

# Exercices

## Exercice n°1:

Faire l'analyse et déduire les algorithmes d'un programme qui permet de **fusionner** le contenu des deux tableaux TN de n chiffres négatifs et TP de n chiffres positifs, dans un tableau T, les n chiffres positifs alternés avec les n chiffres négatifs, avec  $n \in [1..10]$ .

**Remplissage des deux tableaux :** Il est recommandé de n'utiliser qu'un seul module pour réaliser cette tâche, exploité en deux appels : la première fois pour le remplissage de TN et la deuxième fois pour le remplissage de TP.

**Exemple :**

TN	-7	-2	-5	-8	-1					
	1	2	3	4	5					
TP	2	8	9	6	4					
	1	2	3	4	5					
T	-7	2	-2	8	-5	9	-8	6	-1	4
	1	2	3	4	5	6	7	8	9	10

## Exercice n°2 :

Ecrire un programme, en Turbo Pascal, qui permet de calculer les sommes S1 et S2 des entiers se trouvant respectivement sur la 1<sup>ère</sup> moitié et la 2<sup>ème</sup> moitié d'une matrice M de nXn entiers ( $n \in [3..10]$ ).

**Calcul de la somme d'une moitié :** Il est recommandé de n'utiliser **qu'une seule fonction récursive** pour réaliser cette tâche, exploité en deux appels : la première fois pour le calcul de la 1<sup>ère</sup> somme (S1) et la deuxième fois pour le calcul de la 2<sup>ème</sup> somme (S2).

**Exemple :**

M de taille 4x4

12	3	0	5
16	2	6	25
26	15	9	86
8	92	71	0

$$S1 = 16+26+15+8+92+71 = 228$$

$$S2 = 3+0+5+6+25+86 = 125$$

[www.BAC.org.tn](http://www.BAC.org.tn)  
 Page: BAC-TUNISIE  
 Tél: 25 361 197 / 53 371 502

**Exercice n°3 :**

Faire l'analyse et déduire l'algorithme d'un module permettant de calculer et d'afficher le  $n^{\text{ième}}$  terme de la suite U donnée par la relation de récurrence suivante :

$$U \begin{cases} U_1=1 \\ U_2=3 \\ U_n=3U_{n-1} + \frac{U_{n-2}}{3} \quad (\text{pour } n > 2). \end{cases}$$

- Proposer une solution itérative utilisant un tableau,
- Proposer une solution itérative sans utiliser de tableau,
- Proposer une solution récursive.

**Exercice n°4 :**

www.BAC.org.tn  
Page BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

Soient U et V deux suites croisées définies par :

$$U \begin{cases} U_1=1 \\ U_n = \sqrt{V_{n-1}} + (U_{n-1}^2) \quad (\text{pour } n \geq 2). \end{cases}$$

&

$$V \begin{cases} V_1=1 \\ V_n = U_n + V_{n-1} \quad (\text{pour } n \geq 2). \end{cases}$$

Ecrire un module récursif permettant de vérifier si un réel x (un paramètre formel) est un terme de l'une des deux suites U ou V.

**Exercice n°5 :**

Soit A une matrice carrée de taille  $n \times n$  (avec  $1 < n \leq 10$ ) contenant des caractères imprimables, leurs codes ASCII  $\in [33.. 127]$ , proposés par l'ordinateur (aléatoires).

Ecrire un programme, en Turbo Pascal, qui permet de remplir et d'afficher une  $2^{\text{ème}}$  matrice B par les valeurs (des codes ASCII) correspondantes aux caractères se trouvant dans la matrice A.

**Exemple :**

Si  $n = 5$  et le contenu de A est :

-	E	''	-	e
C	I	a	}	'
}	-	d	E	C
X	\$	b	}	c
d	x	7	G	S

Le contenu de B sera :

45	69	34	45	101
67	105	97	125	39
125	45	100	69	67
88	36	98	125	99
100	120	55	71	83

**Exercice n°6 :**

Faire l'analyse et déduire les algorithmes d'un programme qui permet d'afficher la fréquence de chaque lettre se trouvant dans une chaîne de caractères non vide.

**Exemple :** Si la chaîne de caractères **CH = "Essentiel Informatique"**

L'ordinateur affichera :

- |     |          |     |          |
|-----|----------|-----|----------|
| "A" | = 1 fois | "N" | = 2 fois |
| "E" | = 4 fois | "Q" | = 1 fois |
| "F" | = 1 fois | "R" | = 1 fois |
| "I" | = 3 fois | "S" | = 2 fois |
| "L" | = 1 fois | "T" | = 2 fois |
| "M" | = 1 fois | "U" | = 1 fois |
| "O" | = 1 fois |     |          |

**Exercice n°7 :**

Faire l'analyse et déduire l'algorithme d'un module qui servira pour le calcul de la somme (S) des entiers se trouvant sur l'une des deux parties, selon le choix (p) de l'utilisateur, de la matrice M de taille  $n \times n$  (la partie des cases blanches ou la partie des cases noires). M, n et p sont des paramètres formels :

Soit M de Taille  $6 \times 6$  :  
 S des **blanches** = 96  
 S des **noires** = 635

2	15	-2	40	3	6
-9	78	26	-5	6	31
99	38	38	99	452	-7
567	26	304	105	2	60
10	15	-8	0	41	300
-800	-875	42	84	-9	-43

www.BAC.org.tn  
 Page BAC-TUNISIE  
 Tél: 25 361 197 / 53 371 502

**Exercice n°8 :**

Ecrire un programme, en Turbo Pascal, permettant d'afficher un losange d'hauteur n lignes (avec  $2 \leq n \leq 24$  et n est impair).

**Exemple :** pour n =11, l'ordinateur affichera :

```

      *
     ***
    *****
   *********
  ***********
 *****
  *****
   *****
    *****
     *****
      *
    
```

**Exercice n°9 :**

Ecrire un programme, en Turbo Pascal, qui permet de calculer les sommes  $S_i$  des différents cadres que l'ordinateur peut former dans une matrice A de taille n X n ( $n \in [2..10]$  et impair).

**Exemples :**

1) A de taille 3x3

12	40	3
1	15	2
6	90	15

- ✓  $S_1 = 12+40+3+1+6+90+15+22 = 169$
- ✓  $S_2 = 15$

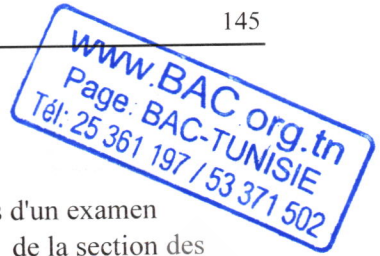
2) A de taille 5x5

1	2	4	1	3
8	-1	15	0	5
5	10	40	4	-2
4	9	8	3	0
30	28	5	11	3

- ✓  $S_1 = 1+2+4+1+3+5-2+0+3+11+5+28+30+4+5+8 = 108$
- ✓  $S_2 = -1+15+0+4+3+8+9+10 = 48$
- ✓  $S_3 = 40$

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502





**Exercice n°10 :**

On désire faire une étude statistique des notes d'un examen informatique pratique des  $n$  élèves ( $n \in [50.. 180]$ ) de la section des sciences de l'informatique de votre lycée.

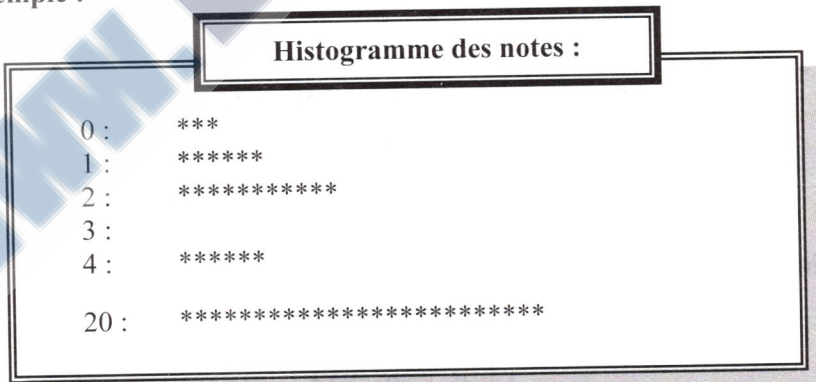
On aura besoin, donc, des structures de données suivantes :

- 1) Un premier tableau intitulé **TN** sera réservé pour contenir les notes initiales.
- 2) Un deuxième tableau intitulé **TA** sera réservé pour contenir les  $n$  notes du tableau TN arrondies à l'entier le plus proche.
- 3) Un troisième tableau intitulé **TH** dont chaque  $TH[i]$  est inspiré d'une case  $TA[i]$  du tableau TA et donne le nombre d'élèves ayant une note  $i-1$  ( $0 \leq i \leq 20$ ).

Faire l'analyse et déduire les algorithmes d'un programme qui doit assurer les tâches suivantes :

- ✓ Saisir le nombre de notes  $n$ ,
- ✓ Remplir les  $n$  notes dans TN,
- ✓ Remplir leurs arrondies dans TA,
- ✓ Remplir les occurrences des notes arrondies dans TH,
- ✓ Afficher un graphique sous formes d'histogramme à partir du tableau TH sur 21 lignes (une par note). Chaque ligne comprendra la note suivie d'autant d'étoiles (\*) qu'il n'y a d'élèves ayant obtenu cette note.

**Exemple :**



L'histogramme indique que 3 élèves ont eu 0, 6 élèves ont eu 1, 11 élèves ont eu 2, 0 élèves ont eu 3..... 25 élèves ont eu 20.

**Exercice n°11 :**

Ecrire un programme, en Turbo Pascal, qui permet de construire une matrice **A** de taille 3X3 initialisée par des zéro, à partir de trois tableaux la représentant **TV**, **TL** et **TC**.

Les trois tableaux sont de  $n$  entiers chacun ( $n \in [3, 6, 9]$ ), saisis par l'utilisateur de la manière suivante :

- Le premier tableau **TV** contenant les valeurs non nulles des éléments de la matrice **A**,
- Le deuxième tableau **TL** contenant leurs numéros de ligne dans la matrice **A** :
  - ✓ Les numéros des lignes ne peuvent être que 1,2, **et/ou** 3.
  - ✓ Un numéro de ligne ne peut se présenter que 3 fois au maximum.
- Le troisième tableau **TC** contenant leurs numéros de colonne dans la matrice **A** :
  - ✓ Les numéros des colonnes ne peuvent être que 1,2, **et/ou** 3.
  - ✓ Un numéro de colonne ne peut se présenter que 3 fois au maximum.

Une combinaison de numéro de ligne et de colonne ne peut dans aucun cas être redondante (une case de **A** n'est exploitée qu'une seule fois).

**Exemple :**

**TV**

9	12	53	101	<b>88</b>	49
---	----	----	-----	-----------	----

**Valeurs non nulles dans A**

**TL**

1	1	2	2	<b>2</b>	3
---	---	---	---	----------	---

 N° de lignes

**TC**

2	3	1	2	<b>3</b>	3
---	---	---	---	----------	---

 N° de colonnes

**Résultat :**

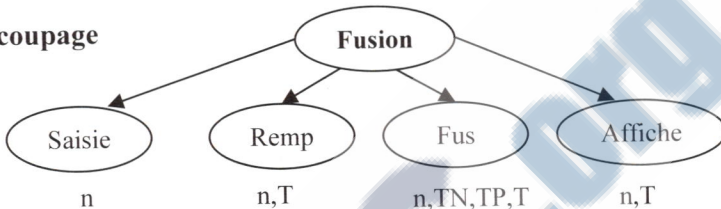
La matrice **A** sera comme suit :

	1	2	3
1	0	9	12
2	53	101	<b>88</b>
3	0	0	49

# Corrigés

## Exercice n°1 :

### I- Découpage



- ✓ **Fusion** : programme principal, qui permet de fusionner le contenu des deux tableaux TN et TP dans le tableau T.
- ✓ **Saisie** : procédure, permettant de saisir le nombre d'éléments à remplir dans un tableau TN ou TP.
- ✓ **Remp** : procédure, permettant de remplir un tableau T par n chiffres.
- ✓ **Fus** : procédure, permettant de fusionner les deux tableaux TN et TP dans le tableau T.
- ✓ **Affiche** : procédure, permettant d'afficher le contenu du tableau T.

### II- Analyse & Algorithme

\*\*\* Programme principal : Fusion \*\*\*

#### 1) Spécification du problème :

Résultat = Affiche( 2\*n,T)

Traitements =

- ✓ n = Saisie (n)
- ✓ T = Fus (n, TN, TP, T)
- ✓ TN = Remp (n, -1, TN)
- ✓ TP = Remp (n, 1, TP)
- ✓ Saisie, Fus, Remp et Affiche = Procédures

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**T.D.O.U.G**

Objet	Nature / Type	Rôle
Affiche	Procédure	Afficher le contenu du tableau T.
n	Entier	Contenir le nombre de chiffres à remplir.
T	TAB	Contenir des chiffres.
Saisie	Procédure	Saisir le nombre de chiffres à remplir.
TN	CHIF	Contenir des chiffres négatifs.
TP	CHIF	Contenir des chiffres positifs.
Fus	Procédure	Fusionner TN et TP dans T.
Remp	Procédure	Remplir un tableau par des chiffres.

**T.D.N.T**

Type
CHIF = Tableau de 10 Octet
TAB = Tableau de 20 Octet

**2) Algorithme :**

- 0) Début Fusion
- 1) Saisie (n)
- 2) Ecrire (" \*\*\* Saisie des chiffres négatifs \*\*\* ")  
Remp (n, TN, -1)
- 3) Ecrire (" \*\*\* Saisie des chiffres positifs \*\*\* ")  
Remp (n, TP, 1)
- 4) Fus (n, TN, TP, T)
- 5) Ecrire ("Le contenu du nouveau tableau T : ")  
Affiche (2\*n, T)
- 6) Fin Fusion

**\*\*\* Procédure : Saisie \*\*\***

**1) Spécification du problème :**

**Résultat = n**

**Traitements =**

- ✓ n = Donnée avec  $n \in [1..10]$
- ✓ n = Paramètre formel

[www.BAC.org.tn](http://www.BAC.org.tn)  
 Page : BAC-TUNISIE  
 Tél : 25 361 197 / 53 371 502



2) **Algorithmme :**

- 0) Procédure Saisie (VAR n : Entier)
  - 1) Répéter
    - Ecrire ("Saisir le nombre de chiffres à introduire : ")
    - Lire (n)
    - Jusqu'à n Dans [1..10]
  - 2) Fin Saisie

\*\*\* Procédure : Remp \*\*\*

1) **Spécification du problème :**

Résultat = T

Traitements =

- ✓ T = Données
  - ↳  $T[i] = \text{Donnée avec } S * T[i] \in [0..9]$
- ✓ i = Compteur allant de 1 à n
- ✓ n, T et S = Paramètres formels



T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) **Algorithmme :**

- 0) Procédure Remp (n, S : Entier ; VAR T : CHIF)
  - 1) Pour i de 1 à n Faire
    - Répéter
      - Ecrire ("T[" , i, "] = ")
      - Lire (T[i])
      - Jusqu'à  $S * T[i]$  Dans [0..9]
    - FinPour
  - 2) Fin Remp

S = 1 :  $1 * T[i] = T[i]$ , donc l'utilisateur doit taper un chiffre positif pour que la condition d'arrêt soit vérifiée.  
 S = -1 :  $-1 * T[i] = -T[i]$ , donc l'utilisateur doit taper un chiffre négatif pour que la condition d'arrêt soit vérifiée.

\*\*\* Procédure : Fus \*\*\*

1) Spécification du problème :

Résultat = T

Traitements =

- ✓  $T = [ \quad ]$
- ↳  $T [2*i-1] \leftarrow TN[i]$  et
- ↳  $T [2*i] \leftarrow TP[i]$

Remarquez qu'il s'agit d'une relation de récurrence entre les compteurs des 3 tableaux TN, TP et T.

- ✓  $i$  = Compteur allant de 1 à  $n$
- ✓  $n, TN, TP$  et  $T$  = Paramètres formels

T.D.O.U.L

Objet	Nature / Type	Rôle
i	Entier	Compteur

2) Algorithme :

0) Procédure Fus ( $n$  : Entier ; TN, TP : CHIF ; VAR T: TAB)

- 1) Pour  $i$  de 1 à  $n$  faire
  - $T [2*i-1] \leftarrow TN[i]$
  - $T [2*i] \leftarrow TP[i]$
- FinPour

2) Fin Fus

\*\*\* Procédure : Affiche \*\*\*

1) Spécification du problème :

Résultat = Aff

Traitements =

- ✓ Aff = [  $\quad$  ]
- Afficher (T[i])
- ✓  $i$  = Compteur allant de 1 à  $n$
- ✓  $n$  et  $T$  = Paramètres formels

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

T.D.O.U.L

Objet	Nature/Type	Rôle
i	Entier	Compteur

2) **Algorithme :**

0) Procédure Affiche (n: Entier ; T:TAB)

1) Pour i de 1 à n Faire

Ecrire (T [i])

FinPour

2) Fin Affiche

N'oubliez pas que n est un paramètre formel et sa valeur effective est égale à  $(2*n)$ .

**Exercice n°2 :**

**Program** Sommes;

Uses WinCRT;

Type

Mat = Array [1..10 , 1..10] of integer;

Var

M :Mat;

n : integer;

**Procedure** Saisie (Var n : integer);

Begin

Repeat

Write ('Saisir le nombre de lignes/colonnes : ');

Readln (n);

Until n in [3..10];

**End;**

**Procedure** Remp (n : integer ; Var M: MAT);

Var

i, j : integer;

Begin

For i := 1 to n Do

For j := 1 to n Do

begin

Write ('M['',i'',',j,'] = ');

Readln (M[i,j]);

end;

**End;**

**Function** Calcul (n: integer ; M: Mat ; x: integer): Longint;

Var

S : Longint;

i, j : integer;

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

```

begin
  S:=0;
  For i:= 1 to n Do
    For j:= 1 to n Do
      If x*i > x*j
        Then
          S:= S+M[i,j];
    Calcul := S;
  End;

```

x = 1 : i>j le numéro de la ligne est supérieur à celui de la colonne, c'est la moitié se trouvant **au dessous** de la diagonale.  
 x = -1 : -i>-j le numéro de la colonne est supérieur à celui de la ligne c'est la moitié se trouvant **au dessus** de la diagonale.

{Programme principal}

```

Begin
  Saisie (n);
  Remp (n, M);
  Writeln ('S1 = ', Calcul (n, M, 1));
  Write ('S2 = ', Calcul (n, M, -1));
End.

```

www.BAC.org.tn  
 Page: BAC-TUNISIE  
 Tél: 25 361 197 / 53 371 502

### Exercice n°3 :

a- Solution itérative utilisant un tableau :

\*\*\* Procédure : Affiche \*\*\*

1) Spécification du problème :

Résultat : Afficher (T[n])

Traitements :

✓  $T = [ T[1] \leftarrow 1 ; T[2] \leftarrow 3 ]$

↪  $T[i] \leftarrow 3T[i-1] + \frac{T[i-2]}{3}$

✓ i = Compteur allant de 3 à n

✓ n = Paramètre formel



**T.D.O.U.L**

Objet	Nature/Type	Rôle
i	Entier	Compteur.
T	Tableau de 78 réels	Contenir les n termes de la suite U.

**78 réels ??? :**

Le dernier terme que l'ordinateur peut calculer et afficher est le 78<sup>ème</sup> terme dont la valeur  $\approx 7,916 \cdot 10^{37}$ . La valeur du 79<sup>ème</sup> terme dépasse la borne supérieure de la plage des valeurs d'une variable de type réel (en Turbo Pascal) étant presque égale à  $3,4 \cdot 10^{38}$ .

**2) Algorithme :**

0) Procédure Affiche (n : Entier)

1)  $T[1] \leftarrow 1$  $T[2] \leftarrow 3$ 

Pour i de 3 à n faire

 $T[i] \leftarrow 3 * T[i-1] + T[i-2] / 3$ 

FinPour

2) Ecrire ("U", n, " = ", T[n])

3) Fin Affiche

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**b- Solution itérative n'utilisant pas un tableau :****\*\*\* Procédure : Affiche \*\*\*****1) Spécification du problème :****Résultat :** Afficher (U)**Traitements :** $U = [ a \leftarrow 1 ; b \leftarrow 3 ]$ ✓  $U \leftarrow 3b + \frac{a}{3}$  ; U est le terme actuel à calculer✓ La variable (a) représente la valeur avant précédente de U et qui doit être renouvelée à chaque fois où l'ordinateur passe pour calculer un nouveau terme de U par la valeur de b :  $a \leftarrow b$ ✓ La variable b représente la valeur précédente de U et qui doit être renouvelée à chaque fois où l'ordinateur passe pour calculer un nouveau terme de U par la valeur de U :  $b \leftarrow U$

- ✓ Il faut utiliser un compteur **i allant de 3 à n**
- ✓  $n$  = Paramètre formel

**T.D.O.U.L**

Objet	Nature/Type	Rôle
i	Entier	Compteur.
a	Réel	Contenir $U_{i-2}$ .
b	Réel	Contenir $U_{i-1}$ .
U	Réel	Contenir le terme actuel à calculer U.

**2) Algorithme :**

0) Procédure Affiche ( $n$  : Entier)

1)  $a \leftarrow 1$

$b \leftarrow 3$

Pour  $i$  de 3 à  $n$  faire

$U \leftarrow 3*b + a/3$

$a \leftarrow b$

$b \leftarrow U$

FinPour

2) Ecrire ("U", $n$ , " = ", U)

3) Fin Affiche



**c- Solution récursive :**

\*\*\* Procédure : Affiche \*\*\*

**1) Spécification du problème :**

Résultat : Afficher (U) si  $i = n$

Traitements :

- ✓  $U \leftarrow 3b + \frac{a}{3}$  ; U est le terme actuel à calculer
- ✓ Renouveler l'appel à la procédure Affiche ( $n, i+1, U, b$ ) si  $i < n$ 
  - Le paramètre effectif  $i+1$  sera substitué au paramètre formel  $i$  de la procédure Affiche.
  - Le paramètre effectif U sera substitué au paramètre formel  $b$  de la procédure Affiche.
  - Le paramètre effectif  $b$  sera substitué au paramètre formel  $a$  de la procédure Affiche.
- ✓  $n, i, a$  et  $b$  = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
U	Réel	Contenir le terme actuel à calculer.

2) **Algorithmme :**

0) Procédure Affiche (n, i : Entier ; b, a : Réel)

1)  $U \leftarrow 3*b + a/3$

2) Si  $i = n$

Alors

Ecrire ("U", n, " = ", U)

Sinon

Affiche (n, i+1, U, b)

FinSi

3) Fin Affiche



**Exercice n°4 :**

\*\*\* Procédure : Verif \*\*\*

1) **Spécification du problème :**

Résultat : Trait

Traitements :

- ✓ Trait = [ ]
- ↻ Soit afficher (x, " est terme de l'une des deux suites ") Si  $U$  ou  $V = x$
- ↻ Soit afficher (x, " n'est pas un terme des deux suites ") Si  $U$  et  $V > x$  ( $U$  et  $V$  sont deux suites croissantes)
- ↻ Soit appeler de nouveau Verif ( $U, V, x$ ) Si  $U$  ou  $V < x$
- ✓  $U \leftarrow \text{Racine\_carré}(V) + \text{Carré}(U)$
- ↻ La valeur de  $U$  actuelle est un résultat de  $U$  et  $V$  précédentes
- ✓  $V \leftarrow (U + V)$
- ↻ La valeur de  $V$  actuelle est un résultat de  $U$  actuelle et  $V$  précédente
- ✓  $U, V$  et  $x =$  Paramètres formels

**2) Algorithme :**

0) Procédure Verif (U, V, x : Réel)

1)  $U \leftarrow \text{Racine\_Carré}(V) + \text{Carré}(U)$  $V \leftarrow U + V$ 

2) Si (U = x) OU (V = x)

Alors

Ecrire (x, " est un terme de l'une des deux suites ")

Sinon

Si (U &gt; x) ET (V &gt; x)

Alors

Ecrire (x, " n'est pas un terme des deux suites ")

Sinon

**Verif (U, V, x)**

Finsi

FinSi

3) Fin Verif

**Exercice n°5 :****Program ASCII;**

Uses WinCRT;

Type

MAT = Array [1..10, 1..10] of **Integer**;TAM = Array [1..10, 1..10] of **Char**;

Var

A : TAM ;

B : MAT;

n:integer;

**Procedure RempA (Var n : integer ; Var A : TAM);**

Var

L, C : integer;

Begin

Repeat

Write ('Saisir le nombre de lignes et/ou de colonnes :');

Readln (n);

Until n in[2..10];





**Randomize;**

```
For L:=1 to n do
  For C:=1 to n do
    A[L,C]:=CHR(Random(95)+33);
```

Remplir la matrice A par des caractères imprimables aléatoires.

**End;**

**Procedure** RempB (n:integer; A : TAM; Var B: MAT);

Var

L,C:integer;

Begin

```
For L:=1 to n do
  For C:=1 to n do
    B[L,C] := ORD(A[L,C]);
```

Remplir la matrice B par le code ASCII de chaque caractère se trouvant dans la matrice A.

**End;**

**Procedure** Affiche1 (n : integer ; M: TAM);

Var

L, C : integer;

Begin

Writeln ('\*\*\* Le contenu de la matrice A : \*\*\*');

```
For L:=1 to n do
begin
  For C:=1 to n do
    Write (M[L,C]:4);
  Writeln;
end;
```

Affiche1 et Affiche2 ont le même corps de programme mais l'unique différence réside au niveau du type de la matrice à afficher.

**End;**

**Procedure** Affiche2 (n : integer ; M: MAT);

Var

L, C : integer;

Begin

Writeln ('\*\*\* Le contenu de la matrice B : \*\*\*');

```
For L:=1 to n do
begin
  For C:=1 to n do
    Write (M[L,C]:4);
  Writeln;
end;
```

**End;**

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

{Programme principal}

**Begin**

RempA (n, A);

RempB (n, A, B);

Affiche1 (n, A);

Writeln;

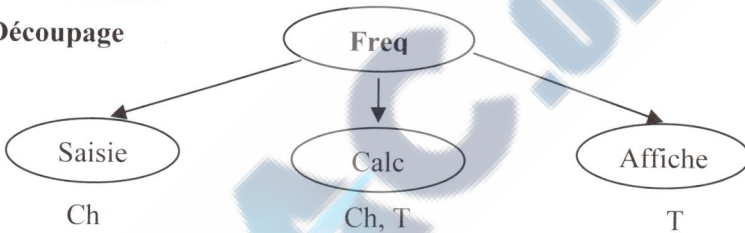
Affiche2 (n, B);

**End.**



**Exercice n°6 :**

**I- Découpage**



- ✓ **Freq** : programme principal, qui permet d'afficher la fréquence d'occurrences de chaque lettre se trouvant dans une chaîne Ch
- ✓ **Saisie** : procédure, permettant de saisir une chaîne non vide.
- ✓ **Calc** : procédure, permettant de calculer la fréquence de chaque lettre se trouvant dans Ch en remplissant un tableau de compteurs T.
- ✓ **Affiche** : procédure, permettant d'afficher les valeurs non nulles du tableau de compteurs T.

**II- Analyses & Algorithmes**

\*\*\* Programme principal : Freq \*\*\*

**1) Spécification du problème :**

Résultat = Affiche (T)

Traitements =

- ✓ T = Calc (Ch, T)
- ✓ Ch = Saisie (Ch)
- ✓ Affiche, Saisie et Calc = Modules

**T.D.O.U.G**

Objet	Nature/Type	Rôle
T	TAB	Contenir les compteurs des lettres.
Ch	Chaîne de caractères	Contenir une chaîne non vide
Calc	Procédure	Remplir le tableau des compteurs T.
Affiche	Procédure	Afficher les compteurs non nuls de T.
Saisie	Procédure	Saisir une chaîne de caractères non vide

**T.D.N.T**

Type
TAB = Tableau de "A" à "Z" Octet

**2) Algorithme :**

- 0) Début Freq
- 1) Saisie (Ch)
- 2) Calc (Ch, T)
- 3) Affiche (T)
- 4) Fin Freq

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**\*\*\* Procédure : Saisie \*\*\***

**1) Spécification du problème :**

Résultat = CH

Traitements =

- ✓ CH = Donnée avec CH est non vide
- ✓ CH = Paramètre formel

**2) Algorithme**

- 0) Procédure Saisie (Var Ch : Chaîne de caractères)
- 1) Répéter
  - Ecrire ("Saisir une chaîne de caractères non vide : ")
  - Lire (Ch)
  - Jusqu'à Ch <> ""
- 2) Fin Saisie

\*\*\* Procédure : Calc \*\*\*

1) Spécification du problème :

Résultat = T

Traitements =

- ✓  $T = [ \quad ]$ 
  - ↪  $T[j] \leftarrow 0$  : Initialiser les cases de T (compteurs) par 0,
  - ↪  $T[\text{Majus}(\text{ch}[i])] \uparrow +1$  : Chaque lettre se trouvant dans Ch engendre l'incréméntation de son compteur (une case de T) par 1.
- ✓  $j =$  Compteur allant de "A" à "Z"
- ✓  $i =$  Compteur allant de 1 à la fin de Ch
- ✓ Ch et T = Paramètres formels

T.D.O.U.L

Objet	Nature/Type	Rôle
j	Caractère	Compteur.
i	Entier	Compteur.

2) Algorithmes :

0) Procédure Calc (Ch : Chaîne de caractères ; Var T : TAB )

1) Pour j de "A" à "Z" Faire

$T[j] \leftarrow 0$

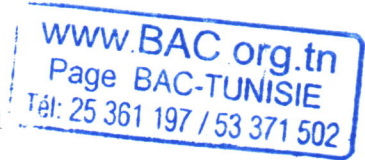
FinPour

2) Pour i de 1 à Long (Ch) Faire

$T[\text{Majus}(\text{ch}[i])] \leftarrow T[\text{Majus}(\text{ch}[i])] + 1$

FinPour

3) Fin Calc



**Quelle instruction!!!**

$T[\text{Majus}(\text{ch}[i])] \leftarrow T[\text{Majus}(\text{ch}[i])] + 1$

Supposons que la lettre pointée (ch[i]) est "a" la case T["A"] sera incrémentée par 1, si Ch[i] est "D" la case T["D"] sera incrémentée par 1, etc.



**\*\*\* Procédure : Affiche \*\*\***

**1) Spécification du problème :**

Résultat = Aff

Traitements =

- ✓ Aff = [            ]
- Afficher (i, T[i]) Si  $T[i] \neq 0$
- ✓ i = Compteur allant de "A" à "Z"

**T.D.O.U.L**

Objet	Nature/Type	Rôle
i	Caractère	Compteur.

**2) Algorithme :**

0) Procédure Affiche (T : TAB)

- 1) Pour i de "A" à "Z" Faire
  - Si  $T[i] \neq 0$
  - Alors
  - Ecrire (i, " : ", T[i], " fois")
  - FinSi
  - FinPour
- 2) Fin Affiche

[www.BAC.org.tn](http://www.BAC.org.tn)  
 Page: BAC-TUNISIE  
 Tél: 25 361 197 / 53 371 502

**Exercice n°7 :**

**\*\*\* Fonction : Somme \*\*\***

**1) Spécification du problème :**

Résultat : Somme  $\leftarrow$  S

Traitements :

- ✓  $S = [ \quad S \leftarrow 0 \quad ]$
- ↻  $S \leftarrow S + M[i,j]$  Si  $(i + j) \text{ MOD } 2 = P$
- ↻ Lorsque  $P = 0$  cela veut dire que  $(i+j) \text{ MOD } 2 = 0$  : i et j ont la même parité on parle alors des cases blanches de la matrice M,
- ↻ Lorsque  $P = 1$  cela veut dire que  $(i+j) \text{ MOD } 2 = 1$  : i et j ont une parité opposée on parle alors des cases noires de la matrice M,
- ✓ i = Compteur allant de 1 à n

- ✓  $j$  = Compteur allant de 1 à  $n$
- ✓  $M, n$  et  $P$  = Paramètres formels

**T.D.O.U.L**

Objet	Nature/Type	Rôle
S	Entier Long	Contenir une somme d'entiers.
$i, j$	Entier	Compteurs.

**1) Algorithme :**

0) Fonction Somme ( $M$  : MAT ;  $n, P$ : Entier) : Entier Long

1)  $S \leftarrow 0$

Pour  $i$  de 1 à  $n$  Faire

    Pour  $j$  de 1 à  $n$  Faire

        Si  $(i+j) \text{ MOD } 2 = P$

            Alors  $S \leftarrow S + M[i,j]$

        FinSi

    FinPour

FinPour

2) Somme  $\leftarrow S$

3) Fin Somme

www.BAC.org.tn  
Page BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**Exercice n°8 :**

**Program** Losange;

Uses WinCRT;

Var

$n$ : Integer ;

**Procedure** Saisie (Var  $n$  : integer);

Begin

    Repeat

        Write (' Ecrire l"hauteur du losange entre 3 et 23 (impair) : ');

        Readln ( $n$ ) ;

    Until ( $n$  In [3..23]) and ( $n \text{ Mod } 2 = 1$ ) ;

**End;**

**Procedure** Affiche (n : integer);

Var

C, L, i, j, h : integer;

Begin

Clrscr;

C := 40;

h := n div 2;

For i := 1 To h+1 Do

Begin

Gotoxy (C-i, i);

For j := 1 To 2\*i-1 Do

Write ('\*') ;

End;

L:=h+1;

For i := h downto 1 Do

Begin

L:=L+1;

Gotoxy (C-i, L);

For j := 2\*i-1 downto 1 Do

Write ('\*') ;

End;

**End;**

{Programme principal}

**Begin**

Saisie (n);

Affiche (n);

**End.**

**Exercice n°9:**

**Program** Cadres;

Uses WinCRT;

**Type**

MAT = Array [1..10,1..10] Of integer;

Var

A: MAT;

n : integer;

h est le milieu du losange :  
l'hauteur d'une moitié du losange.

Affichage de la 1<sup>ère</sup> partie du  
losange, c'est le 1<sup>er</sup> triangle isocèle  
du losange.

Affichage de la 2<sup>ème</sup> partie du  
losange, c'est le 2<sup>ème</sup> triangle isocèle  
du losange.

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**Procedure** Saisie (Var n : integer);

Begin

Repeat

Write ('Le nombre de lignes/ colonnes : ');

Readln (n);

Until n in [3, 5, 7, 9];

**End;**

**Procedure** Remp (n: integer ; Var A : MAT);

Var

i, j: integer;

Begin

For i := 1 To n Do

For j := 1 To n Do

Repeat

Write ('A[',i,',',j,'] = ');

Readln (A[i,j]);

Until A[i,j] <> 0;

**End;**

**Function** Som (x, n : integer ; A: MAT): LongInt;

Var

S : LongInt;

i, j: integer;

Begin

S := 0;

For i := x To n-x+1 Do

S := S + A[x,i] + A[n-x+1,i] + A[i,x] + A[i,n-x+1];

S := S - (A[x,x] + A[n-x+1,x] + A[x,n-x+1] + A[n-x+1,n-x+1]);

Som := S;

**End;**

Remplissage de la matrice  
A par des entiers non nuls.

**Som** : Calculer la somme des entiers  
se trouvant dans un cadre numéro x.

On doit trancher de S les entiers se trouvant sur  
les 4 coins du cadre et qui sont comptabilisés  
deux fois au cours du premier calcul de S.

Si par exemple,  $x = 1$  cela veut dire que la fonction Som doit retourner la somme des entiers se trouvant sur la 1<sup>ère</sup> ligne, la 1<sup>ère</sup> colonne, la dernière ligne et la dernière colonne.

Attention, les intersections entre ces deux lignes et ces deux colonnes forment 4 coins, exemple l'intersection entre la 1<sup>ère</sup> ligne et la 1<sup>ère</sup> colonne est un coin comptabilisé doublement : la 1<sup>ère</sup> fois avec la ligne et la 2<sup>ème</sup> fois avec la colonne !!

**Procedure** Sommes (n : integer ; A : MAT);

Var

m, i: integer;

Begin

m := n DIV 2;

For i := 1 m To Do

Writeln ('S ',i,' = ',Som(i,n,A));

Writeln ('S ',m+1,' = ', A[m+1,m+1]);

End;

Combiner tous les cadres possibles de la matrice. Si une matrice de taille 7 x 7, elle ne contient que 7 Div 2 = 3 cadres

Le dernier cadre est celui du centre de la matrice.

**Procedure** Affiche (n: integer; A : MAT);

Var

i,j: integer;

Begin

For i := 1 To n Do

Begin

For j := 1 To n Do

Write (A[i,j]:5);

Writeln;

end

End;

{Programme principal}

**Begin**

Saisie (n);

Remp (n, A);

Affiche (n, A);

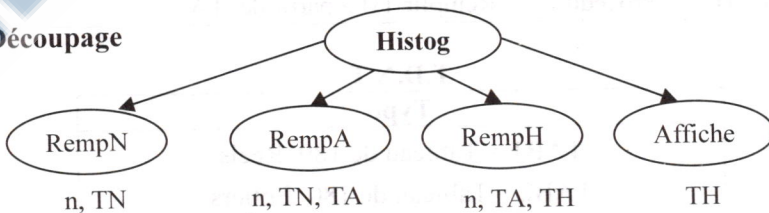
Sommes (n, A);

End.

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**Exercice n°10 :**

**I- Découpage**





- ✓ **Histog** : programme principal, qui permet d'afficher les résultats d'un examen sous forme d'un histogramme.
- ✓ **RempN** : procédure, permettant de saisir n et de remplir le tableau des notes TN.
- ✓ **RempA** : procédure, permettant de remplir le tableau TA contenant les arrondis des notes se trouvant dans TN.
- ✓ **RempH** : procédure, permettant de remplir le tableau TH contenant le nombre d'occurrences de chaque TA[i].
- ✓ **Affiche** : procédure, permettant d'afficher l'histogramme des notes.

## II- Analyse & Algorithme

\*\*\* Programme principal : Histog \*\*\*

### 1) Spécification du problème :

Résultat = Affiche (TH)

Traitements =

- ✓ TH = RempH (n, TA, TH)
- ✓ TA = RempA (n, TN, TA)
- ✓ n, TN = RempN (n, TN)
- ✓ RempH, RempA, RempN et Affiche = Procédures

www.BAC.org.tn  
Page BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

### T.D.O.U.G

Objet	Nature / Type	Rôle
Affiche	Procédure	Afficher le contenu du tableau TH.
n	Entier	Contenir le nombre de notes.
TN	TAB1	Contenir des notes d'un examen.
RempN	Procédure	Saisir n et remplir TN.
TA	TAB2	Contenir des arrondis des notes.
TH	TAB3	Contenir les occurrences de chaque TA[i].
RempA	Procédure	Remplir TA à partir de TN.
RempH	Procédure	Remplir TH à partir de TA.

### T.D.N.T

#### Type

TAB1 = Tableau de 180 Réels  
 TAB2 = Tableau de 180 Entiers  
 TAB3 = Tableau de 21 Entiers

2) **Algorithme :**

- 0) Début Histog
- 1) RempN (n, TN)
- 2) RempA (n, TN, TA)
- 3) RempH (n, TA, TH)
- 4) Affiche (TH)
- 5) Fin Histog



\*\*\* Procédure : RempN \*\*\*

1) **Spécification du problème :**

Résultat = n et TN

Traitements =

- ✓ n = Donnée avec  $n \in [50..180]$
- ✓ TN = Données
  - ↳  $TN[i] = \text{Donnée avec } T[i] \in [0..20]$
- ✓ i = Compteur allant de 1 à n
- ✓ n et T = Paramètres formels

**T.D.O.U.L**

Objet	Nature / Type	Rôle
i	Entier	Compteur.

2) **Algorithme :**

- 0) Procédure RempN (Var n : Entier ; Var TN : TAB1)
- 1) Répéter
  - Ecrire ("Saisir le nombre de notes : ")
  - Lire (n)
  - Jusqu'à n Dans [50..180]
- 2) Pour i de 1 à n Faire
  - Répéter
    - Ecrire (" Note n° ", i, " : ")
    - Lire (TN[i])
    - Jusqu'à (TN[i]>=0) Et (TN[i]<=20)
  - FinPour
- 3) Fin RempN

**\*\*\* Procédure : RempA \*\*\***

**1) Spécification du problème :**

Résultat = TA

Traitements =

- ✓ TA = [                    ]
- ↪ TA[i] = Arrondi(TN[i])
- ✓ i = Compteur allant de 1 à n
- ✓ n, TN, et TA = Paramètres formels

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**T.D.O.U.L**

Objet	Nature/Type	Rôle
i	Entier	Compteur

**2) Algorithme :**

- 0) Procédure RempA (n : Entier ; TN : TAB1; Var TA : TAB2)
- 1) Pour i de 1 à n Faire
  - TA[i] ← Arrondi (TN[i])
  - FinPour
- 2) Fin RempA

**\*\*\* Procédure : RempH \*\*\***

**1) Spécification du problème :**

Résultat = TH

Traitements =

- ✓ TH = [                    ]
- ↪ TH [TA[i]] ↗ +1 pour chaque TA[i]
- ↪ TH [i] ← 0
- ✓ i = Compteur allant de 0 à 20
- ✓ i = Compteur allant de 1 à n (2<sup>ème</sup> utilisation du compteur i)
- ✓ n, TA et TH = Paramètres formels

**T.D.O.U.L**

Objet	Nature / Type	Rôle
i	Entier	Compteur

**2) Algorithme :**

- 0) Procédure RempH (n : Entier ; TA : TAB2 ; Var TH: TAB3)
- 1) Pour i de 0 à 20 faire  
TH [i] ← 0  
FinPour
- 2) Pour i de 1 à n faire  
TH [TA[i]] ← TH [TA[i]] + 1  
FinPour
- 3) Fin RempH

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

**\*\*\* Procédure : Affiche \*\*\***

**1) Spécification du problème :**

Résultat = Aff

Traitements =

- ✓ Aff = [                          ]  
Afficher ("\*") un nombre de fois = TH[i] fois, il nous faut un 2<sup>ème</sup> compteur outre que i, soit j par exemple.
- ✓ j = Compteur allant de 1 à TH[i]
- ✓ i = Compteur allant de 0 à 20
- ✓ TH = Paramètre formel

**T.D.O.U.L**

Objet	Nature/Type	Rôle
j	Entier	Compteur
i	Entier	Compteur

**2) Algorithme :**

- 0) Procédure Affiche (TH: TAB3)
- 1) Pour i de 0 à 20 Faire  
Ecrire (i , " : ")  
Pour j de 1 à TH[i] Faire  
Ecrire ("\*")  
FinPour  
Retour\_à\_la\_ligne  
FinPour
- 2) Fin Affiche

Supposons que i = 0, l'ordinateur affiche "0 : " suivi d'un nombre de "\*" égal au nombre de notes ≈ 0, ce nombre de zéros = TH[0].



**Exercice n°11 :****Program** REMP\_MAT;

Uses WinCRT;

**Type**

MAT = Array [1..3 , 1..3] Of integer;

TAB = Array [1..9] Of integer;

**Var**

A: MAT;

TV, TL, TC : TAB;

n : integer;

**Procedure** Saisie (Var n : integer);**Begin**

Repeat

Write ('Le nombre de lignes/ colonnes :');

Readln (n);

Until n in [3, 6, 9];

**End;****Procedure** RempTV (n: integer ; Var T : TAB);**Var**

i : integer;

**Begin**

For i := 1 To n Do

Repeat

Write ('La valeur n°', i, ' :');

Readln (T[i]);

Until T[i] &lt;= 0;

**End;****Function** Verif (T: TAB ; i : integer) : Boolean;**Var**

V : Boolean;

nb, j: integer;

**Begin**

V:= False;

nb:=0;

j:=0;

Repeat

Inc (j);

If T[i] = T[j] Then Inc (nb);

Until (nb&gt; 3 ) Or (j = i) ;

www.BAC.org.tn  
Page: BAC-TUNISIE  
Tél: 25 361 197 / 53 371 502

Remplissage du tableau  
des valeurs TV.

Vérifier qu'un numéro de ligne est utilisé  
plus que 3 fois ou non.

La matrice est de 3 colonnes, alors une ligne  
est exploitée 3 fois au maximum.



V := nb <= 3 ;

Verif := V;

End;

V := nb <= 3 ;  
Si nb <= 3 : V ← Vrai  
Si nb > 3 : V ← Faux

**Procedure** RempTL (n: integer ; Var T : TAB);

Var

i: integer;

Begin

For i := 1 To n Do

Repeat

Write ('La valeur n° ', i, ' :');

Readln (T[i]);

Until (Verif (T,i)) And (T[i] in [1.. 3]);

End;

Remplissage du  
tableau des lignes TL.

**Function** Redond (T, TL : TAB ; i : integer) : Boolean;

Var

V : Boolean;

nb,j: integer;

Begin

V:= False;

nb:=0;

j:=0;

Repeat

Inc(j);

If (T[i] = T[j]) And (TL[i] = TL[j]) Then Inc(nb);

Until (nb > 1) Or (j=i) ;

V := nb = 1 ;

Redond := V;

End;

Un couple de numéro de ligne et de  
numéro de colonne (la référence d'une  
case dans une matrice) : (TL[i], TC[i])  
ne doit exister q'une seule fois!  
Une case de la matrice ne peut contenir  
qu'une seule valeur.

**Procedure** RempTC (n : integer ; Var T : TAB);

Var

i: integer;

Begin

For i := 1 To n Do

Repeat

Write ('La valeur n° ', i, ' :');

Readln (T[i]);

Until (Verif (T,i)) And (T[i] in [1..3]) And (Redond (T, TL, i));

End;

Remplissage du tableau  
des colonnes TC.

```

Procedure RempA (n : integer ; Var A : MAT ; TV, TL, TC : TAB);
Var i: integer;
Begin
  For i := 1 To n Do
    A[TL[i],TC[i]]:=TV[i]

```

Remplissage de la  
matrice A.

**End;**

```

Procedure Affiche (n : integer ; A : MAT);

```

```

Var i,j: integer;

```

```

Begin

```

```

  For i := 1 To 3 Do

```

```

    Begin

```

```

      For j := 1 To 3 Do

```

```

        Write (A[i,j]:5);

```

```

        Writeln;

```

```

      end;

```

```

End;

```

{Programme principal}

```

Begin

```

```

  Saisie (n);

```

```

  Writeln ('Remplissage du tableau des valeurs : ');

```

```

  RempTV (n, TV);

```

```

  Writeln ('Remplissage du tableau des lignes : ');

```

```

  RempTL (n, TL);

```

```

  Writeln ('Remplissage du tableau des colonnes : ');

```

```

  RempTC (n, TC);

```

```

  RempA (n, A, TV, TL, TC);

```

```

  Clrscr;

```

```

  Writeln ('Affichage du contenu de la matrice A ');

```

```

  Affiche (n, A);

```

```

End.

```



" أحمض و اتركه "

مثل شعبي تونسي