481	300	99	210	110
1	300	696	29	133	-37	61
2	-155	951	153	100	300	49
3	31	-11	81	-1	54	452
4	13	17	-541	300	696	301

Le tableau T contiendra les enregistrements suivants :

0	1	2	3	4	5
0 0	1 2	2 0	3 4	4 4	5 3

La procédure **Recherche_Minimax** affiche :

MiniMax = 300

Les positions sont : (2,0) (3,4)

En effet, **MiniMax** est égale à **300** et se trouve dans la colonne numéro **2**, la ligne numéro **0** et la colonne numéro **3**, la ligne numéro **4** dans la matrice M .