

LYCEE PILOTE BEJA
ANNEE SCOLAIRE 2021/2022

REVISION BAC THEORIQUE 2022
MATIERE : INFORMATIQUE
ENSEIGNANT : ABDELHAMID GUIZANI

SOMMAIRE

Série	Page
Série de révision N°1	02
Série de révision N°1- Correction	05
Série de révision N°2	09
Série de révision N°2- Correction	13
Série de révision N°3	19
Série de révision N°3- Correction	22
Série de révision N°4	25
Série de révision N°4- Correction	29
Série de révision N°5	33
Série de révision N°5- Correction	38
Série de révision N°6	41
Série de révision N°6- Correction	47

Exercice 1 :

On veut déterminer et afficher le nombre de diviseurs d'un entier **n** strictement positif.

Exemple :

Pour $n=6$, le nombre de diviseurs de 6 est égal à 4. En effet les diviseurs de 6 sont {1 ; 2 ; 3 ; 6}.

On vous propose les trois séquences d'instructions algorithmiques suivantes :

<i>Séquence 1</i>	<i>Séquence 2</i>	<i>Séquence 3</i>
$c \leftarrow 0$ Pour k de 1 à n faire Si $(n \bmod k = 0)$ alors $c \leftarrow c+1$ Fin Si Fin Pour Ecrire (c)	$c \leftarrow 1$ Pour k de 2 à $(n \div 2)$ faire Si $(n \bmod k = 0)$ alors $c \leftarrow c+1$ Fin Si Fin Pour Ecrire (c)	$c \leftarrow 2$ Pour k de 2 à $(n-1)$ faire Si $(n \bmod k = 0)$ alors $c \leftarrow c+1$ Fin Si Fin pour Ecrire (c)

- 1) Compléter le tableau ci-dessous par la valeur de la variable **c** après exécution de chaque séquence, et ce pour **n=4**.

Séquence	Valeur de la variable c
1	
2	
3	

- 2) Donner le numéro de la séquence qui ne permet pas d'afficher le nombre de diviseurs. Justifier votre réponse.

.....
.....

Exercice 2 :

Un médecin veut chercher la fiche d'un de ses patients en connaissant son nom. Pour cela, il utilise un tableau T contenant N noms.

- 1- Compléter la séquence algorithmique présentée ci-dessous afin de vérifier l'existence d'un nom donné NOM dans un tableau T non vide.

```
Algorithme recherche
Début
Écrire ("Donner le nom à chercher : ")
Lire ( ....)
Existe ← .....
i ← .....
Répéter
Si (T[i] = NOM) alors
    Existe ← .....
Sinon
    i ← .....
Finsi
Jusqu'à (.....) ou (.....)
Si (.....) alors
    Écrire ("Le nom recherché existe dans ce tableau.")
Sinon
    Écrire ("Le nom recherché n'existe pas dans ce tableau.")
Finsi
Fin
```

T.D.O.	
Objet	Type/Nature
T	Tableau de N chaînes
N, i	Entier
NOM	Chaîne
Existe	Booléen

- 2- Maintenant, le médecin veut chercher les numéros des fiches de ses patients ayant le même nom. Modifier la séquence algorithmique présentée ci-dessus afin d'afficher ces numéros.

.....

.....

.....

.....

.....

.....

.....

.....

Exercice 3 :

La propagation de l'épidémie Covid-19 suit une croissance exponentielle. Pour déterminer et afficher le nombre total de personnes contaminées pendant un nombre de jours donné (**N**) et pour **x** personnes initialement contaminées on utilise la formule suivante :

$$e^x = \sum_{i=0}^N \frac{(x)^i}{i!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots + \frac{(x)^N}{N!}$$

Donner un algorithme solution à ce problème.

Exercice 4 :

Un débutant en anglais veut élaborer son propre **carnet** comme étant un dictionnaire **FRANÇAIS/ANGLAIS** pour l'utiliser afin de traduire des phrases au cours de sa formation. Pour cela, on admettra qu'un traducteur du français à l'anglais peut être simplifié par une traduction de mot-à-mot.

Ce carnet **FRANÇAIS/ANGLAIS** est formé par **N** mots en français et par **N** mots en anglais, de sorte que chaque mot en français lui correspond son équivalent en anglais, avec $2 \leq N \leq NMAX$ (**NMAX** est une constante égale à 100).

Pour chaque mot ajouté en français, dans le carnet, en lui ajoute en même temps son équivalent en anglais. Sachant que chaque mot en français qu'en anglais est une chaîne non vide de longueur maximale 15 lettres non accentuées.

Après l'élaboration du carnet, on veut traduire une phrase donnée en français (contenant que des lettres et des espaces) en son équivalent en anglais. Dans le cas où l'un des mots ne figure pas dans le carnet, le mot en français va figurer dans la phrase en anglais mais entre deux accolades.

La phrase traduite doit être suivie par un message de succès ou un message d'échec indiquant le nombre de mots non traduits.

NB : On suppose que la phrase ne contient pas un espace au début, un espace à la fin et des espaces superflus (un seul espace sépare deux mots).

Exemple : Pour **N=5**, le carnet sera présenté comme suit :

FR	école	je	libre	Suis	Un
	0	1	2	3	4
ENG	School	i	free	Am	A
	0	1	2	3	4

FR contient les mots en français et **ENG** contient les mots en anglais.

- Pour la phrase "école libre"
Le résultat affiché sera : "school free : Traduction totale"
- Pour la phrase "je suis un être libre"
Le résultat affiché sera : "i am a {être} free : Traduction partielle, 1 mot(s) non traduit(s)"

On vous demande d'élaborer:

- 1) un algorithme du programme principal, solution à ce problème, en le décomposant en modules,
- 2) l'algorithme de chaque module.

Exercice 1 :

- 1- Compléter le tableau ci-dessous par la valeur de la variable **c** après exécution de chaque séquence, et ce pour **n=4**.

Séquence	Valeur de la variable c
1	3
2	2
3	3

- 2- Donner le numéro de la séquence qui ne permet pas d'afficher le nombre de diviseurs. Justifier votre réponse. **La séquence 2 : car elle détermine et affiche les diviseurs de N, moins 1**

Exercice 2 :

- 1- Compléter la séquence algorithmique présentée ci-dessous afin de vérifier l'existence d'un nom donné **NOM** dans un tableau **T** non vide.

```

Algorithme recherche
Début
    Écrire ("Donner le nom à chercher : ")
    Lire ( NOM)
    Existe ← Faux
    i ← 0
    Répéter
        Si (T[i] = NOM) alors
            Existe ← Vrai
        Sinon
            i ← i+1.
    Finsi
    Jusqu'à (i=N) ou (Existe)
    Si (Existe) alors
        Écrire ("Le nom recherché existe dans ce tableau.")
    Sinon
        Écrire ("Le nom recherché n'existe pas dans ce tableau.")
    Finsi
Fin
    
```

- 2- Maintenant, le médecin veut chercher les numéros des fiches de ses patients ayant le même nom. Modifier la séquence algorithmique présentée ci-dessus afin d'afficher ces numéros.

```

Algorithme recherche
Début
    Ecrire ("entrer le nom à chercher :");
    Lire(NOM) ;
    Pour i de 0 à N-1 Faire
        Si T[i] = NOM alors
            Ecrire ("Fiche n° : ", i)
        Fin si
    Fin pour
Fin
    
```

Exercice 3 :

Fonction Covid (X, N : entier) : réel

Début

```
f ← 1          /* représente i !
e ← 1          /* valeur initiale de e pour i=0
Px ← 1         /* puissance de X pour i=0
Pour i de 1 à N Faire
    f ← f * i
    Px ← Px * X
    e ← e + Px / f
FinPour
Retourner e
```

Fin

Exercice 4 :

Algorithme du programme principal

Algorithme TRADUCTION

Début

```
Saisir (N, FR, ENG)
SaisirPH (PHF)
Traduire (N, FR, ENG, PHF, PHG, NbErreur)
Afficher (PHG , NbErreur)
```

Fin

TDOG :

Objet	Type/Nature
Nmax	Constante = 100
N, NbErreur : Nombre de mots non traduits	Entier
FR, ENG	TAB
PHF : phrase en français, PHG : phrase en anglais	Chaine de caractère
Saisir, SaisiePH, Traduire, Afficher	Procédures

Tableau de déclaration de nouveaux types (TDNT)

Types

TAB = Tableau de Nmax Chaines de caractères

Procedure Saisir (@N: entier, @FR: TAB, @ENG: TAB)

Début

Répéter

Ecrire ("Donnez le nombre de mots du dictionnaire : ") ;

Lire (N)

Jusqu'à (2 <= N <= 100)

Pour i de 0 à N-1 faire

Répéter

Ecrire ("Donnez un mot en français : ") ;

Lire (FR[i])

Jusqu'à (0 < long (FR[i]) <= 15) et (alpha(FR[i]))

Répéter

Ecrire ("Donnez la traduction en anglais du ", FR[i], " : ") ;

Lire (ENG[i])

Jusqu'à (0 < long (ENG[i]) <= 15) et (alpha (ENG[i]))

Fin pour

Fin

TDOL

Objet	Type/Nature
I	entier
Alpha	fonction

Fonction alpha (ch : chaîne) : booléen

Début

```

i ← 0
test ← Vrai
Tantque (i < long (ch)) et (test = Vrai) faire
    test ← Majus(ch[i]) ∈ ["A".."Z"]
    i ← i + 1
FinTantque
Retourner test

```

TDOL	
Objet	Type/Nature
i	Entier
test	Booléen

Fin

Procédure SaisirPH(@PHF:chaîne)

Début

```

Répéter
    Ecrire ("Donnez une phrase en français : ")
    Lire(PHF)
Jusqu'à valide(PHF)

```

Fin

Fonction Valide (ch : chaîne) : booléen

Début

```

i ← 1
test ← Vrai
Tantque (i < long (ch) et (test = Vrai) faire
    test ← (Majus(ch[i]) ∈ ["A".."Z"]) ou (ch[i] = " ")
    i ← i + 1
Fintantque
Retourner test

```

TDOL	
Objet	Type/Nature
i	entier
Test	booléen

Fin

Procédure Traduire (N: entier, FR: Tab, ENG: Tab, PHF: chaine, @PHG: chaine, @NbErreur:entier)

Début

NbErreur ← 0

PHG ← ""

Tantque(PHF ≠ "") Faire

p ← Pos (" ", PHF)

Si (p ≠ -1) alors

mot ← Sous_chaine(PHF, 0, p)

PHF ← Effacer (PHF, 0, p+1)

Sinon

mot ← PHF

PHF ← ""

Finsi

ind ← Chercher (mot, FR, N)

Si ind = N alors

PHG ← PHG + "{" + mot + "}" + " "

NbErreur ← NbErreur + 1

Sinon

PHG ← PHG + ENG [ind] + " "

Finsi

FinTantque

PHG ← sous_chaine(PHG, 0, long(ch)-1)

Fin

Fonction Chercher (mot : Chaine, FR : Tab, N : entier): entier

Début

i ← 0

Tantque (i < N) et (FR[i] ≠ mot) faire

i ← i+1

FinTantque

Retourner i

Fin

Procédure Afficher (PHG : chaine, NbErreur :entier)

Début

Si (NbErreur = 0) alors

Ecrire (PHG, " : Traduction totale")

Sinon

Ecrire (PHG, " : Traduction partielle", NbErreur, "Mot(s) non traduit(s) ")

Fin si

Fin

TDOL	
Objet	Type/Nature
p, ind	Entier
mot	Chaine
chercher	fonction

TDOL	
Objet	Type/Nature
i	Entier

Exercice 1 :

Dans un contexte algorithmique et pour chacune des propositions ci-dessous, cochez, en mettant une croix (X) dans la case qui correspond à l'unique bonne réponse.

- 1) Quelle est la valeur de Z après l'instruction suivante :

$Z \leftarrow 10 / 5 + 10 \text{ div } 5 + 10 \bmod 5$;

<input type="checkbox"/>	Z=4
<input type="checkbox"/>	Z=2
<input type="checkbox"/>	Z=3

- 2) Quelle est l'instruction erronée parmi les suivantes ?

<input type="checkbox"/>	$X \leftarrow X + 2$
<input type="checkbox"/>	$X \leftarrow X$
<input type="checkbox"/>	$2 \leftarrow X$

- 3) quelle est la valeur finale de A et B après exécution de la séquence suivante :

$A \leftarrow 50$

$B \leftarrow 30$

$A \leftarrow B$

$B \leftarrow A$

<input type="checkbox"/>	A=30 et B=30
<input type="checkbox"/>	A=30 et B=40
<input type="checkbox"/>	A=40 et B=30

- 4) A quelle condition la boucle suivante s'arrête-t-elle ?

Répéter

..

Jusqu'à (x=b) Ou Non(x=0)

<input type="checkbox"/>	x différent de b ou égale à zéro
<input type="checkbox"/>	x égale à b ou différent de zéro
<input type="checkbox"/>	x égale à b et différent de zéro

- 5) Laquelle des instructions suivantes est fausse ?

<input type="checkbox"/>	Lire (5)
<input type="checkbox"/>	Ecrire(5)
<input type="checkbox"/>	Ecrire ("5")

Exercice 2 :

Un nombre heureux est un nombre entier qui, lorsqu'on ajoute les carrés de chacun de ses chiffres, puis les carrés des chiffres de ce résultat et ainsi de suite jusqu'à l'obtention d'un nombre à un seul chiffre égal à 1 (un).

Exemple :

N=7 est heureux, puisque :

- $7^2=49$
- $4^2+9^2=97$
- $9^2+7^2=130$
- $1^2+3^2+0^2=10$
- $1^2+0^2=1$

On est arrivé à un nombre d'un seul chiffre qui est égal à 1, donc N=7 est heureux

Entrez un nombre : 8

- $8^2=64$
- $6^2+4^2=52$
- $5^2+2^2=29$
- $2^2+9^2=85$
- $8^2+5^2=89$
- $8^2+9^2=145$
- $1^2+4^2+5^2=42$
- $4^2+2^2=20$
- $2^2+0^2=4$

On est arrivé à un nombre d'un seul chiffre (4) (<10 et $\neq 1$), donc N=8 n'est pas heureux.

(Idem pour tous les nombres non heureux)

Ecrire un algorithme d'une fonction qui prend en paramètres un entier n et retourne si n est un nombre heureux ou non.

Exercice 3 :

Le code ci-dessous est censé effectuer un tri par ordre alphabétique décroissant d'un tableau L de 5 éléments mais le résultat n'est pas celui attendu.

Initialement L contient dans l'ordre les chaînes suivantes : "un", "deux", "trois", "quatre", "cinq"

Pour i de 0 à 4 faire

mi ← i

Pour j de i à 4 faire

Si $L[mi] < L[j]$ alors mi ← j

Fin SI

e ← L [i]

L [mi] ← L [i]

L[i] ← e

Fin Pour

Fin Pour

1- Donner le contenu du tableau L après exécution du code ci-dessus.

.....
.....

2- Corriger le code précédent de sorte que dans le tableau T on aura le contenu suivant :

"un", "trois", "quatre", "deux", "cinq"

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

Exercice 4 :

A l'occasion de son vingtième anniversaire et pour encourager ses clients, une société de télécommunication, dont les indicatifs sont 65 et 66, accorde un bonus de recharge pour les clients gagnants. Un client est dit gagnant si le montant de ses communications est supérieur ou égal à la consommation moyenne de N clients de la société.

Pour automatiser cette tâche, on se propose d'écrire un programme qui permet de :

1) remplir 2 tableaux T et C respectivement, par les numéros et les consommations téléphoniques des N clients, avec $10 \leq N \leq 100$, sachant que :

- a. chaque numéro est unique et est formé de huit chiffres commençant obligatoirement par 65 ou 66,
- b. pour chaque numéro de téléphone T[i], on fait correspondre un montant de consommation C[i] exprimé en dinars tunisiens,

2) afficher les numéros de téléphone gagnants et les bonus de recharge, obtenus en multipliant par 20, la somme des six derniers chiffres de chaque numéro gagnant.

Exemple :

Pour N=7 et les tableaux T et C suivants

T	66456789	66127721	65980765	66123321	66285585	66285582	65347743
---	----------	----------	----------	----------	----------	----------	----------

C	200	20	300	250	45	99	180
---	-----	----	-----	-----	----	----	-----

Le programme affiche :

Le montant moyen des consommations est : 156.285

Le numéro 66456789 a gagné 780 DT

Le numéro 65980765 a gagné 700 DT

Le numéro 66123321 a gagné 240 DT

Le numéro 65347743 a gagné 560 DT

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 5 :

Le responsable du service informatique dans une entreprise veut sécuriser l'accès aux données confidentielles mises en réseau. Pour cela, il veut affecter un nom d'utilisateur « **LOGIN** » et un mot de passe « **PASSWORD** » à un nombre d'utilisateurs du réseau.

Le LOGIN et le PASSWORD d'un utilisateur sont générés à partir du numéro de sa carte d'identité nationale **NCIN** selon les étapes suivantes :

➤ **Etape1** : Le LOGIN sera composé du mot "CIN" suivi par les 8 chiffres représentant le numéro carte d'identité nationale de l'utilisateur. Les **LOGIN** doivent être distincts d'un utilisateur à un autre.

Exemple :

Si NCIN="01987312" alors le LOGIN correspondant sera "CIN01987312"

➤ **Etape2** : Le PASSWORD sera obtenu en remplaçant chaque chiffre de NCIN par la lettre alphabétique majuscule qui lui correspond selon le principe suivant :

Chiffre	0	1	2	3	4	5	6	7	8	9
Lettre correspondante	"A"	"B"	"C"	"D"	"E"	"F"	"G"	"H"	"I"	"J"

Exemple : Si NCIN="01987312" alors PASSWORD="ABJIHDBC"

Pour aider le responsable, on vous demande de réaliser un programme qui permet de :

- Remplir un tableau TL par les LOGIN de **N** utilisateurs selon le principe ci-dessus (avec $4 \leq N \leq 50$).
- Remplir un tableau TP par les **N** PASSWORD correspondants aux LOGIN du 1^{er} tableau.
- Chercher et afficher le **LOGIN** et le **PASSWORD** d'un utilisateur à partir d'un NCIN donné.

Exemple :

Pour N= 4 et le TL suivant :

TL

CIN43679536	CIN05278439	CIN00438432	CIN54932678
-------------	-------------	-------------	-------------

Le contenu de tableau TP sera :

EDGHJFDG	AFCHIEDJ	AAEDIEDC	FEJDCGHI
----------	----------	----------	----------

Et pour le NCIN donné = 00438432 , le programme affichera :

Login : CIN00438432

Mot de passe : AAEDIEDC

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 1 :

Dans un contexte algorithmique et pour chacune des propositions ci-dessous, cochez, en mettant une croix (X) dans la case qui correspond à l'unique bonne réponse.

- 1) Quelle est la valeur de Z après l'instruction suivante :

$Z \leftarrow 10 / 5 + 10 \text{ div } 5 + 10 \bmod 5 ;$

<input checked="" type="checkbox"/>	Z=4
<input type="checkbox"/>	Z=2
<input type="checkbox"/>	Z=3

- 2) Quelle est l'instruction erronée parmi les suivantes ?

<input type="checkbox"/>	$X \leftarrow X + 2$
<input type="checkbox"/>	$X \leftarrow X$
<input checked="" type="checkbox"/>	$2 \leftarrow X$

- 3) quelle est la valeur finale de A et B après exécution de la séquence suivante :

$A \leftarrow 50$

$B \leftarrow 30$

$A \leftarrow B$

$B \leftarrow A$

<input checked="" type="checkbox"/>	A=30 et B=30
<input type="checkbox"/>	A=30 et B=40
<input type="checkbox"/>	A=40 et B=30

- 4) A quelle condition la boucle suivante s'arrête-t-elle ?

Répéter

..

Jusqu'à (x=b) Ou Non(x=0)

<input type="checkbox"/>	x différent de b ou égale à zéro
<input checked="" type="checkbox"/>	x égale à b ou différent de zéro
<input type="checkbox"/>	x égale à b et différent de zéro

- 5) Laquelle des instructions suivantes est fausse ?

<input checked="" type="checkbox"/>	Lire (5)
<input type="checkbox"/>	Ecrire(5)
<input type="checkbox"/>	Ecrire ("5")

Exercice 2 :

Fonction heureux (n : entier) : booléen

Début

ch←convch(n)

test ←faux

limite ← faux

Tantque (test = faux) et (limite = faux) faire

s←0

Pour i de 0 a long(ch)-1 faire

s←s+ valeur(ch[i])*valeur(ch[i])

FinPour

Si s=1 alors

test ←vrai

FinSi

Si s<10 alors

limite ← vrai

FinSi

ch← convch(s)

FinTantque

Retourner test

Fin

TDO	
Objet	Type/Nature
test, limite	Booléen
ch	Chaine
s, i	Entier

Exercice 3 :

3- Donner le contenu du tableau L après exécution du code ci-dessus.

"un", "deux", "deux", "deux", "cinq"

4- Corriger le code précédent de sorte que dans le tableau T on aura le contenu suivant :

"un", "trois", "quatre", "deux", "cinq"

Pour i de 0 à 3 faire

mi← i

Pour j de i+1 à 4 faire

Si L[mi] < L [j] alors

mi← j

Fin SI

Fin Pour

e← L [mi]

L [mi] ← L [i]

L[i] ← e

Fin Pour

Exercice 4 :

Algorithme de programme principal :

Algorithme bonus

Début

Saisir (n)

RemplirT (T, n)

RemplirC (C, n)

Afficher (T, C, n)

Fin

TDOG :

Objet	Type/Nature
N	Entier
T	Tab1
C	Tab2
Saisir , RemplirT, RemplirC , Afficher	Procédures

Procédure Saisir(@n:entier)

Début

Répéter

Ecrire ("Donnez le nombre de clients : ") ;

Lire (n)

Jusqu'à (10 <=n<=100)

Fin

Procédure RemplirT(@T:Tab1, n:entier)

Début

Pour i de 0 à n-1 faire

Répéter

Ecrire ("Donnez le numéro de téléphone de client n° ", i+1, " : ");

Lire(T[i])

Jusqu'à (Estnum(T[i])=vrai) et (long(T[i])=8) et (sous_chaine(T[i], 0, 2) ∈ ["65", " 66"]) et (Existe(T[i], T, i)=faux)

Fin pour

Fin

Fonction Existe (ch :chaine, T :Tab1, i :entier) :booléen

Début

test←faux

j←0

Tantque (test=faux) et (j<i) faire

Si T[j]=ch alors

test ←vrai

Sinon

j←j+1

FinSi

FinTantque

Retourner test

Fin

Tableau de déclaration de nouveaux types (TDNT)	
Types	
Tab1= Tableau de 100 Chaines de caractères	
Tab2= Tableau de 100 réels	

TDOL	
Objet	Type/Nature
i	entier
Existe	fonction

TDOL	
Objet	Type/Nature
j	Entier
test	Booléen

Procédure RemplirC(@C:Tab2, n:entier)

Début

Pour i de 0 à n-1 faire

Répéter

Ecrire ("Donnez la consommation téléphonique de client n° ", i+1, " : ");

Lire(C[i])

Jusqu'à (C[i] ≥ 0)

Fin pour

Fin

TDOL	
Objet	Type/Nature
i	entier

Procédure Afficher(T :Tab1, C :Tab2, n :entier)

Début

m ← Moyenne(C, n)

Écrire ("Le montant moyen des consommations est :", m)

Pour i de 0 à n-1 faire

Si C[i] ≥ m alors

Ecrire ("Le numéro ", T[i], "a gagné :", Bonus(T[i]), "DT")

FinSi

FinPour

Fin

TDO	
Objet	Type/Nature
i	Entier
Moyenne	Fonction
Bonus	Fonction

Fonction Moyenne(C :Tab2, n :entier) :réel

Début

s ← 0

Pour i de 0 à n-1 faire

s ← s + C[i]

FinPour

Retourner s/n

Fin

TDOL	
Objet	Type/Nature
i	Entier
s	Réel

Fonction Bonus (ch: chaîne): entier

Début

s ← 0

Pour i de 2 à long(ch)-1 faire

s ← s + valeur(ch[i])

FinPour

Retourner s*20

Fin

TDOL	
Objet	Type/Nature
i	Entier
s	Entier

Exercice 5 :

Algorithme de programme principal :

Algorithme

Début

Saisir (N)

Remplir (TL, TP, N)

SaisirCarte(NCIN)

Afficher (NCIN, TL, TP, N)

Fin

TDOG :

Objet	Type/Nature
N	Entier
NCIN	Chaine de caractères
TL, TP	Tab
Saisir , Remplir ,SaisirCarte, Afficher	Procédures

Procédure Saisir(@N:entier)

Début

Répéter

Ecrire ("Donnez le nombre d'utilisateurs: ") ;

Lire (N)

Jusqu'à (4 <=N<=50)

Fin

Procédure Remplir(@TL, @TP :Tab, N:entier)

Début

Pour i de 0 à N-1 faire

Répéter

Ecrire ("Donnez le NCIN de l'utilisateur n° ", i+1, " : ");

Lire(num)

Jusqu'à (Estnum(num)=vrai) et (long(num)=8) et (Existe(num, TL, i)=faux)

TL[i]←"CIN"+num

pw←""

Pour j de 0 à long(num)-1 faire

pw←pw+ chr(ord("A")+valeur(num[j]))

FinPour

TP[i]←pw

Fin pour

Fin

Objet	Type/Nature
i	Entier
num, pw	Chaine
Existe	fonction

Fonction Existe (num :chaine, TL :Tab, i :entier) :booléen

Début

```
test←faux
j←0
Tantque (test=faux) et (j<i) faire
    Si sous_chaine(TL[j], 3, 11)=num alors
        test ←vrai
    Sinon
        j←j+1
    FinSi
FinTantque
Retourner test
```

Objet	Type/Nature
i	Entier
test	Booléen

Fin

Procédure SaisirCarte(@NCIN:entier)

Début

```
Répéter
    Ecrire ("Donnez le NCIN :");
    Lire(NCIN)
Jusqu'à (Estnum(NCIN)=vrai) et (long(NCIN)=8)
```

Fin

Procédure Afficher (NCIN : chaine, TL, TP :Tab, N :entier)

Début

```
p← Chercher (NCIN, TL, N)
Si p = N alors
    Ecrire ("Vérifier le NCIN")
Sinon
    Ecrire ("Login :", TL[p], "Mot de passe :", TP[p])
FinSi
```

Objet	Type/Nature
p	Entier
Chercher	Fonction

Fin

Fonction Chercher (NCIN : chaine, TL :Tab, N :entier) :entier

Début

```
i←0
Tantque (i<N) et (sous_chaine(TL[i], 3, 11)≠ NCIN) faire
    i ←i+1
FinTantque
Retourner i
```

TDOL	
Objet	Type/Nature
i	Entier

Fin

Exercice 1 :

Une chaîne ch est distincte si la chaîne est composée par des caractères distincts (différents).

Exemple :

- Ch="BacSvt" est dite distincte, car elle est formée par des caractères différents.

- Ch="BacMath" est dite non distincte, car elle est formée par des caractères qui ne sont pas distincts (lettre "a").

On vous propose les trois séquences d'instructions algorithmiques suivantes :

Séquence1	Séquence2	Séquence3
Lire (ch) Pour i de 0 à long(ch)-1 faire Si pos(ch[i],ch)=i alors ok←Vrai Sinon ok←Faux FinSi FinPour	Lire (ch) Répéter aux←ch[0] ch←effacer (ch,0,1) ok←Pos(aux, ch)=-1 jusqu'à (ok=faux) ou (long(ch)=1)	Lire (ch) i←0 Répéter ok← pos (ch[i], ch)=i i←i + 1 jusqu'à (ok=faux) ou (i=long (ch))

1) Compléter le tableau ci-dessous par la valeur de la variable ok après exécution de chaque séquence, et ce pour **ch="Sweet"**.

Séquence	Valeur de la variable ok
1	
2	
3	

2) Donner le numéro de la ou (les) séquence(s) qui ne permet pas de vérifier si la chaîne est distincte. Justifier votre réponse.

.....

.....

.....

Exercice2 :

Soit l'algorithme inconnu suivant :

Algorithme inconnu

Début

```
Lire (n)
ch ← ""
Pour i de 0 à n-1 faire
    Répéter
        T[i] ← Aléa(1000, 9999)
        Ch1 ← convch (T[i])
    Jusqu'à (Pos (ch1, ch) = -1)
    ch ← ch + ch1 + "#"
Finpour
```

Fin

Travail demandé :

Soit le tableau T suivant :

3245	1111	8764	1111	6511	2344	9376	5678	2344
------	------	------	------	------	------	------	------	------

- 1) Peut-on admettre le tableau T comme résultat possible de l'algorithme inconnu ? Justifier votre réponse.

.....
.....

- 2) Dédurre le rôle de l'algorithme inconnu.

.....
.....

- 3) Compléter les tableaux des déclarations des objets et des nouveaux types de l'algorithme ci-dessus :

T contient 30 éléments au maximum.

TDNT
Type
Tab =

TDO	
Objet	T/N
T	Tab

- 4) Ecrire l'algorithme d'un module qui permet d'afficher les éléments **automorphes** du tableau T. Un entier N est dit automorphe si son carré N^2 se termine par N.

Exemples : 25, 376, 9376 sont des nombres automorphes car leurs carrés sont respectivement 625, 141376 et 87909376. Mais 36 ne l'est pas car $36^2 = 1296$ ne se termine pas par 36.

Exercice 3 :

Dans le cadre de la campagne de vaccination contre le COVID-19, toute personne inscrit à cette campagne, a la possibilité de consulter l'état de sa vaccination en composant ***2021*code d'inscription#** sur son téléphone portable.

Un code **d'inscription** est valide s'il est composé de **9 chiffres** et **ne commence pas par 0**.

Exemple : ***2021*156390321#**

Le code d'inscription **C = 156390321** est valide.

Une fois le code **C** est valide, on passe à la vérification de son existence dans le tableau **CODES** puis la consultation de son état dans le tableau **VACCINATION** sachant que :

- **CODES** est un tableau contenant **n** codes d'inscription (avec $1 \leq n \leq 1000$).
- **VACCINATION** est un tableau d'entiers de même taille que le tableau **CODES**. Chacune des cases contient l'état de vaccination de la personne dont le code d'inscription correspond à la case du même indice du tableau **CODES** :
 - **0** : la personne est inscrit mais n'est pas encore vacciné.
 - **1** : la personne a reçu la première dose de vaccin.
 - **2** : la personne a reçu les deux doses de vaccin.

On se propose d'écrire un programme permettant de :

- Remplir le tableau **CODES** par **n codes d'inscription (chaines de caractères)** valides et distincts.
- Remplir le tableau **VACCINATION** par **n états (0, 1, 2)**.
- Saisir un essai de consultation sous la forme ***2021*code d'inscription#**.
- Afficher l'un des messages suivants :
 - **"Erreur code"** : Si le code d'inscription n'est pas valide.
 - **"Vérifiez votre code d'inscription ou bien il se peut que vous n'êtes pas encore inscrit"** : Si le code est valide mais n'existe pas dans le tableau **CODES**.
 - **"Vous n'êtes pas encore vacciné"** : Si le code existe et l'état de la vaccination est **0**.
 - **"Vous avez reçu la première dose de vaccin"** : Si le code existe et l'état de la vaccination est **1**.
 - **"Vous avez reçu les deux doses de vaccin"** : Si le code existe et l'état de la vaccination est **2**.

Exemple :

CODES	138656701	215566827	877001006	231564796	556008991
VACCINATION	2	1	0	1	0

Essai de consultation : ***2021*231564796#**

Le programme affiche : **"Vous avez reçu la première dose de vaccin"**

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 1

1) Compléter le tableau ci-dessous par la valeur de la variable ok après exécution de chaque séquence, et ce pour **ch="Sweet"**.

Séquence	Valeur de la variable ok
1	vrai
2	faux
3	faux

2) Donner le numéro de la ou (les) séquence(s) qui ne permet pas de vérifier si la chaîne est distincte. Justifier votre réponse.

La séquence n°1 : car elle permet de vérifier si le dernier caractère est différent des autres caractères de la chaîne.

Exercice2 :

1- Peut-on admettre le tableau T comme résultat possible de l'algorithme inconnu ? Justifier votre réponse.

Non : Car les éléments du tableau ne sont pas distincts. 1111 et 2344 se répètent deux fois dans le tableau.

2- Déduire le rôle de l'algorithme inconnu.

Remplir le tableau T par n nombres aléatoires distincts composés de quatre chiffres.

3- Compléter les tableaux des déclarations des objets et des nouveaux types de l'algorithme ci-dessus :
T contient 30 éléments au maximum.

TDNT
Type
Tab = tableau de 30 entiers

TDO	
Objet	T/N
T	Tab
Ch	chaîne
ch1	chaîne
N	entier
I	entier

4- Ecrire l'algorithme d'un module qui permet d'afficher les éléments **automorphes** du tableau T. Un entier N est dit automorphe si son carré N^2 se termine par N.

Procédure afficher(T :tab, n :entier)

Début

Pour i de 0 à n-1 faire

s ← T[i]*T[i]

ch ← convch(s)

Effacer (ch, 0, long(ch)-4)

Si valeur(ch)=T[i] alors

Ecrire(T[i])

FinSi

FinPour

Fin

Exercice 3 :

Algorithme de programme principal :

Algorithme Covid

Début

Saisir (n)

Remplir (CODES, VACCINATION, n)

SaisirEssai(essai)

Afficher (CODES, VACCINATION, n, essai)

Fin

TDOG :

Tableau de déclaration de nouveaux types (TDNT)
--

Types

tab1= Tableau de 1000 Chaines de caractères tab2= Tableau de 1000 entiers
--

Objet	Type/Nature
n	Entier
CODES	tab1
VACCINATION	tab2
Essai	Chaine
Saisir, Remplir, SaisirEssai, Afficher	Procédures

Procédure Saisir(@n:entier)

Début

Répéter

Ecrire ("Donnez le nombre de personnes: ") ;

Lire (n)

Jusqu'à (1 <=n<=1000)

Fin

Procédure Remplir(@CODES:tab1, @VACCINATION:tab2, n:entier)

Début

Pour i de 0 à n-1 faire

Répéter

Ecrire ("Donnez le code d'inscription de personne n° ", i+1, " : ")

Lire(CODES[i])

Jusqu'à (Estnum(CODES [i])) et (long(CODES[i])=9) et (CODES [0]≠"0") et (Existe(CODES [i], CODES, i)=faux)

Répéter

Ecrire ("Donnez l'état de vaccination de personne n° ", i+1, " : ")

Lire (VACCINATION[i])

Jusqu'à VACCINATION[i] ∈ [0, 1, 2]

Fin pour

Fin

TDOL	
Objet	Type/Nature
i	Entier
Existe	Fonction

Fonction Existe (ch :chaîne, CODES :tab1, i :entier) :booléen

Début

test←faux

j←0

Tantque (test=faux) et (j<i) faire

Si CODES[j]=ch alors

test ←vrai

Sinon

j←j+1

FinSi

FinTantque

Retourner test

Fin

Procédure SaisirEssai(@essai: chaîne)

Début

Répéter

Ecrire ("Donnez un essai :");

Lire(essai)

Jusqu'à (sous_chaine(essai,0, 6)="*2021*") et (essai[long(essai)-1]="#")

Fin

Procédure Afficher (CODES : tab1, VACCINATION: tab2, n : entier, essai : chaîne)

Début

code ←sous_chaine (essai, 6, long (essai)-1)

Si NON (Estnum(code) et long(code)=9 et code[0]≠ "0") alors

Ecrire ("Erreur code")

Sinon

p←Chercher (code, CODES, n)

Si p = N alors

Ecrire ("Vérifiez votre code d'inscription ou bien il se peut que vous n'êtes pas encore inscrit")

Sinon Si VACCINATION[p] =0 alors

Ecrire ("Vous n'êtes pas encore vacciné")

Sinon Si VACCINATION[p] =1 alors

Ecrire ("Vous avez reçu la première dose de vaccin")

Sinon

Ecrire ("Vous avez reçu les deux doses de vaccin")

FinSi

FinSi

Fin

.....

Fonction Chercher (code : Chaîne, CODES : tab1, n : entier): entier

Début

i←0

Tantque (i<N) et (CODES[i] ≠ code) faire

i←i+1

FinTantque

Retourner i

Fin

TDOL	
Objet	Type/Nature
j	Entier
test	Booléen

TDOL	
Objet	Type/Nature
code	chaîne
p	entier
Chercher	fonction

TDOL	
Objet	Type/Nature
i	entier

Exercice 1 :

Mettre une croix ☒ dans les propositions correctes.

a- Soit l'instruction $C \leftarrow \text{SOUS-CHAINE ("informatique", 0, 4)}$.

- ☐ Elle permet d'affecter à C la valeur "infor"
- ☐ La variable C doit être de type caractère
- ☐ La variable C doit être de type chaîne de caractère

b- $X \leftarrow \text{Arrondi}(2.51) - 1 * 65 \bmod 2*3$

- ☐ X=10
- ☐ X=0
- ☐ X=6
- ☐ X=-2

c- Soit l'instruction $A \leftarrow \text{Alea}(0, 10) - 5$ permet d'affecter à la variable A une valeur aléatoire de l'intervalle :

- ☐ [0..5]
- ☐ [5..10]
- ☐ [-5..5]
- ☐ [-5..10]

d- Soit la séquence d'instructions ci-dessous. Combien de fois s'exécute le bloc d'instructions **Traitement** sachant qu'il ne modifie pas le contenu de la variable **a** ?

$a \leftarrow 1, b \leftarrow 0$

Répéter

Traitement

$b \leftarrow a + 4$

Jusqu'à $a = 10$

- ☐ 1 fois
- ☐ 3 fois
- ☐ 5 fois
- ☐ A l'infini

e- Soit le bloc d'instructions suivant :

Pour K de 0 à 4 Faire

Instruction_Cle

Fin Pour

Déterminer à chaque fois le rôle de ce bloc d'instructions lorsqu'on remplace "**Instruction_Cle**" par une instruction donnée. (**T est un tableau de 5 entiers positifs**)

- **Instruction_Cle = $T[K] \leftarrow T[k]*2$**

- ☐ Diviser par deux les éléments de T.
- ☐ Calculer le produit des éléments de T.
- ☐ Multiplier par deux les éléments de T.
- ☐ Calculer la somme des éléments de T.

- **Instruction_Cle = Si ((K MOD 2) = 0) Alors $T[k] \leftarrow T[k] \text{ DIV } 2$ Fin Si**

- ☐ Multiplier par deux les éléments d'indice pair de T.
- ☐ Diviser par deux les éléments d'indice impair de T.
- ☐ Multiplier par deux les éléments d'indice impair de T.
- ☐ Diviser par deux les éléments d'indice pair de T.

- **Instruction_Cle = Si ((K MOD 2) = 0) Alors $T[k] \leftarrow T[k] + 2$ Fin Si**

- ☐ Ajouter deux aux éléments d'indice pair de T.
- ☐ Ajouter deux aux éléments impairs de T.
- ☐ Ajouter deux aux éléments d'indice impair de T.
- ☐ Ne changer pas les éléments impairs du T.

Exercice 2 :

Soit le bloc algorithmique suivant :

```
Pour i de 0 à N-1 faire
    ch ← T[i]
    Répéter
        Si Majus(ch[0]) dans ["A".."Z"] alors
            Effacer(ch,0,1)
        Fin si
    Jusqu'à (Non (Majus(ch[0] dans ["A".."Z"])) ou (ch=""))
    Si ch="" Alors
        Ecrire ("OUI")
    Sinon
        Ecrire ("NON")
    Fin Si
Fin pour
```

Questions :

- 1) a- Compléter les tableaux de déclaration des nouveaux types et des objets. T est un tableau qui contient au maximum 30 éléments.

TDNT
Type
TAB =

TDO	
Objets	Type / Nature
T	TAB

- b- Compléter le tableau suivant, soit par les résultats générés par le bloc algorithmique relatif aux éléments de tableau T, soit par des éléments de T permettant d'aboutir aux résultats mentionnés dans le tableau.

Tableau T	"Baccalauréat"	"Examen"	"Informatique"	"Bac2021"
Résultats	NON	OUI

- c- Déduire le rôle de ce bloc algorithmique.

.....

2- On désire modifier l'affichage dans le bloc algorithmique ci-dessus de telle sorte que si la condition est vérifiée, on affiche l'élément de tableau T, sinon on élimine tous les caractères non alphabétiques de cet élément puis on l'affiche.

Ancien affichage	Nouvel affichage
<i>Si ch="" Alors</i> <i>Ecrire ("OUI")</i> <i>Sinon</i> <i>Ecrire ("NON")</i> <i>Fin Si</i>	

Exercice 4 :

L'examen d'un patient par son médecin commence par la collecte des symptômes à travers un ensemble de questions posées. Chaque symptôme sert à l'identification des maladies possibles et aide le médecin à prendre sa décision.

Pour l'aider à déterminer et à afficher les maladies correspondant à un symptôme donné, on se propose d'écrire un programme qui permet de :

- remplir un tableau M, par les noms de N maladies ($2 < N \leq 100$) où le nom d'une maladie est une chaîne formée par des caractères alphabétiques majuscules et peut contenir des espaces.
- remplir un tableau S par N chaînes, représentant chacune, les principaux symptômes de la maladie M[i] séparés par des "#". Le dernier symptôme ne sera pas suivi du caractère "#". Un symptôme est une chaîne formée par des caractères alphabétiques majuscules et peut contenir des espaces.
- afficher les maladies probables, pour un symptôme donné.

Exemple :

Pour $n = 5$ et les deux tableaux M et S suivants :

M	INSUFFISANCE	SPONDYLITE	TUBERCULOSE	ANEMIE	RENALE NEURALGIE
S	FATIGUE#PERTE APPETIT#MAUX DE TETE# PARALYSIE# DOULEUR	DOULEUR VERTEBRALE# DOULEUR THORACIQUE# PERTE APPETIT# INFLAMMATION OCCULAIRE	SUEUR#TOUX# DOULEUR POITRINE #DOULEUR VERTEBRALE #PERTE APPETIT	PERTE APPETIT# FATIGUE#MAUX DE TETE	PARALYSIE# DOULEUR

Pour le symptôme : "**DOULEUR VERTEBRALE**"

Les maladies probables sont : **SPONDYLITE et TUBERCULOSE**

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 1 :

Mettre une croix ☒ dans les propositions correctes.

b- Soit l'instruction $C \leftarrow \text{SOUS-CHAINE} ("informatique", 0, 4)$.

- | | |
|-------------------------------------|---|
| <input type="checkbox"/> | Elle permet d'affecter à C la valeur "infor" |
| <input type="checkbox"/> | La variable C doit être de type caractère |
| <input checked="" type="checkbox"/> | La variable C doit être de type chaîne de caractère |

f- $X \leftarrow \text{Arrondi}(2.51) - 1 * 65 \bmod 2*3$

- | | |
|-------------------------------------|------|
| <input type="checkbox"/> | X=10 |
| <input checked="" type="checkbox"/> | X=0 |
| <input type="checkbox"/> | X=6 |
| <input type="checkbox"/> | X=-2 |

g- Soit l'instruction $A \leftarrow \text{Alea} (0, 10) - 5$ permet d'affecter à la variable A une valeur aléatoire de l'intervalle :

- | | |
|-------------------------------------|----------|
| <input type="checkbox"/> | [0..5] |
| <input type="checkbox"/> | [5..10] |
| <input checked="" type="checkbox"/> | [-5..5] |
| <input type="checkbox"/> | [-5..10] |

h- Soit la séquence d'instructions ci-dessous. Combien de fois s'exécute le bloc d'instructions **Traitement** sachant qu'il ne modifie pas le contenu de la variable **a** ?

$a \leftarrow 1, b \leftarrow 0$
Répéter
Traitement
 $b \leftarrow a + 4$
Jusqu'à $a = 10$

- | | |
|-------------------------------------|------------|
| <input type="checkbox"/> | 1 fois |
| <input type="checkbox"/> | 3 fois |
| <input type="checkbox"/> | 5 fois |
| <input checked="" type="checkbox"/> | A l'infini |

i- Soit le bloc d'instructions suivant :

Pour K de 0 à 4 Faire
Instruction_Cle
Fin Pour

Déterminer à chaque fois le rôle de ce bloc d'instructions lorsqu'on remplace "Instruction_Cle" par une instruction donnée. (T est un tableau de 5 entiers positifs)

- **Instruction_Cle = T[K] ← T[k]*2**

<input type="checkbox"/>	Diviser par deux les éléments de T.
<input type="checkbox"/>	Calculer le produit des éléments de T.
<input checked="" type="checkbox"/>	Multiplier par deux les éléments de T.
<input type="checkbox"/>	Calculer la somme des éléments de T.

- **Instruction_Cle = Si ((K MOD 2) = 0) Alors T[k] ← T[k] DIV 2 Fin Si**

<input type="checkbox"/>	Multiplier par deux les éléments d'indice pair de T.
<input type="checkbox"/>	Diviser par deux les éléments d'indice impair de T.
<input type="checkbox"/>	Multiplier par deux les éléments d'indice impair de T.
<input checked="" type="checkbox"/>	Diviser par deux les éléments d'indice pair de T.

- **Instruction_Cle = Si ((K MOD 2) = 0) Alors T[k] ← T[k] + 2 Fin Si**

<input checked="" type="checkbox"/>	Ajouter deux aux éléments d'indice pair de T.
<input type="checkbox"/>	Ajouter deux aux éléments impairs de T.
<input type="checkbox"/>	Ajouter deux aux éléments d'indice impair de T.
<input checked="" type="checkbox"/>	Ne changer pas les éléments impairs du T.

Exercice 2 :

Questions :

- 1) a- Compléter les tableaux de déclaration des nouveaux types et des objets. T est un tableau qui contient au maximum 30 éléments.

TDNT
Type
TAB = Tableau de 30 chaînes de caractères

TDO	
Objets	Type / Nature
T	TAB
N	entier
i	entier
ch	chaîne

- b- Compléter le tableau suivant, soit par les résultats générés par le bloc algorithmique relatif aux éléments de tableau T, soit par des éléments de T permettant d'aboutir aux résultats mentionnés dans le tableau.

Tableau T	"Baccalauréat"	"Examen"	"Net@"	"Informatique"	"Bac2022"	"Graphique"
Résultats	OUI	OUI	NON	OUI	NON	OUI

c- D  duire le r  le de ce bloc algorithmique.

Permet de d  terminer et afficher si chaque   l  ment du tableau T est une chaine de caract  res alphab  tiques ou non.

2- On d  sire modifier l'affichage dans le bloc algorithmique ci-dessus de telle sorte que si la condition est v  rifi  e, on affiche l'  l  ment de tableau T, sinon on   limine tous les caract  res non alphab  tiques de cet   l  ment puis on l'affiche.

Ancien affichage	Nouvel affichage
<pre> Si ch="" Alors Ecrire ("OUI") Sinon Ecrire ("NON") Fin Si </pre>	<pre> Si ch="" Alors Ecrire (T[i]) Sinon ch1←T[i] j←0 Tantque j<long (ch1) faire Si NON (Majus(ch1[j]) ��["A".."Z"]) alors Effacer (ch1, j, j+1) Sinon j←j+1 FinSi FinTantque T[i]←ch1 Ecrire (T[i]) Fin Si </pre>

Exercice 4 :

Algorithme du programme principal

Algorithme EXAMEN

D  but

Saisir (N)

Remplir (M, S, N)

Ecrire ("Donner un sympt  me :")

Lire(Sym)

Afficher (M, S, N, Sym)

Fin

Tableau de d��claration de nouveaux types (TDNT)
Types
TAB = Tableau de 100 Chaines de caract��res

TDOG :

Objet	Type/Nature
N	Entier
M, S	TAB
Sym	Chaine de caract��re
Saisir , Remplir, Afficher	Proc��dures

Proc  dure Saisir(@N:entier)

D  but

R  p  ter

Ecrire ("Donnez le nombre de maladies: ")

Lire (N)

Jusqu'   (2 <=N<=100)

Fin

Procédure Remplir(@M:TAB, @)S:TAB, N:entier)

Début

Pour i de 0 à N-1 faire

Répéter

Ecrire ("Donnez une maladie : ");

Lire(M[i])

Jusqu'à (0<long(M[i])) et (alpha(M[i]))

Répéter

Ecrire ("Donnez les symptômes relatifs au maladie ", M[i], " : ");

Lire(S[i])

Jusqu'à (0<long(S[i])) et (valide(S[i]))

Fin pour

Fin

Fonction alpha (ch : chaîne) : booléen

Debut

i ← 0

test ← Vrai

Tantque (i<long(ch)) et (test=Vrai) faire

test ← ch[i] ∈ ["A".. "Z", " ", " "]

i ← i+1

Fin tantque

Retourner test

Fin

Fonction valide(ch : chaîne) : booléen

Debut

i ← 0

test ← Vrai

Tantque (i<long(ch)) et (test=Vrai) faire

test ← ch[i] ∈ ["A".. "Z", " ", "#"]

i ← i+1

Fin tantque

Retourner (test) et (ch[0] ≠ "#") et (ch[long(ch)-1] ≠ "#")

Fin

Procédure Afficher (M, S :TAB , N :entier, Sym :chaîne)

Début

Pour i de 0 à N-1 faire

Si pos (Sym, S[i]) ≠ -1 alors

Ecrire (M[i])

Fin Si

Fin Pour

Fin

TDOL	
Objet	Type/Nature
i	Entier
alpha	Fonction
valide	Fonction

TDOL	
Objet	Type/Nature
i	Entier
test	Booléen

TDOL	
Objet	Type/Nature
i	entier
test	booléen

TDOL	
Objet	Type/Nature
i	entier

Exercice 1 :

On veut déterminer et afficher le nombre de mots dans une phrase non vide. On suppose que la phrase ne contient pas un espace au début, un espace à la fin et des espaces superflus (un seul espace sépare deux mots).

Exemple :

Pour **ch="Félicitation vous avez gagné"**, le nombre de mots est **4**.

On vous propose les trois séquences d'instructions algorithmiques suivantes :

<i>Séquence 1</i>	<i>Séquence 2</i>	<i>Séquence 3</i>
$n \leftarrow 1$ $p \leftarrow \text{pos}(" ", \text{ch})$ Tantque $p \neq -1$ faire $n \leftarrow n+1$ Effacer (ch, 0, p+1) $p \leftarrow \text{pos}(" ", \text{ch})$ Fin Tantque Ecrire (n)	$n \leftarrow 0$ $p \leftarrow \text{pos}(" ", \text{ch})$ Tantque $p \neq -1$ faire $n \leftarrow n+1$ Effacer (ch, p, p+1) $p \leftarrow \text{pos}(" ", \text{ch})$ Fin Tantque Ecrire (n)	$n \leftarrow 1$ $p \leftarrow \text{pos}(" ", \text{ch})$ Tantque $p \neq -1$ faire $n \leftarrow n+1$ Effacer (ch, p, p+1) $p \leftarrow \text{pos}(" ", \text{ch})$ Fin Tantque Ecrire (n)

3- Compléter le tableau ci-dessous par la valeur de la variable n après exécution de chaque séquence, et ce pour **ch= "Python est un langage de programmation"**.

Séquence	Valeur de la variable n
1	
2	
3	

4- Donner le numéro de la séquence qui ne permet pas d'afficher le nombre de mots. Justifier votre réponse.

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

Exercice 2 :

On veut crypter une chaîne de caractères donnée **ch** en une chaîne résultat **Res** de la manière suivante :

Parcourir la chaîne **ch** de gauche à droite en comptant le nombre d'occurrences successives de chaque caractère de la chaîne **ch**, puis de ranger dans la chaîne **Res**, ce nombre suivi du caractère en question.

Exemple :

Si **ch**="aaaFyBssssssssssaz" alors la chaîne **Res** qui sera affichée est "3a1F1y1B12s1a2z"

Le code ci-dessous est censé effectuer ce cryptage mais le résultat n'est pas celui attendu.

```
Res ← ""
s ← 0
Pour i de 0 à long(ch)-2 faire
    Si ch[i]=ch[i+1] alors
        s ← s+1
    Sinon
        Res ← Res + convch(s)+ch[i]
        s ← 0
FinSi
FinPour
```

- Quelle est la valeur de **Res** pour chacun des cas suivants :

ch= "aaDDdkkkkgGGw" Res=

ch="ttTThhhhpxxxxnNjjjjjj" Res=

- Corriger le code précédent pour avoir le résultat attendu.

La « **multiplication Russe** » est une méthode particulière permettant la multiplication de deux entiers A et B en utilisant seulement la multiplication par 2, la division par 2 et l'addition.

A	B
17	19

8 38

→	17	19
	8	38
	4	76
	2	152
→	1	304

Le résultat est la somme des nombres de la deuxième colonne qui sont en face des nombres impairs de la première colonne.

[illegible]

Exercice 4 :

Une station de radio lance chaque début de semaine un concours hebdomadaire intitulé « Hitparade » pour classer cinq titres de chansons qu'elle propose à ses auditeurs et affiche le résultat du classement le samedi.

Tout au long de la semaine, un responsable de la station reçoit par téléphone les propositions des participants au concours parmi ses auditeurs qui sont appelés à choisir le titre préféré afin d'ajouter à son score 3 points.

Pour obtenir le classement final des cinq chansons suite à la participation d'un nombre donné des auditeurs de la station, on se propose d'écrire un programme qui permet de :

- Remplir un tableau T1 par les cinq titres des chansons, sachant qu'un titre est composé uniquement par des lettres alphabétiques et des espaces.
- Saisir le nombre N de participants avec $5 \leq N \leq 100$.
- Générer un tableau T2 représentant les scores des cinq chansons en ajoutant 3 points au score de chacune si le numéro qui lui correspond a été choisi par un participant. Ce numéro est un chiffre allant de 1 à 5, représentant l'emplacement de la chanson dans le tableau T1.
- Afficher le classement des chansons, comme indiqué dans l'exemple ci-après, en commençant par le titre de la chanson ayant le plus grand score. Il est à noter que les chansons ayant un même score auront un même rang dans le classement.

Exemple :

Pour :

T1

Happy	Sorry	Me Quemo	Rosa	Hello
-------	-------	----------	------	-------

Et un nombre de participants N =11 ayant fait les choix suivants :

Choix du participant n°1 : 5

Choix du participant n°2 : 4

Choix du participant n°3 : 5

Choix du participant n°4 : 1

Choix du participant n°5 : 2

Choix du participant n°6 : 5

Choix du participant n°7 : 1

Choix du participant n°8 : 2

Choix du participant n°9 : 4

Choix du participant n°10 : 1

Choix du participant n°11 : 5

On aura :

T2

9	6	0	6	12
---	---	---	---	----

Le programme affiche :

Le classement est :

Rang 1 : Hello

Rang 2 : Happy

Rang 3 : Sorry, Rosa

Rang 4 : Me Quemo

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 5 :

On se propose de crypter un message composé par des mots séparés par un seul espace et ne contenant aucun signe de ponctuation (, ; : ! ?) en utilisant le principe suivant :

- 1) Placer chaque mot du message initial dans une case d'un tableau T. On suppose que le message est composé de 20 mots au maximum.
- 2) Pour chaque élément du tableau T, ajouter autant de fois le caractère "*" pour que sa longueur sera égale a celle du mot le plus long dans le tableau T.
- 3) Dans un nouveau tableau T1 de taille N1 (N1 longueur du mot le plus long), repartir les lettres du mot se trouvant dans la case T[1] de façon a placer la lettre d'indice i du mot dans la case d'indice i du tableau T1.
- 4) Répartir de la même façon les lettres du mot contenu dans la case T[2] en concaténant a chaque fois la lettre d'indice i avec le contenu de la case i du tableau T1.
- 5) Répartir de la même façon le reste des mots de T dans T1.
- 6) Concaténer les mots obtenus dans T1 en les séparant par un espace pour obtenir le message crypté.

Exemple : Si le message à crypter est : "Bonjour Sami j'ai fini mon travail". Les étapes de cryptage sont :

Etape 1 : Répartir les mots de message dans le tableau T :

T	Bonjour	Sami	J'ai	Fini	mon	travail
----------	---------	------	------	------	-----	---------

Etape 2 : Ajouter le caractère "*" autant de fois pour obtenir des mots dont la longueur de chacun est égale a celle du mot le plus long.

Etant donné que Bonjour est le mot le plus long du message (7 caractères), on obtient le tableau T suivant.

T	Bonjour	Sami***	j'ai***	fini***	mon****	travail
----------	---------	---------	---------	---------	---------	---------

Etape 3 : Répartir les lettres de T[1] dans T1.

T1	B	O	n	j	o	u	r
-----------	---	---	---	---	---	---	---

Etape 4 : Répartir les lettres de T[2] dans T1.

T1	BS	Oa	nm	ji	o*	u*	r*
-----------	----	----	----	----	----	----	----

Etapes suivantes : Répartir le reste des mots de T dans T1.

T1	BSjfmt	oa'ior	nmanna	jiii*v	o****a	u****i	r****l
-----------	--------	--------	--------	--------	--------	--------	--------

Le message crypté sera alors "BSjfmt oa'ior nmanna jiii*v o****a u****i r****l"

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

Exercice 1 :

- 1- Compléter le tableau ci-dessous par la valeur de la variable n après exécution de chaque séquence, et ce pour `ch= "Python est un langage de programmation"`.

Séquence	Valeur de la variable n
1	6
2	5
3	6

- 2- Donner le numéro de la séquence qui ne permet pas d'afficher le nombre de mots. Justifier votre réponse.

Séquence 2 : car elle permet de déterminer et afficher le nombre de mots dans ch, moins 1. En effet le dernier mot ne sera pas pris en compte.

Exercice 2 :

- 1- Quelle est la valeur de **Res** pour chacun des cas suivants :

`ch= "aaDDDkkkkkgGGw"`

Res= "1a2D4k0g1G"

`ch="tttTThhhhpxxxxnNjjjjjjj"`

Res="2t1T3h0p3x0n0N"

- 2- Corriger le code précédent pour avoir le résultat attendu.

Res← ""

s←1

Pour i de 0 à long(ch)-2 faire

Si ch[i]=ch[i+1] alors

s←s+1

Sinon

Res ←Res + convch(s)+ch[i]

s←1

FinSi

FinPour

Res ←Res + convch(s)+ch[long(ch)-1]

Exercice 3 :

Fonction multiplication (A, B :entier) :entier

Debut

s←0

Tantque A≠1 faire

Si A mod 2 ≠0 alors

s←s+B

FinSi

A←A div 2

B←B*2

FinTantque

Retourner s+B

Fin

Exercice 4

1) Algorithme du programme principal :

Algorithme Hitparade

Début

Chanson(T1)

Saisir(N)

Score (T2, N)

Trier (T1, T2)

Afficher (T1, T2)

Fin

TDNT
Nouveau type
Tab1=tableau de 5 chaînes
Tab2=tableau de 5 entiers

TDOG	
Objet	Type/Nature
T1	Tab1
T2	Tab2
N	Entier
Saisir	Procédure
Chanson	Procédure
Score	Procédure
Trier	Procédure
Afficher	Procédure

2) Les algorithmes des modules envisagés:

Procédure Chanson(@T1 :Tab1)

Début

Pour i de 0 à 4 faire

Répéter

Ecrire ("Saisir le titre de la chanson n°", i+1, ",")

Lire (T1 [i])

Jusqu'à (long(T[i])>0) et (alpha (T1[i])=vrai)

Finpour

Fin

TDOL	
Objet	Type/Nature
i	entier
alpha	Fonction

Fonction alpha (ch : chaîne) : booléen

Début

test← vrai

i←0

Tantque (i<long (ch)) et (test=vrai) faire

test←majus(ch[i])∈ ["A".."Z"]

i←i+1

Fin Tantque

Retourner test

Fin

TDOL	
Objet	Type/nature
test	Booléen
i	Entier

Procédure Saisir (@ N:entier)

Début

Répéter

Ecrire ("Saisir le nombre de participants:")

Lire(N)

Jusqu'à N ∈ [5..100]

Fin

Procédure Score(@T2 :Tab2, N:entier)

Début

Pour i de 0 à 4 faire

T2 [i] ← 0

Finpour

Pour i de 1 à N faire

Répéter

Ecrire ("Saisir le numéro de la chanson:")

Lire (num)

Jusqu'à (num ∈ [1..5])

T2[num-1] ← T2[num-1]+3

Finpour

Fin

Procédure Trier(@T1 :Tab1,@T2 :Tab2)

Début

Pour i de 0 à 3 faire

indice ← i

Pour j de i+1 à 4 faire

Si T2[j] > T2[indice] alors

indice ← j

Fin si

Fin pour

Si indice ≠ i alors

Aux1 ← T2[indice]

T2[indice] ← T2[i]

T2[i] ← Aux1

Aux2 ← T1[indice]

T1[indice] ← T1[i]

T1[i] ← Aux2

Finsi

Finpour

Fin

TDOL	
Objet	Type/Nature
i, j, indice, Aux1	Entier
Aux2	Chaine

TDOL	
Objet	Type/nature
i, num	entier

Procédure Afficher (T1:Tab1, T2 :Tab2)

Début

Ecrire ("Le classement est :")

rang ← 1

Ecrire ("Rang", rang, ":", T1[0])

Pour i de 1 à 4 faire

Si (T2[i]=T2[i-1]) alors

Ecrire (" ", T1[i])

Sinon

rang ← rang+1

Ecrire ("Rang",rang,":",T1[i])

Fin si

Fin pour

Fin

TDOL	
Objet	Type/nature
i, rang	entier

Exercice 5 :**Algorithme de programme principal :****Algorithme cryptage**

Début

Saisir (msg)

Ecrire ("Le message crypté est : ", Crypter (msg))

Fin

TDOG		
Objet	T/N	
msg	Chaine	Le message à crypter
Crypter	Fonction	Retourner la chaine cryptée
Saisir	procédure	Saisir le message à crypter

Procédure Saisir (@ msg :chaine)

Début

Répéter

Ecrire ("Saisir le message à crypter : ")

Lire (msg)

Jusqu'à valide (msg)

Fin

Fonction valide (msg:chaine): booléen

Debut

i ← 0

test ← faux

Tantque (i < long(ch)) et (test = faux) faire

test ← msg[i] ∈ [",", ".", ":", ";", "!", "?", " ", ":"]

i ← i + 1

FinTantque

Retourner (pos (" ", msg) = -1) et (test = faux)

Fin

Fonction Crypter (msg: chaine) : chaine

Début

Separer (msg, T, n)

AjouterEtoiles (T, n, n1)

Repartir (T, n, T1, n1)

ch ← ""

Pour i de 0 a n1-1 faire

ch ← ch + T1[i] + " "

Fin pour

ch ← sous-chaine (ch, 0, long(ch) - 1)

retourner ch

Fin

TDNT

Type
Tab = tableau de 20 chaines

Saisir une chaine (msg) composée par des mots séparés par un seul espace et ne contenant aucun signe de ponctuation.

TDOL	
Objet	T/N
i, n1, n	Entier
ch	Chaine
T	Tab
T1	Tab
Repartir	Procédure
Separer	Procédure
AjouterEtoiles	Procédure

Concaténer les mots obtenus dans T1 en les séparant par un espace pour obtenir le message crypté.

Effacer l'espace qui se trouve à la fin de la chaine cryptée.

Procédure Separer (msg : chaine, @T :tab, @ n :entier)

Début

n ← 0

Tantque msg ≠ "" faire

p ← pos (" ", ch)

Si p ≠ -1 alors

T[n] ← sous-chaine (msg, 0, p)

Effacer (ch, 0, p+1)

n ← n+1

Sinon

T[n] ← msg

msg ← ""

FinSi

FinTantque

Fin

Procédure AjouterEtoiles (@T :tab, n :entier, @n1 :entier)

Début

n1 ← long(T[0])

Pour i de 1 a n-1 faire

Si (long(T[i]) > n1) alors

n1 ← long(T[i])

Fin si

Fin pour

Pour i de 0 a n-1 faire

Tantque long(T[i]) < n1 faire

T[i] ← T[i] + "*"

FinTantque

Fin pour

Fin

Procédure Repartir(T :tab, n :entier, @T1 :tab, n1 :entier)

Debut

Pour i de 0 a n1-1 faire

T1[i] ← ""

Pour j de 0 a n-1 faire

T1[i] ← T1[i] + T[j][i]

Fin pour

Fin pour

Fin

Placer les mots de la chaine à cryptée dans un tableau T de taille n.

TDOL	
Objet	T/N
p	Entier

- Pour chaque élément du tableau T, ajouter autant de fois le caractère "*" pour que sa longueur sera égale a celle du mot le plus long dans le tableau T
- **n1** est la longueur de la chaine la plus longue dans T

TDOL	
Objet	T/N
i	Entier

Repartir les lettres des mots du tableau T dans un nouveau tableau T1 de taille n1

TDOL	
Objet	T/N
i, j	Entier

Exercice 1 :

Dans un contexte algorithmique et pour chacune des propositions ci-dessous, cochez, en mettant une croix (X) dans la case qui correspond à la bonne réponse.

6) Quelle est la valeur de X après l'instruction suivante : $x \leftarrow 71 \text{ MOD } 12 \text{ DIV } 2 * 5 + 18 / 3 * 2$

<input type="checkbox"/>	13
<input type="checkbox"/>	28
<input type="checkbox"/>	37

7) Quelle est l'instruction erronée parmi les suivantes ?

<input type="checkbox"/>	Lire (n)
<input type="checkbox"/>	Lire ("n")
<input type="checkbox"/>	Lire (n, m)

8) Quelle est la valeur finale de a et b après exécution de la séquence suivante :

$a \leftarrow 25478$

$b \leftarrow 0$

Pour i de 0 à 3 faire

$b \leftarrow b + a \text{ MOD } 10$

$a \leftarrow a \text{ DIV } 10$

FinPour

<input type="checkbox"/>	a=25 et b= 24
<input type="checkbox"/>	a=2 et b=24
<input type="checkbox"/>	a=2 et b=25

9) A quelle condition la boucle suivante s'arrête-t-elle ?

Tantque (N MOD 2 \neq 0) OU NON (N \leq 50) Faire

.....

FinTantque

<input type="checkbox"/>	N est un nombre impair strictement supérieur à 50
<input type="checkbox"/>	N est un nombre pair strictement supérieur à 50
<input type="checkbox"/>	N est un nombre pair inférieur ou égal à 50

Exercice 2 :

Un nombre entier est premier s'il est divisible seulement par un et lui-même.

Ecrire l'algorithme d'une procédure qui prend en paramètre un entier $N \geq 2$ et qui affiche la liste des nombres premiers qui sont inférieurs ou égaux à N.

Exemple : Pour N=10

La procédure affiche la liste suivante : 2, 3, 5, 7

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

Exercice 3 :

On désire chercher le nombre d'occurrences de minimum dans un tableau T de N entiers. Pour cela, on donne le tableau de déclaration des objets suivant :

Objet	Type/Nature
T	Tableau de N entier
N, i, m, occ	Entier

- Compléter la séquence algorithmique présentée ci-dessous afin de chercher le nombre d'occurrences de minimum dans le tableau T

Début

$m \leftarrow \dots\dots$

 Pour i de à faire

 Si $T[i] \dots\dots m$ alors

$m \leftarrow \dots\dots$

 FinSi

 FinPour

 Ecrire ("Le minimum est", m)

$occ \leftarrow 0$

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

 Ecrire ("Le nombre d'occurrences de minimum est", occ)

Fin

Exercice 4 :

Le jeu de devinette se joue entre l'ordinateur et un joueur. Le principe de jeu est le suivant :

L'ordinateur choisit au hasard **N** nombres distincts entre 1 et 100. Ensuite, le joueur essaye de les deviner en proposant une liste de **N** nombres distincts entre 1 et 100. Pour chaque nombre proposé, le joueur gagne 5 points si ce nombre existe dans le bon ordre dans la liste à deviner et 2 points s'il existe dans un ordre différent. A la fin de jeu, l'ordinateur affiche la liste à deviner, la liste proposée et le nombre de points obtenus par le joueur.

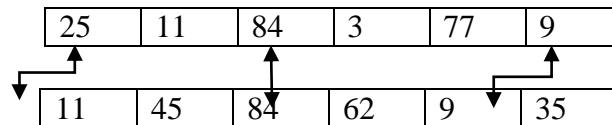
Pour numériser ce jeu, on se propose d'écrire un programme permettant de:

- Saisir N ($3 \leq N \leq 10$).
- Remplir les tableaux LD et LP contenant respectivement la liste de N nombres à deviner et la liste de N nombres proposés.
- Afficher la liste à deviner, la liste proposée et le nombre de points obtenus (score) par le joueur

Exemple :

Pour $N=5$

Liste à deviner (LD)



Liste proposée(LP)

L'ordinateur affiche :

25 11
11 45
84 84
3 62
77 9
9 35

Votre score est : 9 points

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

NB : la solution doit comporter les Tableaux de déclaration des objets et des nouveaux types.

Exercice 5 :

Le jeu de tir à l'arc consiste à tirer des flèches en utilisant un arc vers un endroit cible.

Un club de tir à l'arc organise une compétition entre ses joueurs. Cette compétition consiste à donner à chaque joueur trois essais de tir.

A chaque tir, le joueur gagne de **0** à **10** points selon l'endroit touché par la flèche sur une cible composée de **10** cercles concentriques comme l'illustre la figure ci-contre.

Après le 3^e essai, un score est calculé pour chaque joueur représentant la somme des résultats de ses trois essais, et ce pour classer les joueurs.



On se propose d'écrire un programme permettant :

- de remplir un tableau **A** par **N** noms de joueurs participant à cette compétition (avec $2 \leq N \leq 20$),
- de calculer le score de chaque joueur sachant que le résultat d'un essai est un entier positif inférieur ou égal à **10**.
- d'afficher la liste des joueurs ainsi que les scores correspondants en commençant par le meilleur.

Exemple :

Pour **N** = 4 et le tableau **A** suivant :

A	"ali amdouni"	"sarra trabelsi"	"mohamed Sahli"	"saber toukabri "
----------	---------------	------------------	-----------------	-------------------

En considérant que les résultats des trois essais ainsi que les scores calculés des quatre joueurs sont les suivants :

Joueur	Essai n°1	Essai n°2	Essai n°3	Score
ali amdouni	8	10	9	27
sarra trabelsi	10	7	10	27
mohamed Sahli	7	9	7	23
saber toukabri	9	9	10	28

La liste des joueurs affichée sera la suivante :

saber toukabri	28
ali amdouni	27
sarra trabelsi	27
mohamed Sahli	23

Travail demandé :

- Ecrire, en décomposant en modules, l'algorithme du programme principal.
- Ecrire l'algorithme de chaque module.

NB : la solution doit comporter les Tableaux de déclaration des objets et des nouveaux types.

Exercice 1 :

Dans un contexte algorithmique et pour chacune des propositions ci-dessous, cochez, en mettant une croix (X) dans la case qui correspond à la bonne réponse.

1- Quelle est la valeur de X après l'instruction suivante : $x \leftarrow 71 \text{ MOD } 12 \text{ DIV } 2 * 5 + 18 / 3 * 2$

	13
	28
X	37

2- Quelle est l'instruction erronée parmi les suivantes ?

	Lire (n)
X	Lire ("n")
	Lire (n, m)

3- Quelle est la valeur finale de a et b après exécution de la séquence suivante :

```

a ← 25478
b ← 0
Pour i de 0 à 3 faire
    b ← b+a MOD 10
    a ← a DIV 10
FinPour
    
```

	a=25 et b= 24
X	a=2 et b=24
	a=2 et b=25

4- A quelle condition la boucle suivante s'arrête-t-elle ?

```

Tantque (N MOD 2 ≠ 0) OU NON (N ≤ 50) Faire
    .....
FinTantque
    
```

	N est un nombre impair strictement supérieur à 50
	N est un nombre pair strictement supérieur à 50
X	N est un nombre pair inférieur ou égal à 50

Exercice 2 :

Procédure premiers (N :entier)

Début

Pour i de 2 à N faire

j ← 2

Tantque j ≤ i div 2 et i mod j ≠ 0 faire

j ← j+1

FinTantque

Si j > i div 2 alors

Ecrire(i)

FinSi

Fin Pour

Fin

Exercice 3 :

On désire chercher le nombre d'occurrences de minimum dans un tableau T de N entiers. Pour cela, on donne le tableau de déclaration des objets suivant :

Objet	Type/Nature
T N, i, m, occ	Tableau de N entier Entier

- Compléter la séquence algorithmique présentée ci-dessous afin de chercher le nombre d'occurrences de minimum dans le tableau T

Début

m ← T[0]

Pour i de 1 à n-1 faire

Si T[i] < m alors

m ← T[i]

FinSi

FinPour

Ecrire ("Le minimum est", m)

occ ← 0

Pour i de 0 à n-1 faire

Si T[i]=m alors

occ ← occ+1

FinSi

FinPour

Ecrire ("Le nombre d'occurrences de minimum est", occ)

Fin

Exercice 4 :

Algorithme de programme principal :

Algorithme Devinette

Début

Saisir (N)

RemplirLD (LD, N)

RemplirLP(LP, N)

Afficher (LD, LP, N)

Fin

TDOG :

**Tableau de déclaration de nouveaux types
(TDNT)**

Types

tab= Tableau de 10 entiers

Objet	Type/Nature
N	Entier
LD, LP	tab
Saisir , RemplirLD ,RemplirLP, Afficher	Procédures

Procédure Saisir(@N:entier)

Début

Répéter

Ecrire ("Donnez N: ") ;

Lire (N)

Jusqu'à (3 <=N<=10)

Fin

Procédure RemplirLD(@LD:tab, N:entier)

Début

Pour i de 0 à N-1 faire

Répéter

LD[i]←Alea(1, 100)

Jusqu'à (Existe (LD [i], LD, i)=faux)

Fin pour

Fin

TDOL	
Objet	Type/Nature
i	entier
Existe	fonction

Fonction Existe (x : entier, LD :tab, i :entier) :booléen

Début

test←faux

j←0

Tantque (test=faux) et (j<i) faire

Si LD[j]=x alors

test ←vrai

Sinon

j←j+1

FinSi

FinTantque

Retourner test

Fin

Procédure RemplirLP(@LP:tab, N:entier)

Début

Pour i de 0 à n-1 faire

Répéter

Ecrire ("Donner le nombre n° ", i+1 , " :")

Lire(LP[i])

Jusqu'à (LP[i]∈[1..100] et (Existe(LP [i], LP, i)=faux)

Fin pour

Fin

TDOL	
Objet	Type/Nature
i	entier
Existe	fonction

Procédure Afficher (LD , LP: tab, N : entier)

Début

score ← 0

Pour i de 0 à N-1 faire

indice ← Chercher (LP[i], LD, N)

Si indice=i alors

score ← score+5

Sinon si indice < N alors

score ← score+2

FinSi

FinPour

Pour i de 0 à N-1 faire

Ecrire (LD[i], LP[i])

FinPour

Ecrire ("Votre score est:", score)

Fin

TDOL	
Objet	Type/Nature
i	entier
score	entier
indice	entier
Chercher	fonction

Fonction Chercher (x :entier, TD : tab, N :entier): entier

Début

i ← 0

Tantque (i < N) et (TD[i] ≠ x) faire

i ← i+1

Fintantque

Retourner i

Fin

TDOL	
Objet	Type/Nature
i	entier

Exercice 5 :

Algorithme de programme principal :

Algorithme TirArc

Début

Saisir (N)
Joueurs (A, N)
Scores(S, N)
Trier (A, S, N)
Afficher (A, S, N)

Fin

TDOG :

Tableau de déclaration de nouveaux types (TDNT)	
Types	
Tab1= Tableau de 20 chaines	
Tab2= Tableau de 20 entiers	

Objet	Type/Nature
N	Entier
A	Tab1
S	Tab2
Saisir , Joueurs ,Scores, Trier, Afficher	Procédures

Procédure Saisir(@N:entier)

Début

Répéter
 Ecrire ("Donnez le nombre de joueurs: ") ;
 Lire (N)
Jusqu'à (2 <=N<=20)

Fin

Procédure Joueurs(@A:tab, N:entier)

Debut

Pour i de 0 à N-1 faire
 Répéter
 Ecrire ("Donnez le nom de joueur n°", i+1, ": ")
 Lire (A[i])
 Jusqu'à (0 < long (A[i])) et (alpha(A[i]))
FinPour

Fin

Fonction alpha(ch : chaine) : booléen

Début

i ← 0
test ← Vrai
Tantque (i<long(ch)) et (test=Vrai) faire
 test ← Majus(ch[i]) ∈ ["A".."Z"]
 i ← i+1
Fin tantque
Retourner test

Fin

TDOL	
Objet	Type/Nature
i	entier
alpha	fonction

TDOL	
Objet	Type/Nature
i	entier
test	booléen

Procédure Scores (@S:Tab2, N:entier)

Début

Pour i de 0 à N-1 faire

score←0

Pour j de 1 à 3 faire

Répéter

Ecrire ("Donner l'essai n° ", i, " :")

Lire(essai)

Jusqu'à (essai∈[0..10])

score ←score+essai

FinPour

S [i]←score

Fin pour

Fin

Procédure Trier(@A :Tab1,@S :Tab2, N : entier)

Début

Pour i de 0 à N-2 faire

indice←i

Pour j de i+1 a N-1 faire

Si S[j]> S[indice] alors

indice←j

Fin si

Fin pour

Si indice ≠i alors

Aux1 ←S[indice]

S[indice] ←S[i]

S[i] ← Aux1

Aux2←A[indice]

A[indice] ← A[i]

A[i] ← Aux2

Finsi

Finpour

Fin

TDOL	
Objet	Type/Nature
i, j, indice, Aux1	Entier
Aux2	Chaine

Procédure Afficher (A :Tab1 , S:Tab2, N : entier)

Début

Pour i de 0 à N-1 faire

Ecrire (A[i]), S[i])

FinPour

Fin

TDOL	
Objet	Type/Nature
i	entier

.....