

République Tunisienne	Section : Sc	Niveau : 4^{ème} Année
Ministère de l'Éducation	DS n° 3	Matière : Informatique
CREs : Tataouine - Kébili- Médenine & Gabès	Durée : 1h : 30 mn	Date : 14-05-2024 à 08h

Nom et prénom : – **Classe :**

*Le sujet comporte cinq pages numérotées de 1/5 à 5/5
Les réponses à l'exercice 1, l'exercice 2 et l'exercice 3 doivent être rédigées
sur cette même feuille qui doit être remise avec la copie d'examen à la fin de l'épreuve.*

EXERCICE 1 : (4 POINTS)

Soit le tableau de déclaration de nouveau type suivant :

T.D.N.T

Type
Tab = Tableau de 20 chaînes de caractères Vect = Tableau de 20 booléens

Soit le tableau de déclaration des objets suivant :

T.D.O

Objets	Type / Nature
T	Tab
V	Vect
N	Entier
CH	Chaîne de caractères

Compléter la deuxième colonne du tableau ci-dessous en indiquant le traitement nécessaire pour remplir le tableau **V**, sachant que le tableau **T** comporte **N** éléments :

Proposition	Solution
V[i] contiendra Vrai si la longueur du T[i] est supérieure à son indice i sinon Faux
V[i] contiendra Vrai si les trois premiers caractères du T[i] forment la chaîne " Bac ", sinon Faux
V[i] contiendra Faux si T[i] contient seulement des chiffres sinon Vrai
V[i] contiendra Faux si T[i] contient le caractère " @ " sinon Vrai
V[i] contiendra Vrai si le premier et le dernier caractère du T[i] sont égaux sinon Faux

EXERCICE 2 : (4 POINTS)

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

```

Fonction quoi (.....) :.....
DEBUT
  ch ← Convch(n)
  i ← 0
  Répéter
    i ← i + 1
    x ← Sous_chaine(ch, 0, i)
    test ← Valeur(x) mod i = 0
  Jusqu'à (test=Faux ou i = Long(ch) )
  Retourner test
FIN

```

1) Compléter l'entête de la fonction **quoi** :

Fonction **quoi** (.....) :.....

2) Elaborer le tableau de déclaration des objets locaux de la fonction **quoi** :

Objet	Type / Nature

3) Donner la trace d'exécution et le résultat de l'appel de la fonction **quoi** pour les deux cas suivants :

1^{er} cas : n = 3456

i				
x				
test				

Résultat :

2^{ème} cas : n = 1276

i				
x				
test				

Résultat :

4) Dédire le rôle de la fonction **quoi**.

.....

EXERCICE 3 : (4 POINTS)

On se propose de calculer le **Degré de Ressemblance DR** entre deux mots de même longueur par la formule suivante :

$$\text{DR} = \text{Nombre de caractères en communs bien placés} / \text{Longueur du mot} * 100$$

N.B : Un caractère est dit bien placé lorsqu'il occupe la même position dans les deux mots.

Exemples :

Mot1	Mot2	Degré de ressemblance (DR)
" <u>E</u> X <u>E</u> M <u>P</u> L <u>E</u> "	" <u>E</u> X <u>A</u> M <u>E</u> N <u>S</u> "	DR = 3 / 7 * 100 = 42.85
"CH <u>I</u> EN"	"NIC <u>H</u> E"	DR = 0 / 5 * 100 = 00.00
" <u>N</u> U <u>A</u> G <u>E</u> "	" <u>N</u> IC <u>H</u> E"	DR = 2 / 5 * 100 = 40.00
"TRAITEMENTS"	"INFORMATION"	DR = 0 / 11 * 100 = 00.00

Le professeur a demandé aux élèves d'écrire l'algorithme d'un module **Degre** qui permet de calculer le degré de ressemblance entre deux mots donnés.

1) Pour chacune des propositions suivantes répondre par **Vrai** ou **Faux** et corriger les propositions fausses.

a) Proposition de **Yassin** :

<p>Procédure Degre (mot1, mot2 : chaîne, DR : réel) DEBUT NCC ← 0 Pour i de 0 à Long(mot1)-1 Faire Si mot1[i] = mot2[i] alors NCC ← NCC + 1 Fin Si Fin Pour DR ← (NCC / Long(mot1)) * 100 FIN</p>	<p>Proposition correcte : Vrai / Faux Correction : </p>
---	--

b) Proposition de **Nizar** :

<p>Procédure Degre (mot1, mot2 : chaîne, @ DR : réel) DEBUT NCC ← 0 , i ← -1 Répéter i ← i + 1 Si mot1[i] = mot2[i] alors NCC ← NCC + 1 Fin Si Jusqu'à (i = Long(mot1)) DR ← (NCC / Long(mot1)) * 100 FIN</p>	<p>Proposition correcte : Vrai/ Faux Correction : </p>
---	---

c) Proposition d'Ikram :

<pre> Procédure Degré (mot1, mot2 : chaine, @ DR : réel) DEBUT NCC ← 0, i ← 0 Tantque (i < Long(mot1)) Faire Si mot1[i] = mot2[i] alors NCC ← NCC + 1 Fin Si i ← i + 1 Fin Tantque DR ← (NCC / Long(mot1)) * 100 FIN </pre>	<p>Proposition correcte : Vrai /Faux Correction :</p> <p>.....</p>
---	---

2) Compléter le programme principal ci-dessous en ajoutant une vérification de la saisie pour les deux mots, puis en appelant la procédure **Degré** et en affichant le degré de ressemblance.

NB : les deux mots sont de même longueur et non vides.

```

Algorithme Degré_Ressemblance
DEBUT
.....
Ecrire ("Mot1 = "), lire(ch1)
Ecrire ("Mot2 = "), lire(ch2)
.....
.....
.....
FIN

```

3) Ikram a dit qu'il est préférable que le module **Degré** soit une fonction.

Transformer la procédure **Degré** en une fonction.

```

Fonction Degré (.....) :.....
DEBUT
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
FIN

```

EXERCICE 4 : (8 POINTS)

Un nombre **K** est dit **Nombre Armstrong**, si la somme des puissances **p**-ièmes de ses chiffres est égale à **K**, où **p** désigne le nombre de chiffres de **K**.

Exemples : 153, 548834

- $K = 153 \rightarrow p = 3 \rightarrow 153 = 1^3 + 5^3 + 3^3$
- $K = 548834 \rightarrow p = 6 \rightarrow 548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$

On se propose d'écrire un algorithme d'un programme intitulé **Armstrong** qui permet de :

- Saisir un entier **N** ($3 \leq N \leq 100$) puis remplir un tableau **T** par **N** entiers, distincts deux à deux et chacun est composé de 3 à 9 chiffres.
- Stocker dans un tableau **TA** les nombres **Armstrong** du tableau **T**.
- Afficher le tableau **TA**.

Exemple :

Pour **N = 8** :

<i>i</i>	0	1	2	3	4	5	6	7
T :	153	620	548834	407	92727	8200	9800810	93084

En effet

<i>i</i>	0	1	2	3	4
TA :	153	548834	407	92727	93084

Le programme affichera :

153, 548834, 407, 92727, 93084,

Travail demandé :

1. Écrire l'algorithme du programme principal en le décomposant en modules.
2. Écrire les algorithmes des modules envisagés.