

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2022	NOUVEAU RÉGIME
	Épreuve : INFORMATIQUE	Sections : Mathématiques, Sciences expérimentales et Sciences techniques
	Durée : 1h 30	Coefficient de l'épreuve : 0.5

Corrigé et barème de notation

Exercice 1 (3 points)

Soient le tableau de déclaration des objets globaux ci-dessous :

Objet	Type/Nature
n	Entier
y	Réel
Traitement	Procédure

et l'entête de la procédure **Traitement** suivant :

Procédure Traitement (@ x : réel , a : entier)

Compléter le tableau ci-dessous en mettant une croix (X) dans la case correspondante (**Valide** ou **Invalide**) pour chacun des appels de la procédure **Traitement**. Justifier votre réponse pour les appels invalides.

Appel	Valide	Invalide	Justification
Traitement (y, n)	×		<i>Il y a une compatibilité entre les paramètres formels et effectifs en nombre, ordre et type</i>
Traitement (y, n, 2)		×	<i>Les paramètres effectifs et formels ne s'accordent pas en nombre</i>
Traitement (5.5, 6)		×	<i>Le premier P. effectif doit être une variable</i>
Traitement (y, 3)	×		<i>Compatibilité entre les paramètres effectifs et formels en type, nombre, ordre et le mode de passage du 2^{ème} P. formel est en valeur c'est pour cette raison on acceptera la valeur 3 comme 2^{ème} P. formel</i>

Exercice 2 (7 points)

Soient le tableau de déclaration des nouveaux types et les deux algorithmes des fonctions **f1** et **f2** ci-dessous :

Nouveau type
Tab = tableau de 50 entiers

Algorithme de la fonction f1	Algorithme de la fonction f2												
Fonction f1 (T : Tab , n, x : entier) : entier DEBUT Cpt ← 0 Pour i de 1 à n Faire Si T[i]=x Alors Cpt ← Cpt+1 Fin Si Fin Pour Retourner Cpt FIN	Fonction f2 (T : Tab , n : entier) : entier DEBUT M ← T[1] Pour i de 2 à n Faire Si T[i]>M Alors M ← T[i] Fin si Fin Pour Retourner M FIN												
<table border="1"> <thead> <tr> <th colspan="2">TDOL</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>Cpt, i</td> <td>Entier</td> </tr> </tbody> </table>	TDOL		Objet	Type/Nature	Cpt, i	Entier	<table border="1"> <thead> <tr> <th colspan="2">TDOL</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>M, i</td> <td>entier</td> </tr> </tbody> </table>	TDOL		Objet	Type/Nature	M, i	entier
TDOL													
Objet	Type/Nature												
Cpt, i	Entier												
TDOL													
Objet	Type/Nature												
M, i	entier												

N.B : n est un entier strictement positif et inférieur ou égal à 50

Questions

1) Pour le tableau **T** suivant :

T	5	6	3	3	2	8	6	2	6	3
	1	2	3	4	5	6	7	8	9	10

a. Donner le résultat de chacun des appels ci-dessous de la fonction **f1**.

▪ $f1(T, 10, 6) = \dots\dots 3 \dots\dots$

▪ $f1(T, 6, 3) = \dots\dots 2 \dots\dots$

b. Déduire le rôle de la fonction **f1**.

..... Permet de retourner le nombre d'occurrences d'un entier x dans un tableau T de n entiers.....

c. Donner le résultat de chacun des appels suivants de la fonction **f2**.

▪ $f2(T, 10) = \dots\dots 8 \dots\dots$

▪ $f2(T, 5) = \dots\dots 6 \dots\dots$

d. Déduire le rôle de la fonction **f2**.

..... Permet de retourner le maximum d'un tableau T de n entiers ...

2) Compléter la colonne "**Réponse**" du tableau ci-après par "**Valide**" si l'algorithme de la procédure **remplir** permet le remplissage **aléatoire** de **n** cases d'un tableau **T** par des chiffres de **1** à **9** ou par "**Invalide**" dans le cas contraire.

La définition de la procédure remplir	Réponse (valide/invalide)
Procédure remplir (@ T : Tab , n : entier) DEBUT Pour i de 1 à n faire T[i] ← aléa(1,9) Fin Pour FIN	<p>.....Valide.....</p>
Procédure remplir (@ T : Tab , n : entier) DEBUT Pour i de 1 à n faire T[i] ← aléa(1,10) Fin Pour FIN	<p>.....Invalide.....</p> <p><i>Rappel : la fonction prédéfinie Aléa (vi, vf), retourne un entier aléatoire de l'intervalle [vi, vf]. Alors la boucle ci-contre peut générer la valeur 10</i></p>
Procédure remplir (@ T : Tab , n : entier) DEBUT Pour i de 1 à n faire T[i] ← aléa(0,9) Fin Pour FIN	<p>.....Invalide.....</p> <p><i>La boucle ci-contre peut générer la valeur 0</i></p>

3) Soient **T** un tableau de **n** entiers ($5 \leq n \leq 50$) rempli par des chiffres compris entre **1** et **9** et **V** un tableau de **9** cases de type entier et dont leurs indices sont de **1** à **9**.

Travail demandé :

a) Ecrire un algorithme d'un module intitulé **remplissage** qui permet de remplir le tableau **V** par le nombre d'occurrences de chaque chiffre du tableau **T**.

NB : Pour calculer le nombre d'occurrences l'algorithme solution doit faire appel à la fonction **f1**.

Exemple :

Pour **n=10** et le tableau **T** suivant :

T	5	6	3	3	2	8	6	2	6	3
	1	2	3	4	5	6	7	8	9	10

Suite à l'appel du module **remplissage** on obtient le tableau **V** suivant :

V	0	2	3	0	1	3	0	1	0
	1	2	3	4	5	6	7	8	9

- **V [3] = 3**, car le chiffre 3 apparaît 3 fois dans le tableau T
- **V [6] = 3**, car le chiffre 6 apparaît 3 fois dans le tableau T
- **V [7] = 0**, car le chiffre 7 n'apparaît pas (0 fois) dans le tableau T

Procédure remplissage (T: Tab , n : entier , @ V : Tab)

DEBUT

Pour i de 1 à 9 faire

$V[i] \leftarrow f1(T, n, i)$ // *appel de la fonction f1 qui retourne le nombre*

Fin Pour // *d'occurrence de i dans t et d'affecter le résultat à V[i]*

FIN

TDOL	
Objet	Type/nature
i	entier
f1	fonction

b) Ecrire un algorithme d'un module intitulé **Afficher** qui permet d'afficher, à partir du tableau **V**, le(s) chiffre(s) ayant le plus grand nombre d'occurrences dans le tableau **T**. {2pts}

NB : Pour déterminer le plus grand nombre d'occurrences l'algorithme solution doit faire appel à la fonction **f2**.

Exemple :

Pour le tableau **V** suivant :

V	0	2	3	0	1	3	0	1	0
	1	2	3	4	5	6	7	8	9

Suite à l'appel du module **Afficher** on obtient l'affichage suivant :

Les chiffres les plus fréquents dans le tableau sont : 3 6

Procédure Afficher (V : Tab)

DEBUT

Ecrire ("Les chiffres les plus fréquents dans le tableau sont :")

Pour i de 1 à 9 faire

Si $(V[i]=f2(V, 9))$ alors // *appel de la fonction f2 pour vérifier si*

Ecrire (" ", i) // *v[i] est un maximum ou non si oui on*

Fin si // *affiche son indice i*

Fin Pour

FIN

TDOL	
Objet	Type/nature
i	entier
f2	fonction

Problème (10 points)

1) Algorithme du programme principal :

ALGORITHME VACCIN

DEBUT

saisir(n)
remplir(T,n)
afficher(T,n)

FIN

Déclaration des nouveaux types utilisateurs

Nouveau type
Tab= tableau de 100 chaînes

Déclaration des objets globaux

Objet	Type/Nature
T	Tab
n	Entier
saisir	Procédure
remplir	Procédure
afficher	Procédure

2) Les algorithmes des modules :

1. Algorithme de la procédure saisir

Procédure saisir (@ n:entier)

DEBUT

Répéter
Ecrire ("Donner le nombre de citoyens : ")
Lire(n)
Jusqu'à ($5 \leq n \leq 100$)

FIN

2. Algorithme de la procédure remplir

Procédure remplir (@ T : tab , n:entier)

DEBUT

Pour i de 1 à n faire
Répéter
Ecrire ("donner les informations du citoyen n° ",i," : ")
Lire(T[i])
Jusqu'à (Valide(T[i])) // Appel de la fonction Valide pour vérifier la validité
// des informations du patient n° i

Fin Pour

FIN

Déclaration des objets locaux	
Objet	Type/Nature
i	Entier
Valide	Fonction

3. Algorithme de la procédure afficher

Procédure afficher (T: tab , n:entier)

DEBUT

```

Pour i de 1 à n faire                                     // Parcours du tableau
  ch ← T[i]
  p ← pos("-",ch)
  NomVac ← sous_chaine(ch,p+1,long(ch)-2) // Extraction du nom de vaccin
  d ← ch[long(ch)-1] // Extraction du nom du nombre de doses
  code ← sous_chaine(ch,0,9) // Extraction du nom du code du vaccin
  Si ((d= "2") ou( d="1") ET (NomVac ="Johnson")) alors // Test du schéma vaccinal complet
    Ecrire (" Le titulaire du code ", code," vous pouvez télécharger votre
           passe vaccinale")
  Sinon
    Ecrire (" Le titulaire du code ", code," vous êtes appelé à compléter
           votre schéma vaccinal")
Fin Si
Fin Pour
FIN

```

Déclaration des objets locaux	
Objet	Type/Nature
i,p	Entier
ch, NomVac , code	chaîne
d	caractère

Barème détaillé :

Programme principal : - Modularité - Cohérence (appels + conformité des paramètres)	1.5 points = 0.5 1 = (0.5+0.5)
Saisie du nombre de citoyens N : - Lecture - Contrôle de la saisie	1 point = 0.5 0.5
Remplissage du tableau T : - Parcours du tableau - Lecture de T[i] - Contrôle des contraintes : boucle + appel de la fonction Test + paramètre (T[i])	2 points = 0.5 0.5 1 = (0.25+0.5+0.25)
Affichage des messages : - Parcours du tableau - Extraction du nom de vaccin, code et le nombre de doses - Test du schéma vaccinal complet (2 doses ou 1 dose & Johnson) - Affichage du message pour le retrait du passe vaccinale - Affichage du message pour compléter le schéma vaccinal	3.5 points = 0.5 1 = (0.5+0.25+0.25) 0.5+0.5 0.5 0.5
Déclaration des nouveaux types + déclaration des objets globaux	1 point = (0.5+0.5)
Déclaration des objets locaux	1 point

N.B.

- *Il est à noter que ceci n'est qu'une proposition de solution.*
- *Il existe d'autres solutions possibles pour résoudre ce problème et qui ont été acceptées.*