RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session principale	2024
	Épreuve : **Algorithmique et Programmation**	Section : **Sciences de l'informatique**	
	Durée : **3h**	Coefficient de l'épreuve : **2**	

N° d'inscription | | | | | | |

Le sujet comporte 4 pages numérotées de 1 sur 4 à 4 sur 4.

Exercice 1 : (3 points)

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

Fonction Inconnue(n, k : entier):……………
DEBUT
Si k > n Div 2 Alors
Retourner Vrai
Sinon Si n Mod k = 0 Alors
Retourner Faux
Sinon Retourner Inconnue(n, k+1)
FinSi
FIN

Travail demandé :

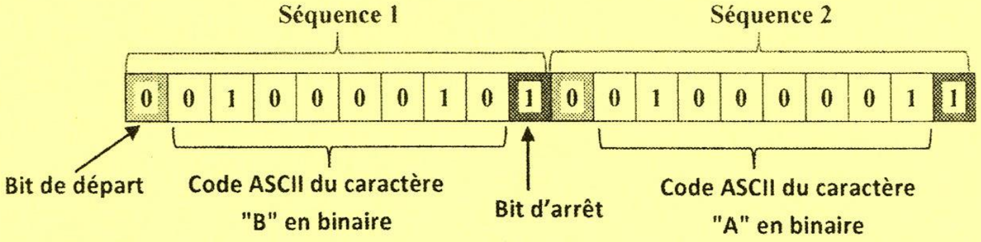
1) Donner le type de retour de la fonction **Inconnue**.
2) Sachant que la valeur de **k** lors de l'appel de la fonction **Inconnue** est toujours égale à **2** :
   a. Donner une trace d'exécution manuelle de la fonction **Inconnue** pour les deux cas suivants :
      - Cas 1 : **n = 4**
      - Cas 2 : **n = 11**
   b. En déduire le rôle de la fonction **Inconnue**.
3) Ecrire un algorithme de la fonction **Inconnue(n)** en utilisant un traitement itératif donnant le même résultat.

Exercice 2 : (5 points)

La méthode de transmission en série asynchrone est une technique de transmission de données numériques qui permet d'envoyer les données d'un message caractère par caractère. L'envoi d'un caractère se fait selon le principe suivant :

- Convertir le code ASCII du caractère à transmettre en binaire sur **8 bits**.
- Ajouter à gauche des 8 bits obtenus un bit de départ dont la valeur est **0** et à droite un bit d'arrêt dont la valeur est **1** pour former une séquence contenant **10 bits**.

Exemple : pour la chaîne de caractères "**BA**" on obtient les deux séquences suivantes :



On se propose de vérifier la validité du contenu d'un tableau **T** contenant le codage en binaire, d'un message initial formé uniquement par des lettres majuscules.

Le contenu du tableau **T** est valide s'il vérifie les contraintes suivantes :

- La taille du tableau **T** est un multiple de **10**.
- Chaque case du tableau **T** doit contenir soit la valeur **0** soit la valeur **1**.
- Chaque séquence de **10** bits doit commencer par **0** (bit de départ) et doit se terminer par **1** (bit d'arrêt).
- Les **8** bits compris entre le bit de départ et le bit d'arrêt de chaque séquence doivent représenter l'équivalent binaire du code ASCII d'une lettre majuscule.

Exemple :

Pour T=

0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1

Le contenu du tableau **T** est valide car :

- La taille du tableau **T** est égale à 30 qui est un multiple de 10.
- Toutes les cases du tableau **T** ne contiennent que les valeurs **0** et **1**.
- Chaque séquence de **10** bits commence par **0** et se termine par **1**.
- Les **8** bits compris entre le bit de départ et le bit d'arrêt de chaque séquence représentent l'équivalent binaire du code ASCII d'une lettre majuscule.

  En effet :

  - l'équivalent en décimal des 8 bits **01000010** est **66** qui correspond au code ASCII de la lettre "**B**".
  - l'équivalent en décimal des 8 bits **01000001** est **65** qui correspond au code ASCII de la lettre "**A**".
  - l'équivalent en décimal des 8 bits **01000011** est **67** qui correspond au code ASCII de la lettre "**C**".

Travail demandé :

Ecrire un algorithme d'une fonction booléenne **Verif (T , N)** qui permet de vérifier, selon les contraintes décrites précédemment, si le contenu du tableau **T** de type **Tab** et de taille **N** est valide ou non.

Exercice 3 : (4,5 points)

On se propose de vérifier si un entier **M**, supérieur ou égal à **2**, est **n-rond**.

M est dit **n-rond** s'il existe un entier **n** tel que le plus grand facteur premier de **M**, noté **P**, vérifie la condition $P \leq \sqrt[n]{M}$ (avec $\sqrt[n]{M}$ est la racine **n**$^{\text{ième}}$ de **M**).

Afin de calculer une valeur approchée de la racine **n**$^{\text{ième}}$ de **M** ($\sqrt[n]{M}$), on utilise la suite x définie comme suit :

$$x \begin{cases} x_0 = \dfrac{M}{2} \\ \\ x_k = \dfrac{1}{n}\left((n-1)\, x_{k-1} + \dfrac{M}{(x_{k-1})^{n-1}}\right) \qquad \textit{pour } k \geq 1 \end{cases}$$

Travail demandé :

1. Ecrire un algorithme d'une fonction **RacineN (M , n)** qui permet de retourner une valeur approchée de la racine **n**$^{\text{ième}}$ de **M**, en utilisant la suite x. Le calcul s'arrête lorsque $| x_k - x_{k-1} | \leq 10^{-4}$ et la valeur approchée de $\sqrt[n]{M}$ correspond alors au dernier terme calculé x_k.

2. Ecrire un algorithme d'une fonction **Facteur (M)** qui permet de retourner le plus grand facteur premier **P** de l'entier **M**.

Exemples :

- Pour **M=21** la fonction **Facteur** retourne **7** car sa décomposition en facteurs premiers donne **21** = 3 * 7.
- Pour **M=432** la fonction **Facteur** retourne **3** car sa décomposition en facteurs premiers donne $432 = 2^4 * 3^3$.

3. En faisant appel aux deux fonctions **RacineN** et **Facteur**, écrire un algorithme d'une fonction **NRond (M)** qui permet de retourner le plus grand entier **n** qui vérifie $P \leq \sqrt[n]{M}$ dans le cas où **M** est **n-rond** et de retourner **-1** dans le cas contraire (avec **P** est le plus grand facteur premier de **M**).

Exemples :

- **NRond (432)** retourne **5** car pour **P=3,** qui est le plus grand facteur premier de **432**, l'entier **n**=**5** correspond au plus grand entier qui vérifie $3 \leq \sqrt[5]{432}$. En effet on a :

n	Racine $n^{ième}$	Constatation
2	$\sqrt[2]{432}$= 20,7846	$3 < \sqrt[2]{432}$
3	$\sqrt[3]{432}$ = 7,5595	$3 < \sqrt[3]{432}$
4	$\sqrt[4]{432}$ = 4,5590	$3 < \sqrt[4]{432}$
5	$\sqrt[5]{432}$= 3,3658	**$3 < \sqrt[5]{432}$**
6	$\sqrt[6]{432}$= 2,7494	$3 > \sqrt[6]{432}$

- **NRond (21)** retourne **-1** car pour **P=7**, qui est le plus grand facteur premier de **21**, il n'existe pas un entier **n** tel que $7 < \sqrt[n]{21}$. En effet $7 > \sqrt[2]{21} = 4,5825$.

Exercice 4 : (7,5 points)

Dans le but d'attribuer des cadeaux à des invités présents dans une soirée, on se propose de choisir les personnes dont le nom est triangulaire et ayant le score le plus élevé. Ces personnes seront déclarées gagnantes.

Un nom est dit **triangulaire**, si son score est un nombre triangulaire.

Le score d'un nom est la somme des rangs dans l'alphabet de toutes les lettres qui le constituent.

Un nombre S est triangulaire s'il existe un entier n tel que :

$$S = \frac{n\,(n+1)}{2} \quad pour \quad n \geq 1$$

Exemple 1 : "Saber" est triangulaire car son score S est un nombre triangulaire.

En effet :

$$S = \underbrace{19}_{\text{Rang de S dans l'alphabet}} + \underbrace{1}_{\text{Rang de a dans l'alphabet}} + \underbrace{2}_{\text{Rang de b dans l'alphabet}} + \underbrace{5}_{\text{Rang de e dans l'alphabet}} + \underbrace{18}_{\text{Rang de r dans l'alphabet}} = \mathbf{45}$$

S = **45** est un nombre triangulaire car $\mathbf{45 = \frac{9*(9+1)}{2}}$

Exemple 2 : "**Ali**" n'est pas triangulaire car son score **S** n'est pas un nombre triangulaire.

En effet :

$$S = \underbrace{1}_{\text{Rang de A dans l'alphabet}} + \underbrace{12}_{\text{Rang de l dans l'alphabet}} + \underbrace{9}_{\text{Rang de i dans l'alphabet}} = 22$$

$S = 22$ n'est pas un nombre triangulaire car il n'existe pas un entier **n** tel que $22 = \frac{n*(n+1)}{2}$

Sachant que les noms des invités sont enregistrés dans le fichier texte "**Invites.txt**", on procède comme suit pour sélectionner les personnes gagnantes :

- Transférer, à partir du fichier "**Invites.txt**" les noms triangulaires vers un nouveau fichier d'enregistrements nommé "**Triangulaires.dat**" où chaque enregistrement est constitué de deux champs :
  - Nom : le nom de la personne.
  - Score : le score du nom de la personne.
- Afficher les noms des personnes gagnantes (ayant le score le plus élevé).

Travail demandé :

1. Ecrire un algorithme du programme principal en respectant le procédé décrit précédemment et en le décomposant en modules.
2. Ecrire un algorithme pour chaque module envisagé.

NB :

- Le fichier "**Invites.txt**" est déjà enregistré sous la racine du disque **C** et il contient au maximum **50** noms à raison d'un nom par ligne.
- Le fichier "**Triangulaires.dat**" sera créé sous la racine du disque **C**.