

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT	Session 2024
	Épreuve : Informatique	Sections : Mathématiques, Sciences expérimentales et Sciences techniques
	Durée : 1h 30	Coefficient de l'épreuve : 0.5

Corrigé et barème de notation

Le barème est noté sur 40 points, Pour obtenir la note sur 20, arrondir sur le deuxième chiffre après la virgule.

Exercice N°1 (6 points)

1. En exploitant la fonction **Inverse**, compléter la condition de la structure conditionnelle dans l'algorithme de la procédure **Affiche** ci-dessous afin d'afficher si un entier strictement positif x donné est symétrique ou non. (1.5 point)

Procédure Affiche (x : Entier)

DEBUT

Si $x = \text{Inverse}(x)$ Alors

Ecrire(x, "est symétrique")

Sinon

Ecrire(x, "n'est pas symétrique")

FinSi

FIN

2.

- a. Pour chaque proposition d'algorithme, compléter le tableau ci-dessous par le résultat de l'instruction suivante : **Ecrire (Inconnue (6371))** (2*2 points)

	Proposition A	Proposition B
Résultat	1736	6371

- b. En déduire la proposition de l'algorithme équivalente à la fonction **Inverse**. (0.5 point)
- Proposition A**

Exercice N°2 (10 points)

1. Compléter le tableau ci-dessous par la déclaration du nouveau type correspondant au tableau T. (1 point)

T. D. N. T.
Nouveau type
Tab = Tableau de 100 Réel

2. Ecrire, en utilisant la boucle **Tant que**, l'équivalent de la séquence d'instructions délimitée par des tirets dans l'algorithme de la fonction **Existe**.

$i \leftarrow 0$

(0.5 point)

Tant que $(T[i] \neq a)$ **Et** $(i < n-1)$ **Faire**

(2 points = 4*0.5)

$i \leftarrow i+1$

Fin Tant que

3. En se référant à l'algorithme de la fonction **Existe**, initialement donné dans l'énoncé, compléter l'algorithme de la fonction **Existe1** ci-dessous, pour simuler ce nouveau principe de recherche. (6.5 points)

Fonction Existe1 (a : Réel, T : Tab, n : Entier) : Entier

DEBUT

6.5 points =

$i \leftarrow -1$

0.25

Répéter

0.25

$i \leftarrow i+1$

0.25

Jusqu'à $(T[i]=a)$ Ou $(T[n-1-i] = a)$ Ou $(i = (n-1) \text{ div } 2)$

3 x 0.75

$x \leftarrow -1$

0.5

Si $T[i] = a$ Alors

0.5

$x \leftarrow i$

0.5

Sinon Si $T[n-1-i] = a$ Alors

0.5

$x \leftarrow n-1-i$

0.5

Fin Si

Retourner x

0.5

FIN

T.D.O.L. 0.5

Objet	Type/Nature
i, x	Entier

Problème (24 points)

1. Algorithme du programme principal

Algorithme MVP

DEBUT

Saisie($n, 5, 30, \text{"Donner le nombre de joueurs"}$)

Remplissage(TJ, n)

Saisie($m, 2, 50, \text{"Donner le nombre de journalistes"}$)

CalculScore(TJ, TS, n, m)

Affiche(TJ, TS, n)

FIN

T. D. N. T.

Nouveau type
Tab1 = Tableau de 30 Chaînes de caractères
Tab2 = Tableau de 50 Entier

T. D. O. G.

Objet	Type/Nature
n, m	Entier
TJ	Tab1
TS	Tab2
Saisie	Procédure
Remplissage	Procédure
CalculScore	Procédure
Affiche	Procédure

2. Algorithmes des modules

Procédure Saisie($@ n$:Entier, $binf, bsup$: Entier, msg :chaîne)

DEBUT

Répéter

Ecrire(msg)

Lire(n)

Jusqu'à $(binf \leq n \leq bsup)$

FIN

Procédure Remplissage(@TJ : Tab1, n : Entier)

DEBUT

Pour i de 0 à n-1 Faire

Répéter

Ecrire("Donner le nom et le prénom du joueur N°", i, " : ")

Lire(TJ[i])

Jusqu'à (Test(TJ[i]) Et Existe (TJ[i], TJ, i) = -1)

//on accepte aussi (Existe (TJ[i], TJ, i-1) = -1)

Fin Pour

FIN

T. D. O. L.

Objet	Type/Nature
i	Entier
Test	Fonction
Existe	Fonction

Procédure CalculScore(TJ:Tab1, @TS:Tab2, n:Entier, m:Entier)

DEBUT

Pour i de 0 à m-1 Faire

TS[i] ← 0

Fin Pour

Pour i de 0 à m-1 Faire

Répéter

Ecrire("Choix 1 = "), Lire(nom1)

Ecrire("Choix 2 = "), Lire(nom2)

Ecrire("Choix 3 = "), Lire(nom3)

p1 ← Existe (nom1, TJ, n)

p2 ← Existe (nom2, TJ, n)

p3 ← Existe (nom3, TJ, n)

Jusqu'à (nom1 ≠ nom2) Et (nom1 ≠ nom3) Et (nom2 ≠ nom3) Et (p1 ≠ -1) Et (p2 ≠ -1) Et (p3 ≠ -1)

TS[p1] ← TS[p1] + 5

TS[p2] ← TS[p2] + 3

TS[p3] ← TS[p3] + 1

Fin Pour

FIN

T. D. O. L.

Objet	Type/Nature
i, p1, p2, p3	Entier
nom1, nom2, nom3	Chaîne de caractères
Existe	Fonction

Procédure Affiche(TJ:Tab1, TS:Tab2, n:Entier)

DEBUT

max ← Maximum(TS, n)

Ecrire ("Le score MVP est : ", max)

Ecrire ("Le(s) joueur(s) MVP : ")

Pour i de 0 à n-1 Faire

Si TS[i] = max Alors

Ecrire(TJ[i] , " , ")

FinSi

Fin Pour

FIN

T. D. O. L.

Objet	Type/Nature
i, max	Entier
Maximum	Fonction

Fonction Maximum(T:Tab2, n:Entier):Entier

DEBUT

max ← T[0]

Pour i de 1 à n-1 Faire

Si T[i] > max Alors

max ← T[i]

FinSi

Fin Pour

Retourner max

T. D. O. L.

Objet	Type/Nature
i, max	Entier

Barème détaillé (24 points)

N.B. Il suffit de trouver un module dans le programme principal pour attribuer la note entière de la modularité.

Programme principal : <ul style="list-style-type: none"> - Modularité = (décomposition + logique d'appel) - Cohérence (appels + conformité des paramètres) 	<u>3 points =</u> $1 = (0.5 + 0.5)$ $2 = (1+1)$
Saisie de la taille N : <ul style="list-style-type: none"> - Lecture - Contrôle de la saisie (boucle + condition) 	<u>1.5 points =</u> 0.5 $1 = 0.5 + 0.5$
Remplissage du tableau TJ : <ul style="list-style-type: none"> - Parcours du tableau - Lecture - Contrôle de la saisie (boucle + appel de Test + appel de Existe) 	<u>3 points =</u> 1 0.5 $1.5 = 0.5 + 0.5 + 0.5$
Remplissage du tableau TS : <ul style="list-style-type: none"> - Initialisation du tableau TS (boucle + affectation) - Calcul score <ul style="list-style-type: none"> o Boucle o Lecture des choix o Contrôle de saisie des choix (boucle + Appel de Existe + vérification 3 votes distincts) - Affectation des points dans le tableau TS 	<u>9 points =</u> $1.5 = 1 + 0.5$ $6 =$ 1 $1.5 = 3 * 0.5$ $3.5 = 0.5 + 3 * 0.5 + 3 * 0.5$ $1.5 = 3 * 0.5$
Affichage du résultat : <ul style="list-style-type: none"> - Recherche du maximum dans TS <ul style="list-style-type: none"> o Initialisation o Parcours o Test o Affectation - Affichage du score et des joueurs MVP <ul style="list-style-type: none"> o Affichage du score MVP (message + valeur) o Affichage des joueurs MVP <ul style="list-style-type: none"> ▪ Affichage du message ▪ Parcours ▪ Test ▪ Affichage du nom du joueur 	<u>4.5 points =</u> 2 points = 0.25 1 0.5 0.25 2.5 points = $0.25 + 0.25$ 0.25 1 0.5 0.25
Déclaration des nouveaux types + Déclaration des objets globaux	<u>1.5 points = (0.5+1)</u>
Déclaration des objets locaux	<u>1.5 points</u>