

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION EXAMEN DU BACCALAURÉAT SESSION 2025	Session 2025	
	Épreuve : INFORMATIQUE	Sections : Mathématiques, Sciences expérimentales et Sciences techniques
	Durée : 1h 30	Coefficient de l'épreuve: 0.5

N.B :

Corrigé et barème de notation

- Accepter toute solution algorithmique correcte.
- -0.25 par erreur.

Exercice n°1 (3.25 points = Résultat : $0.25 * 5$; Rôle : $0.5 * 4$)

Séquence algorithmique	Résultat	Rôle
$x \leftarrow 0$ Pour i de 1 à n Faire Si (n Mod i = 0) Alors $x \leftarrow x + 1$ FinSi Fin Pour Ecrire (x)	$n = 4$ 3	Déterminer et afficher le <u>nombre de diviseurs de n</u>
$s \leftarrow 0$ $ch \leftarrow \text{Convch}(n)$ Pour i de Long(ch) -1 à 0 (Pas= -1) Faire $s \leftarrow s + \text{Valeur}(ch[i])$ Fin Pour Ecrire(s)	$n = 4253$ 14	Calculer et afficher la <u>somme des chiffres de n</u>
$c \leftarrow ""$ Pour i de 0 à Long(ch) -1 Faire $c \leftarrow ch[i] + c$ Fin Pour Ecrire(c)	$ch = \text{"bac"}$ cab	Afficher <u>l'inverse de ch</u>
$i \leftarrow 1$ Tant que (i < n) et ($T[i-1] \geq T[i]$) Faire $i \leftarrow i+1$ Fin Tant que Ecrire (i = n)	1 ^{er} cas : $n = 5$ $T = \begin{array}{ c c c c c } \hline 25 & 20 & 17 & 12 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$ vrai 2 ^{ème} cas : $n = 5$ $T = \begin{array}{ c c c c c } \hline 25 & 9 & 12 & 20 & 17 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$ faux	Vérifier et afficher si T est <u>trié par ordre décroissant</u> ou non

Exercice n° 2 (3.75 points)

1) Déterminer le type de résultat retourné par la fonction **Quoi ? Entier.**

0.5

2) Soit le tableau T suivant de 10 entiers

T	112	54	59	-15	1	-5	29	17	0	47
	0	1	2	3	4	5	6	7	8	9

a) Donner le résultat de chacun des appels ci-dessous de la fonction **Quoi.**

0.25 * 3

- $\text{Quoi}(T, 4, 8) = 5$
- $\text{Quoi}(T, 6, 9) = 8$
- $\text{Quoi}(T, 0, 9) = 3$

b) Dédurre le rôle de la fonction **Quoi.**

0.5 = $0.25 * 2$

La fonction *Quoi* permet de retourner l'indice du minimum d'une partie d'un tableau T délimitée par un indice de début *d* et un indice de fin *f*.

- 3) Ecrire l'algorithme d'un module intitulé **ORDONNER** qui permet de réordonner les éléments d'un tableau **V** de type **Tab** de **N** entiers dans un ordre croissant. L'algorithme **ORDONNER** doit faire appel à la fonction **Quoi**.

NB. Le candidat n'est pas appelé à déclarer le nouveau type : TAB.

Procédure **ORDONNER** (@V : Tab, N : Entier)

DEBUT

Pour *i* de 0 à (N - 2) Faire

p ← **Quoi** (V, *i*+1, N-1)

Si V[*i*] > V[*p*] **Alors**

 Aux ← V[*i*]

 V[*i*] ← V[*p*]

 V[*p*] ← Aux

FinSi

Fin Pour

FIN

Objet	Type/Nature
Aux, i, p	Entier
Quoi	Fonction

Entête de la procédure Ordonner	2 points =
Parcours	0.5
Appel de la fonction Quoi	0.5
Permutation	0.5

Problème (13 points)

ALGORITHME Nombres_Keith

DEBUT

Saisir (N1)

Remplir (T1, N1)

Generer (T1, N1, T2, N2)

Afficher (T2, N2)

FIN

Nouveau Type
Tab= Tableau de 20 entiers

Objet	Type/Nature
Saisir, Remplir, Generer, Afficher	Procédure
T1, T2	Tab
N1, N2	Entier

Procédure Saisir (@N : Entier)

DEBUT

Répéter

 Ecrire("Donner la taille du tableau T1 :")

 Lire (N)

Jusqu'à N ∈ [5..20]

FIN

Procédure Remplir (@T : Tab, N : Entier)

DEBUT

Pour *i* de 0 à N-1 Faire

 Répéter

 Ecrire ("Donner un nombre de 2 à 5 chiffres :")

 Lire (T[*i*])

 Jusqu'à (Long (Convch(T[*i*])) ∈ [2..5])

Fin Pour

FIN

Objet	Type/Nature
i	Entier

Procédure Generer (T1 : Tab, N1 : Entier, @T2 : Tab, @N2 : Entier)

DEBUT

N2 ← 0
 Pour i de 0 à N1-1 Faire
 Si (Keith(T1[i]) Alors
 T2[N2] ← T1[i]
 N2 ← N2+1
 FinSi
 Fin Pour

Objet	Type/Nature
i	Entier
Keith	Fonction

FIN

Fonction Keith (n : Entier) : Booléen

DEBUT

m ← Convch(n)
 Pour i de 0 à Long(m)-1 Faire
 Tkeith [i] ← Valeur(m[i])
 Fin Pour
 i ← Long(m)
 Répéter
 S ← 0
 Pour j de (i-Long(m)) à (i-1) Faire
 S ← S+ Tkeith [j]
 Fin Pour
 Tkeith [i] ← S
 i ← i+1
 Jusqu'à (Tkeith [i-1] ≥ n)
 Retourner (Tkeith [i-1] = n)

Objet	Type/Nature
m	Chaine de caractères
Tkeith	Tab
i, j, S	Entier

FIN

Procédure Afficher (T : Tab, N : Entier)

DEBUT

Pour i de 2 à 5 Faire
 Ch ← ""
 Pour j de 0 à N-1 Faire
 Si Long(Convch(T[j])) = i Alors
 ch ← ch+Convch(T[j])+" , "
 FinSi
 Fin Pour
 Si ch ≠ "" Alors
 ch ← souschaine(ch,0, Long(ch)-1)
 Ecrire (ch, " : nombre(s) de Keith de ", i, " chiffres")
 Sinon
 Ecrire ("Aucun nombre de Keith de ", i, " chiffres")
 FinSi
 Fin Pour

Objet	Type/Nature
ch	Chaine
i, j	Entier

FIN

Barème détaillé :

Programme principal : - Modularité - Cohérence (appels + conformité des paramètres)	1.5 points = 0.5 1 = (0.5+0.5)
Saisie de la taille N1 : - Lecture - Contrôle de la saisie	0.75 point = 0.25 0.5
Remplissage du tableau T1 : - Parcours - Lecture - Contrôle de la saisie	1.5 points = 0.5 0.5 0.5
Vérification d'un nombre de Keith - Calcul des k premiers termes - Initialisation - Parcours - Affectation - Calcul des termes suivants - Parcours - Calcul de la somme (initialisation + parcours+ affectation) - Affectation de la somme ($T_{Keith}[i] \leftarrow S$) - Incrémentation de i - Retour de résultat	3 points = 0.25 0.5 0.25 0.5 0.75 0.25 0.25 0.25
Génération des éléments qui sont des nombres de Keith dans T2 - Initialisation - Parcours du tableau T1 - Vérification si l'élément courant est un nombre de Keith - Affectation dans T2 - Incrémentation de N2	1.75 points = 0.25 0.5 0.5 0.25 0.25
Affichage du résultat : - Parcours - Initialisation de la chaîne ch à vide - Parcours du tableau T2 - Test de la longueur de l'élément courant - Ajout de l'élément courant à la chaîne - Test si la chaîne est vide ou non et affichages adéquats	3 points = 0.5 0.25 0.5 0.5 0.5 0.75 = 3 * 0.25
Déclaration des nouveaux types + déclaration des objets globaux	1 point = (0.5+0.5)
Déclaration des objets locaux	0.5 point