

Devoir de synthèse n° 2

Proposé par : Bassem Amor.

Année scolaire : 2022-2023.

Classe : 4^{ème} sciences et Math.

Matière : informatique.

Durée : 1 h 30 mn

Nom et prénom : Classe : N°:

Exercice 1 : (7.5 points)

Soit l'algorithme de la fonction INCONNUE suivant :

Fonction INCONNUE (ch,st: chaine) : entier

Début

s ← 0

pour i de 0 à long(ch)-1 faire

 Si (pos(ch[i],st)≠-1) alors

 s ← s+1

 finsi

fin pour

retourner(s)

fin

- 1- Remplir le tableau de déclaration des objets locaux de la fonction INCONNUE. (T.D.O.L)

Objet	Type

- 2- Donner le résultat de la fonction INCONNUE :

INCONNUE("abcd","ae")=.....

INCONNUE("abed","aeuiyo")=.....

INCONNUE("2*5-4","!#*-")=.....

INCONNUE("abcd","+-*/")=.....

- 3- déduire le rôle de la fonction INCONNUE.

.....
.....

- 4- Écrire l'équivalent de la fonction INCONNUE en Python.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

5- Soit T un tableau de n chaînes de caractères ($1 \leq n \leq 10$), on se propose d'écrire la procédure

filtrer qui retourne V et P avec :

- V est un tableau qui contient les éléments de T qui ne contiennent aucun caractère parmi {"+", "-", "*", "/"}
- P le nombre d'éléments de V.

a- À l'aide du T.D.N.T déclarer le type TAB pour représenter les tableaux T et V.

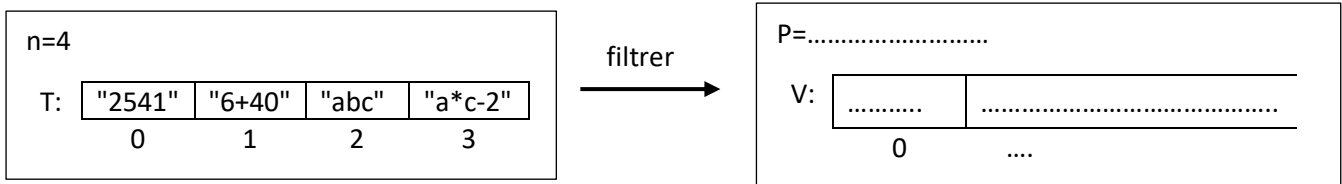
TDNT:

Type
TAB=.....

b- Compléter l'entête de la procédure filtrer.

Procédure filtrer (.....)

c- Déterminer V et P après appel de la procédure filtrer avec n et T comme ci-dessous :



d- En appelant **impérativement** la fonction INCONNUE, compléter l'algorithme de la procédure filtrer.

Procédure filtrer (.....)

début

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Fin.

e- Compléter le TDOL de la procédure filtrer.

objet	type

f- Écrire l'équivalent de la procédure filtrer en Python.

Exercice 2 : (2.5 points)

On se propose d'écrire un algorithme d'une fonction qui retourne U_n le terme d'ordre n de la suite réelle U donnée par la formule récurrente ci-dessous :

$$(U_n)_{n \in \mathbb{N}} = \begin{cases} u_0 = 3 \\ u_1 = 2 \\ u_n = 5u_{n-2} \text{ pour } n \geq 2 \end{cases}$$

Exercice 3 : (10 points)

On définit le poids d'une chaîne comme étant la somme des produits de la position de chaque voyelle dans cette chaîne par son rang dans l'alphabet français.

Si la chaîne ne contient pas de voyelles alors son poids est égal à zéro.

N.B : les voyelles sont A, E, I, O, U, Y et leurs rangs respectifs sont :1, 5, 9, 15, 21, 25

Exemple :

La chaîne "BONNE" contient 2 voyelles "O" et "E", son poids est égal à $1*15+4*5=35$

La chaîne "CHANCE" contient 2 voyelles "A" et "E", son poids est égal à : $2*1+5*5=27$

On se propose d'écrire un programme permettant de :

- Remplir un tableau **T** par **n** chaînes de caractères de **poids distincts** avec ($5 \leq n \leq 10$), chaque case doit contenir une chaîne formée uniquement par des lettres **majuscules non accentuées**.
- Former et afficher un tableau **V** qui va contenir les éléments de **T** classés par l'ordre croissant de leurs poids en appliquant le principe ci-dessous :

Pour chaque élément $T[i]_{(0 \leq i \leq n-1)}$ de **T** :

- ✓ Déterminer **j** le nombre d'éléments de **T** qui ont un poids inférieur au poids de **T[i]**.
- ✓ Placer **T[i]** dans la position **j** dans le tableau **V**.

Exemples :

Pour $N=6$ et le tableau T suivant :

T :	"SALIM"	"AMJED"	"BILEL"	"OMAR"	"YASSINE"	"KARIMA"
	0	1	2	3	4	5

i	0	1	2	3	4	5
Poids de T[i]=	28	15	24	2	67	33
j=	3	1	2	0	5	4

On aura \longrightarrow V :

"OMAR"	"AMJED"	"BILEL"	"SALIM"	"KARIMA"	"YASSINE"
0	1	2	3	4	5

Travail demandé :

- 1- Ecrire un algorithme du programme principal en le décomposant en modules.
- 2- Ecrire un algorithme pour chacun des modules proposés.