

FENNI Salah

# Exercices en Turbo Pascal



*Chebba*

*Le présent manuel est conforme au programme  
d'informatique de la 4<sup>ème</sup> année de  
l'enseignement secondaire relatif aux sections :  
Sciences Informatiques, Mathématiques,  
Sciences Expérimentales et Techniques.  
Ces exercices sont assez variés et tiennent compte  
des différents niveaux des élèves.*

# LES STRUCTURES SIMPLES

## Exercice 1

Compléter le tableau suivant par les valeurs finales des variables indiquées.

Instructions	Valeurs
A ← 6 + 2 * 5 div 3 - 10	A = .....
B ← (5>2) OU (8<8) ET Non (6=6)	B = .....
X ← Tronc (10.5)	X = .....
Y ← Arrondi (10.5)	Y = .....
W ← Aléa(6) + 4	W = .....
Valeur ("0025.38", Z, E)	Z = ..... E = .....
L ← Long (Sous_Chaine (ch, 1, 0))	L = .....
P ← Pos (Majus ("a"), "PASCAL")	P = .....
Ch ← "information" Efface (Ch, Long (Ch)-1, 3)	Ch = .....

## Exercice 2

a) Les variables N, P et Q sont entières et contiennent respectivement les valeurs 5, 7 et 3. Les expressions suivantes sont-elles correctes. Si oui, donnez leur type et leur valeur.

- N mod P \* Q
- N mod P div Q
- N = P Or N <= Q
- SUCC ('N' + 'P')

b) La variable C est de type caractère et contient la valeur 'e'. Les expressions suivantes sont-elles correctes. Si oui, donnez leur type et leur valeur.

- PRED (C) + 'e'
- ORD (C) + 2.5
- CHR (ORD (C) - 32)
- CHR (SUCC (ORD (C)))

## Exercice 3

Soient les déclarations Pascal suivantes :

```
VAR  A: Real;
     C: Char;
     D: String;
     E: Boolean;
     T: Array [1..20] of Integer;
```

Dans la case de chacune des affectations suivantes écrites en Pascal, mettre **V** si l'opération est permise et **F** sinon.

	A := 3E+2 ;
	C := upcase ('2') ;
	D := '3' + '2' ;
	C := D ;

	E := (C >= 'C') ;
	T [1] := -8 ;
	A := T [1] DIV 2 ;
	T [5] := T [1] / 2 ;

#### Exercice 4

Soient les déclarations suivantes, en Pascal :

**Type** jour\_semaine = (lun, mar, mer, jeu, ven, sam, dim) ;

**Const** Nom = 'ali' ;

**Var** Test : boolean ; A, B : integer ; Jour : jour\_semaine ;

Pour chaque opération de lecture ou d'écriture, mettre **V** si l'opération est possible et **F** dans le cas contraire.

[ ] Readln (A) ;	[ ] Readln ('A') ;	[ ] Readln (A+B) ;
[ ] Readln (45) ;	[ ] Readln ('A =', A);	[ ] Readln (Nom) ;
[ ] Readln (Test) ;	[ ] Readln (Jour) ;	[ ] Writeln (Jour) ;
[ ] Writeln ('A = ', A) ;	[ ] Writeln (A:6:2) ;	[ ] Writeln (45) ;
[ ] Writeln (5 mod 7 div 2) ;	[ ] Writeln (Nom[2]) ;	[ ] Writeln (A+B, Nom, Test);

#### Exercice 5

Ecrire un algorithme puis la traduction en Pascal d'un programme intitulé **Sortie\_inverse**, qui saisit trois nombres dans un ordre donné et les affiche dans l'ordre opposé à l'entrée.

#### Exercice 6

Ecrire un algorithme puis la traduction en Pascal d'un programme intitulé **Cylindre**, qui calcule et affiche le volume d'un cylindre après saisie son rayon R et sa hauteur H.

#### Exercice 7

Ecrire un algorithme puis la traduction en Pascal d'un programme **Surface\_Rectangle**, qui calcule la surface d'un rectangle de dimensions données et affiche le résultat sous la forme suivante : "La surface du rectangle dont la longueur mesure .... m et la largeur mesure .... m, a une surface égale à .... mètres carrés".

#### Exercice 8

Ecrire un algorithme puis la traduction en Pascal d'un programme **Piscine**, qui lit les dimensions d'une piscine, et affiche son volume et la quantité d'eau nécessaire pour la remplir.

#### Exercice 9

Ecrire un algorithme puis la traduction en Pascal d'un programme **Trapèze**, qui lit les dimensions d'un trapèze et affiche sa surface.

#### Exercice 10

Ecrire un algorithme puis la traduction en Pascal d'un programme intitulé **Permut**, qui fait la permutation de deux variables A et B.

#### Exercice 11

Proposer une marche à suivre qui fait, une **permutation circulaire** à droite, des valeurs de trois variables A, B et C. Par exemple : à partir de (A, B, C) = (10, 25, 4), on passe à (A, B, C) = (4, 10, 25).

#### Exercice 12

Proposer une marche à suivre qui fait la **permutation** de deux variables numériques X et Y, sans faire appel à aucune variable intermédiaire.

#### Exercice 13

Ecrire un algorithme puis la traduction en Pascal d'un programme intitulé **Division**, qui fait calculer et afficher le quotient et le reste de la division euclidienne de A par B.

#### Exercice 14

Ecrire un algorithme puis la traduction en Pascal d'un programme qui lit une **température** en degrés Celsius et affiche son équivalent en Fahrenheit.

### Exercice 15

Ecrire un algorithme puis la traduction en Pascal d'un programme permettant de déterminer et d'afficher la conversion en **mile marin** d'une distance mesurée en kilomètre.

### Exercice 16

Ecrire un algorithme puis la traduction en Pascal d'un programme qui permet de convertir et d'afficher en octets, kilo octets, méga octets et giga octets un nombre donné en **bits**.

### Exercice 17

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui convertit en heures, minutes et secondes, une **durée T** donnée en secondes. Il affiche le résultat sous la forme digitale comme celle d'une montre électronique (hh : mn : ss).

### Exercice 18

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui **conjugue** un verbe du premier groupe au futur simple. On ne traite pas les verbes irréguliers.

### Exercice 19

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui calcule et affiche, l'intérêt et la valeur acquise par une somme placée en épargne pendant 5 ans à **intérêt simple**.

### Exercice 20

On sait qu'avec un réservoir de L litres, une voiture a parcouru Y km. Ecrire un algorithme puis la traduction en Pascal d'un programme, qui fait lire les données nécessaires et fait calculer et afficher le taux de **consommation** aux 100 km de cette voiture.

### Exercice 21

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui affiche la résistance équivalente à trois résistances R1, R2, R3.

- si les résistances sont branchées en série :  $R_{ser} = R_1 + R_2 + R_3$

- si les résistances sont branchées en parallèle :  $1/R_{par} = 1/R_1 + 1/R_2 + 1/R_3$

### Exercice 22

On se propose de saisir un entier N de trois chiffres non nuls, de déterminer et d'afficher tous les nombres qui peuvent être formés par les chiffres de N.

Exemple : pour N=427

Les nombres formés par les chiffres de N sont : 427, 472, 724, 742, 247, 274.

### Exercice 23

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui fait entrer deux entiers A et B et fait calculer et afficher leur **inf** et leur **sup**. Utiliser une formule mathématique donnant directement le résultat.

### Exercice 24

On se propose d'écrire un programme intitulé **IMAGE** qui calcule le nombre de pixels et le poids d'une image numérique en méga octet (MO), d'une dimension (hauteur, largeur) exprimées en pouces et d'une résolution en dpi (points par pouce).

**NB.**

- La résolution indique le nombre de points sur une unité de longueur de l'image.

- Le nombre de pixels dans une image = hauteur \* résolution \* longueur \* résolution

- L'image est codée en 24 bits/pixel ou 48 bits/pixel c'est à dire 8 ou 16 bits par canal R (rouge), V (vert) et B (bleu).

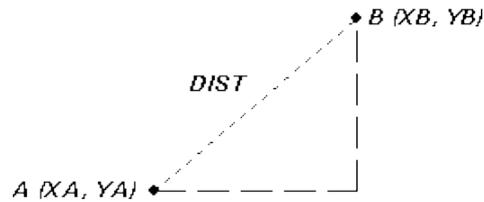
- Le nombre total d'octets dans l'image (poids) = Nombre de pixels \* nombre d'octets par pixel.

### Exercice 25

Écrire en Pascal le programme intitulé **DATE**, qui initialise la date maintenue par le système d'exploitation, en utilisant la procédure SetDate (...) de la bibliothèque WinDos.

### Exercice 26

Ecrire un programme Pascal, qui calcule et affiche la distance DIST entre deux points A et B du plan dont les coordonnées (XA, YA) et (XB, YB) sont entrées au clavier comme entiers.



### Exercice 27

- a) Ecrire un programme Pascal qui calcule le prix TTC d'un article à partir du prix net et du pourcentage de TVA à ajouter. Utiliser la formule suivante :

$$PTTC = PNET + PNET \cdot \frac{TVA}{100}$$

- b) Ecrire un programme Pascal qui calcule le prix net d'un article à partir du prix TTC et du pourcentage de TVA qui a été ajoutée.  
(Déduire la formule du calcul de celle indiquée ci-dessus)

### Exercice 28

Deux trains, distants de 60 km, roulent l'un vers l'autre sur le même itinéraire.

- Le train A avance à une vitesse de 70 km/h.
- Le train B avance à une vitesse de 55 km/h.

Ecrire un programme intitulé **RENCONTRE** qui permet de calculer le temps où les deux trains vont se rencontrer?

# LES STRUCTURES CONDITIONNELLES

## Exercice 1

Ecrire un programme Pascal intitulé **MIN2** qui affiche le plus petit de deux nombres A et B.

## Exercice 2

Ecrire un programme Pascal intitulé **MAX3** permettant d'obtenir le maximum de trois nombres donnés.

## Exercice 3

Ecrire un programme Pascal intitulé **RACINE** qui affiche la racine carrée d'un réel donné.

## Exercice 4

Ecrire un programme Pascal intitulé **Aire\_triangle** qui calcule et affiche l'aire d'un triangle dont il faut entrer les longueurs des trois côtés. Utilisez la formule :

$$S^2 = P(P-A)(P-B)(P-C)$$

où A, B, C sont les longueurs des trois côtés et P le demi-périmètre du triangle.

## Exercice 5

Sans utiliser la fonction prédéfinie ABS, écrire un programme Pascal intitulé **ABS\_DIFF** qui détermine et affiche la valeur absolue de (a-b).

## Exercice 6

Ecrire un programme Pascal intitulé **PARITE** qui saisit un nombre entier et détecte si ce nombre est pair ou impair.

## Exercice 7

Ecrire un programme Pascal intitulé **chez\_la\_fourmi**, permettant de simuler un jeu entre deux joueurs régi par le principe suivant :

Deux joueurs A et B se cachent la main droite derrière le dos. Les deux joueurs se montrent la main droite en même temps, en choisissant de tendre un certain nombre de doigts (de 1 à 5). Si la somme des nombres de doigts montrés est paire, le premier joueur a gagné, sinon c'est le second.

## Exercice 8

Ecrire un programme Pascal intitulé **INVITE**, qui saisit le prénom d'une personne et son titre (Mr, Mlle ou Mme), puis affiche l'invite suivante :

"Titre"foulen", soyez le bienvenu"      ou      "Titre"foulena", soyez la bienvenue"

## Exercice 9

Ecrire un programme Pascal intitulé **SUP\_INF** qui saisit deux entiers A et B, teste si A est supérieur, inférieur ou égale à B puis affiche le résultat.

## Exercice 10

Ecrire un programme Pascal intitulé **ORDRE** qui ordonne dans le sens croissant, en échangeant leur valeur si nécessaire, deux entiers saisis au clavier.

## Exercice 11

Ecrire un programme Pascal intitulé **TRI**, qui fait lire trois entiers A, B et C, les permute de façon à les classer par ordre croissant puis affiche le résultat.

## Exercice 12

Ecrire un programme Pascal intitulé **EQUA\_1D**, qui fait résoudre dans IR l'équation **ax+b=0** pour tout couple de réels (a, b) donné.

### Exercice 13

Ecrire un programme Pascal intitulé **EQUA\_2D**, qui fait résoudre dans IR l'équation  $ax^2+bx+c=0$  pour tout triplet de réels (a, b, c) donné.

### Exercice 14

Ecrire un programme Pascal intitulé **INEQUATION**, qui fait résoudre dans IR l'inéquation  $ax+b<0$  pour tout couple de réels (a, b) donné.

### Exercice 15

Ecrire un programme Pascal intitulé **TOUCHE** qui affiche selon le cas, la nature du caractère (consonne, voyelle, chiffre ou symbole) correspondant à une touche saisie.

### Exercice 16

Ecrire un programme Pascal intitulé **CALCULETTE**, qui fait lire dans cet ordre : un réel, un opérateur arithmétique (+, -, \*, /) et un autre réel. A chaque opérateur valide correspond une opération arithmétique qu'il faut exécuter et afficher le résultat ou un message d'erreur, le cas échéant.

### Exercice 17

Ecrire un programme Pascal intitulé **BULLETIN**, qui permet de saisir la moyenne annuelle d'un élève et affiche la décision du conseil de classe, la mention adéquate dans le cas où l'élève est admis :

$18 \leq \text{moy}$	Excellent
$16 \leq \text{moy} < 18$	Très Bien
$14 \leq \text{moy} < 16$	Bien
$12 \leq \text{moy} < 14$	Assez Bien
$10 \leq \text{moy} < 12$	Passable
$\text{Moy} < 10$	Redouble

### Exercice 18

Ecrire un programme Pascal intitulé **JOURS** qui saisit le numéro du mois et affiche le nombre de jours de ce mois. Pour le cas de février, on lit l'année, si l'année est bissextile, le nombre de jours est 29, sinon c'est 28 jours.

### Exercice 19

Ecrire un programme Pascal intitulé **DATE**, qui saisit une date sous la forme jj/mm/aaaa (chaîne de 10 caractères) et l'affiche décodée (jj Mois aaaa).  
Exemple : 10/05/2007 donne 10 Mai 2007.

### Exercice 20

Ecrire un programme Pascal intitulé **ANCIENNETE**, qui fait lire une date initiale JI/MI/AI et une date finale JF/MF/AF et qui fait calculer et afficher la durée (exprimée en années, mois et jours) qui les sépare.

### Exercice 21

Ecrire un programme Pascal intitulé **LENDEMAIN**, qui saisit une date quelconque (jj, mm, aa) et affiche la date du lendemain.

### Exercice 22

Ecrire un programme Pascal intitulé **TOUTE\_LETTR**, qui traduit en toutes lettres un entier naturel donné entre 0 et 99.

Exemples :  
10 → Dix  
21 → Vingt et un  
85 → Quatre vingt cinq

### Exercice 23

Ecrire un programme Pascal intitulé **Jour\_Semaine**, qui permet de déterminer le jour de la semaine correspondant une date donnée.

Exemple : le 10/05/2007 est un jeudi.

### Exercice 24

Ecrire un programme Pascal qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe du produit de A et B sans faire la multiplication.

### Exercice 25

Ecrire un programme Pascal qui lit deux valeurs entières (A et B) au clavier et qui affiche le signe de la somme de A et B sans faire l'addition.

### Exercice 26

Considérez la séquence d'instructions suivante:

```
Si (A>B)
  Alors Ecrire ("premier choix")
  Sinon Si (A>10)
    Alors Ecrire ("deuxième choix")
  Fin Si
Si (B<10)
  Alors Ecrire ("troisième choix")
  Sinon Ecrire ("quatrième choix")
Fin Si
```

Déterminez les réponses sur l'écran pour chacun des couples de nombres suivants :

- A=10 et B=5
- A=5 et B=5
- A=5 et B=10
- A=10 et B=10
- A=20 et B=10
- A=20 et B=20

### Exercice 27

Considérez la séquence d'instructions suivante:

```
Si (A>B)
  Alors Si (A>10)
    Alors Ecrire ("premier choix")
    Sinon Si (B<10)
      Alors Ecrire ("deuxième choix")
      Sinon Si (A=B)
        Alors Ecrire ("troisième choix")
        Sinon Ecrire ("quatrième choix")
  Fin Si
```

1. Pour quelles valeurs de A et B obtient-on les résultats : premier choix, deuxième choix, ... sur l'écran ?
2. Pour quelles valeurs de A et B n'obtient-on pas de réponse sur l'écran ?

### Exercice 28

Écrire un programme Pascal qui permet de saisir un temps (heure et minute) lui ajoute 5 minutes puis l'affiche.

**Exemples :**

```
heure : 10  minute : 20  après 5 minutes : 10:25
heure : 10  minute: 57  après 5 minutes : 11: 02
heure : 23  minute: 55  après 5 minutes : 00:00
```

### **Exercice 29**

Un nombre de **Niven** est un entier qui est divisible par la somme de ses chiffres.

**Exemple :** 112 est un nombre Niven.

*En effet 112 est divisible par la somme de ses chiffres qui est 4 ( $1+1+2=4$ )*

Ecrire un programme Pascal, qui permet de lire un entier **N** de trois chiffres, puis de vérifier et d'afficher s'il est un nombre de Niven ou non.

Fenni Salah

# LES STRUCTURES ITERATIVES

## Exercice 1

Ecrire un programme Pascal qui affiche **l'alphabet** en majuscule, d'abord à l'endroit, puis à l'envers, après un passage à la ligne.

## Exercice 2

Ecrire un programme Pascal qui affiche la **table de multiplication par 3**, pour les entiers de 1 à 10.

## Exercice 3

Ecrire un programme Pascal qui calcule et affiche la **somme** et le **produit**, des 20 premiers entiers (de 1 à 20).

## Exercice 4

Considérons la suite définie par la relation :  $U_{n+1} = U_n + 3$  et  $U_1 = 2$   
On veut calculer la somme de 100 premiers termes de cette suite. Donner trois solutions distinctes en utilisant les boucles : **POUR**, **TANT QUE** et **REPETER**.

## Exercice 5

Ecrire un programme Pascal qui affiche la **table de Pythagore** (table de multiplication) pour les 9 premiers nombres entiers.

## Exercice 6

Ecrire cinq programmes Pascal permettant d'afficher les triangles d'étoiles suivants :

**Exemple : (pour nL=5)**

```

      *          *          *          *          *          *
     ***        *          **         ****         *****
    *****    *          ***         *****       *
   *          *          *          *          *          *
  *          *          *          *          *          *
 *          *          *          *          *          *

```

Ecrire deux programmes Pascal permettant d'afficher les pyramides de nombres suivants :

**Exemple : (pour nL=6)**

```

           1
          232
         34543
        4567654
       567898765
      67890109876

```

```

           1
          121
         12321
        1234321
       123454321
      12345654321

```

## Exercice 7

Ecrire un programme Pascal qui permet de calculer et afficher la **moyenne** de notes fournies au clavier avec un "dialogue" se présentant ainsi :

```

Combien de notes : 4
Note 1 : 12
Note 2 : 15.25
Note 3 : 13.5
Note 4 : 8.5
Moyenne de ces 4 notes : 12.31

```

## Exercice 8

Ecrire un programme Pascal faisant calculer et afficher le **factoriel** d'un entier naturel N donné. Sachant que (pour  $N > 0$ ) :  $N! = N \times (N-1) \times (N-2) \times \dots \times 3 \times 2 \times 1$ .

### Exercice 9

Le "jeu du nombre mystérieux" consiste à jouer contre l'ordinateur comme suit : L'ordinateur choisit, au hasard, un entier entre 1 et 100 et on doit le trouver en 7 essais au maximum grâce aux indices "C'est grand" et "C'est petit". Au moment venu on affichera "Bravo vous avez gagné !!". Si le nombre d'essais est atteint sans trouver le nombre mystérieux, le programme affichera alors "Perdu, le nombre cherché est : ", suivie du nombre à trouver. Ecrire un programme Pascal pour s'amuser avec l'ordinateur.

### Exercice 10

Ecrire un programme Pascal qui permet de saisir un entier  $n$ , de déterminer et d'afficher tous ses **chiffres qui le divisent**.

Exemple : pour  $n = 2376$  alors les chiffres 2, 3 et 6 seront affichés.

### Exercice 11

Ecrire un programme Pascal **SOM\_15**, qui détermine toutes les manières possibles d'obtenir un total de 15 en ajoutant trois entiers choisis entre 1 et 9.

### Exercice 12

On se propose d'afficher un **histogramme** à l'aide des lettres A, B et C comme celui de l'exemple ci-dessous.

Ecrire un programme Pascal qui saisit le nombre de A, le nombre de B et le nombre de C puis affiche l'histogramme correspondant. Les nombres sont des entiers naturels inférieurs ou égaux à 15.

Exemple d'exécution : Entrer trois nombres entiers compris entre 0 et 15 : 4,7, 2

```

          B
          B
          B
    A      B
    A      B
    A      B   C
    A      B   C
  
```

### Exercice 13

Ecrire un programme Pascal **SOM\_CHIFFRES**, qui permet de déterminer la somme des chiffres d'un nombre entier donné (exemple : pour  $N = 25418$ , on aura  $2+5+4+1+8 = 20$ ).

### Exercice 14

Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses trois chiffres.

Exemple : 153 est cubique car  $153 = 1^3 + 5^3 + 3^3$

Ecrire un programme Pascal **NBR\_CUBE** qui cherche et affiche tous les entiers cubiques de trois chiffres.

### Exercice 15

Ecrire un programme Pascal **SOMME**, qui calcule et affiche les sommes suivantes :

- $S_1 = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$
- $S_2 = 1 + 1/3 + 1/5 + \dots + 1/n$       Avec  $n$ , un entier naturel impair donné.
- $S_3 = -1/2 - 1/4 - 1/6 - \dots - 1/(n-1)$

### Exercice 16

On donne un entier naturel  $n$  strictement positif et on définit la suite de **Syracuse** par :

$$\begin{cases} S_0 = n \\ S_k = S_{k-1} \text{ div } 2 & \text{Si } S_{k-1} \text{ est pair} \\ S_k = 3S_{k-1} + 1 & \text{Si } S_{k-1} \text{ est impair} \end{cases}$$

Ecrire un programme Pascal qui fait afficher les 50 premiers termes de cette suite.

### Exercice 17

Si nous lançons 3 dés, le total des points est compris entre 3 et 18. Quelle est la probabilité d'avoir un total de 12 ?

Ecrire un programme Pascal **PROBABILITE**, qui répond à cette question en simulant 100 lancers successifs.

### Exercice 18

Un nombre réel X et un nombre entier N étant donné, proposer un programme Pascal qui fait calculer  $X^N$ . Etudier tous les cas possibles (N positive ou négative).

### Exercice 19

Ecrire un programme Pascal qui saisit deux entiers X et Y, et fait calculer l'expression  $S=X*Y$  par **additions successives** ( $X*Y=X+X+X+\dots$ ). Choisir la somme qui fait intervenir le minimum de termes.

### Exercice 20

Pour un entier naturel N donné. Ecrire un programme Pascal qui fait calculer et afficher la suite :

$$S = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

### Exercice 21

On remarque que :  
12 x 42 = 21 x 24  
12 x 63 = 21 x 36  
12 x 84 = 21 x 48

Il y a 14 produits qui vérifient la propriété :  $(10a + b)(10c + d) = (10b + a)(10d + c)$ , où a est différent de b et c est différent de d. Ecrire un programme Pascal qui fait sortir tous ces entiers.

### Exercice 22

On se propose de déterminer une **valeur approchée de  $\Pi$**  par la méthode de Wallis, définie par la formule suivante :

$$\frac{\pi}{2} = \frac{2}{1} \times \frac{2}{3} \times \frac{4}{3} \times \frac{4}{5} \times \frac{6}{5} \times \frac{6}{7} \times \frac{8}{7} \times \frac{8}{9} \times \dots$$

Ecrire un programme Pascal qui utilise la formule ci-dessus pour déterminer et afficher une valeur approchée de  $\Pi$  à  $10^{-8}$  près.

### Exercice 23

On se propose de déterminer le **PGCD** (Plus Grand Commun Diviseur) de deux entiers positifs non nuls A et B en utilisant l'algorithme d'**Euclide** :

Sachant que  $\text{PGCD}(A, B) = \text{PGCD}(B, R)$ , avec  $R = A \bmod B$ .

Tant que le reste R est non nul, on remplace A par B et B par R. Le dernier reste R non nul est alors le PGCD des deux nombres.

Exemple :  $\text{PGCD}(32, 12) = \text{PGCD}(12, 8) = \text{PGCD}(8, 4) = \text{PGCD}(4, 0) = 4$ .

### Exercice 24

On se propose de déterminer le **PGCD** (Plus Grand Commun Diviseur) de deux entiers positifs non nuls A et B en utilisant la **méthode de la différence** :

Tant que  $(a \neq b)$  on répète la recherche :

$\text{PGCD}(a, b) = \text{PGCD}(a-b, b)$  si  $a > b$ , sinon  $\text{PGCD}(a, b) = \text{PGCD}(a, b-a)$

Exemple :  $\text{PGCD}(10, 16) = \text{PGCD}(10, 6) = \text{PGCD}(4, 6) = \text{PGCD}(4, 2) = \text{PGCD}(2, 2) = 2$ .

### Exercice 25

On se propose de déterminer le **PPCM** (Plus Petit Commun Multiple) de deux entiers positifs non nuls M et N.

### Exercice 26

#### **Suite de Fibonacci**

Les nombres de **Fibonacci** sont donnés par la récurrence :

$$F_n = F_{n-2} + F_{n-1} \quad \text{avec} \quad F_0 = F_1 = 1.$$

Ecrire un programme Pascal qui affiche les 20 premiers nombres de Fibonacci.

### Exercice 27

Un entier supérieur à 1 est dit **premier** s'il admet exactement deux diviseurs : 1 et lui-même.

Ecrire un programme Pascal qui cherche et affiche tous les nombres premiers  $\leq$  à 400.

### Exercice 28

Un nombre entier naturel est dit **parfait** s'il est égal à la somme de ses diviseurs sauf lui-même. Ecrire un programme Pascal permettant de déterminer et d'afficher tous les nombres parfaits compris entre a et b ( $2 < a < b$ ).

### Exercice 29

Deux entiers m et n sont dit **amis** si et seulement si la somme des diviseurs de m sauf lui-même est égale à n et la somme des diviseurs de n sauf lui-même est égale à m.

Ecrire un programme Pascal permettant de déterminer et d'afficher tous les nombres amis compris entre 1 et 1000.

### Exercice 30

Ecrire un programme Pascal permettant de décomposer un entier N donné ( $2 \leq N \leq 100$ ) en produit de **facteurs premiers** et d'afficher N et le produit de ses facteurs trouvés.

Exemple : Si n = 60 alors on affiche  $60 = 2 * 2 * 3 * 5$ .

### Exercice 31

Sachant que  $\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$  tel que  $x \in [0, 2\pi]$ .

Ecrire un programme Pascal qui permet d'afficher **sin(x)** en utilisant la formule ci-dessus. Le calcul s'arrête quand la différence entre deux termes consécutifs devient  $\leq$  à  $10^{-4}$ . La dernière somme calculée est une valeur approchée de sin(x).

### Exercice 32

Ecrire un programme Pascal qui donne une approximation de :

$$e^x \simeq 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} \quad \text{à epsilon près.}$$

Le calcul s'arrête lorsque  $\left| \frac{x^n}{n!} \right| \leq \text{epsilon}$ . X et Epsilon sont des données.

### Exercice 33

Soit la somme  $S_n$  suivante :

$$S_n = 1 + \frac{3}{2^2} + \frac{5}{3^3} + \frac{7}{4^4} + \dots + \frac{2n-1}{n^n}$$

Ecrire un programme Pascal intitulé **SOMME** permettant de calculer et d'afficher la somme  $S_n$  pour un entier n positif donné en utilisant la formule ci-dessus.

### Exercice 34

Ecrire un programme Pascal intitulé **COMBINAISON**, qui lit deux entiers naturels n et p avec ( $0 < p < n$ ), puis calcule et affiche le nombre de combinaisons de p objets parmi n.

On rappelle que :  $C_n^p = \frac{n!}{p!(n-p)!}$

### Exercice 35

Un entier de n chiffres ( $1 < n < 9$ ) est dit **bien ordonné** si ses chiffres forment, de gauche à droite, une suite strictement croissante.

Exemple :

L'entier de 3 chiffres, 147 est bien ordonné car  $1 < 4 < 7$

L'entier de 4 chiffres, 1265 n'est pas bien ordonné car  $6 > 5$

Ecrire un programme Pascal qui saisit un entier n ( $1 < n < 9$ ) et fait sortir tous les entiers bien ordonnés de n chiffres et le nombre total de ces entiers.

### Exercice 36

Ecrire un programme Pascal qui affiche tous les entiers positifs impairs inférieurs à 100, en omettant les nombres divisibles par 7. L'affichage doit être sur 5 colonnes comme l'aspect suivant :

1	3	5	9	11
13	17	19	23	25
27	29	31	33	37
.....	.....	.....	.....	.....

### Exercice 37

On se propose d'écrire un programme Pascal qui permet de déterminer et d'afficher toutes les représentations sous forme de sommes d'entiers consécutifs d'un entier n donné.

**Exemple d'exécution :**

Entrer un entier : 45

Les sommes consécutives sont :

$$45 = 1+2+3+4+5+6+7+8+9$$

$$45 = 5+6+7+8+9+10$$

$$45 = 7+8+9+10+11$$

$$45 = 14+15+16$$

$$45 = 22+23$$

### Exercice 38

En mathématiques, un entier N est un **carré parfait** s'il existe un entier k tel que  $N = k^2$ .

Ecrire un programme Pascal permettant de déterminer et d'afficher tous les carrés parfaits compris entre 1 et 9999.

### Exercice 39

Considérons un échiquier classique de 64 cases disposées en un carré de 8 lignes composées de 8 cases. On demande d'écrire un programme **REINE** permettant de lire les coordonnées (x, y) définissant la position de la reine, d'afficher la lettre **R** à la position correspondante et une \* (astérisque) à chacune case accessible.

Exemple :

Pour x=2 et y=5 on aura l'affichage :

```

      * * *
    * * * * R * * *
      * * *
        * * *
          * * *
            * * *
              * * *

```

### Exercice 40

**PGCD et PPCM de deux entiers :**

La méthode la plus simple est de chercher le premier multiple de a qui soit divisible par b, sachant que le produit du pgcd par le ppcm est ab, on a les deux en même temps.

### Exercice 41

Soient A et B deux entiers naturels quelconques non nuls.

On a:  $PGCD(A, B) * PPCM(A, B) = A * B$

Ecrire un programme Pascal qui permet de calculer  $A * B$  en utilisant cette propriété.

#### Exercice 42

Soient a et b deux réels quelconques. Pour tout entier impair n, supérieur ou égal à 3, on a :  
 $a^n + b^n = (a+b) (a^{n-1} - a^{n-2}b + \dots - ab^{n-2} + b^{n-1})$

Ecrire un programme Pascal qui permet de calculer  $a^n + b^n$ .

#### Exercice 43

L'entier naturel X est **divisible par 11** si et seulement si la somme de ses chiffres :

$x_0 - x_1 + x_2 - x_3 + \dots + (-1)^n x_n$  est divisible par 11.

**Exemple** : Si X = 6182319

La somme :  $9-1+3-2+8-1+6 = 22$  Or 22 est divisible par 11, donc 6182319 l'est aussi.

Ecrire un programme Pascal qui permet de vérifier cette règle de divisibilité.

#### Exercice 44

Sachant que :  $C_n^p = \frac{n!}{p!(n-p)!} = \frac{n \times (n-1) \times \dots \times (n-p+1)}{p!}$

Ecrire un programme Pascal qui permet de calculer la somme :

$$1 - (C_{2n}^1)^2 + (C_{2n}^2)^2 - \dots + (-1)^{2n} (C_{2n}^{2n})^2$$

#### Exercice 45

Un entier naturel est **divisible par 7** si la différence entre le nombre de dizaines et 2 fois le chiffre des unités est divisible par 7.

**Exemple** : Pour vérifier si 17381 est divisible par 7, on fait :

$$1738 - (2 \times 1) = 1736 ; 173 - (2 \times 6) = 161 ; 16 - (2 \times 1) = 14$$

Or 14 est divisible par 7, donc 17381 l'est aussi !

Un entier naturel est **divisible par 13** si l'addition du nombre de dizaines et 4 fois le chiffre des unités est divisible par 13.

**Exemple** : Pour vérifier si 8541 est divisible par 13, on fait :

$$854 + (4 \times 1) = 858 ; 85 + (4 \times 8) = 117 ; 11 + (4 \times 7) = 39$$

Or 39 est divisible par 13, donc 8541 l'est aussi !

Ecrire un programme Pascal permettant d'afficher sur l'écran :

- D'abord, les 100 premiers entiers divisibles par 7
- Ensuite, les 100 premiers entiers divisibles par 13.

#### Exercice 46

Un entier positif K à n chiffres est dit nombre de **Kaprekar** si lorsqu'on élève K au carré, la somme du nombre composé des n chiffres de droite au nombre composé des n ou n-1 chiffres de gauche redonne le nombre d'origine.

**Exemples** : 9, 45 et 297 sont des nombres de Kaprekar :

$$9^2 = 81 \text{ et } 1+8=9 ; 45^2 = 2025 \text{ et } 25+20=45 ; 297^2 = 88209 \text{ et } 209+88 = 297$$

Ecrire un programme Pascal qui permet d'afficher tous les nombres de KAPREKAR inférieurs ou égaux à 1000 en utilisant la méthode décrite ci-dessus.

#### Exercice 47

Un nombre premier N est dit circulaire s'il vérifie la propriété suivante : chacune des rotations de ses chiffres d'un élément vers la droite, forme à son tour un nombre premier.

**Exemples** :

- Si **N=719**, N est un nombre premier circulaire car **719**, **971** et **197** sont des nombres premiers, avec :
  - **971** est le nombre obtenu après une rotation des chiffres de **719** d'un élément vers la droite.
  - **197** est le nombre obtenu après une rotation des chiffres de **971** d'un élément vers la droite.
- Si **N=23**, N n'est pas un nombre premier circulaire car il est premier mais **32** ne l'est pas.
- Si **N=6102**, N n'est pas un nombre premier circulaire car il n'est premier.

On veut chercher tous les nombres circulaires se trouvant dans l'intervalle [p, q] (avec p et q sont deux entiers donnés tels que  $10 < p < q < 20000$ ).

#### Exercice 48 Fractions égyptiennes

Toute fraction peut s'écrire comme une somme de fractions ayant 1 comme numérateur. Cette décomposition est appelée décomposition en fractions égyptiennes. En voici un exemple :

$$\frac{7}{8} = \frac{1}{2} + \frac{1}{3} + \frac{1}{24}$$

Ecrire un programme Pascal prenant en entrée le numérateur N et le dénominateur D d'une fraction et affiche sa décomposition en fraction égyptiennes.

#### Exercice 49

Un nombre est dit **super premier** s'il est premier et si, en supprimant des chiffres à partir de sa droite, le nombre restant est aussi premier.

**Exemple** : Le nombre 59399 est super premier car les nombres 59399, 5939, 593, 59 et 5 sont tous premiers.

Ecrire un programme Pascal qui permet de :

- Saisir un entier n tel que  $40000 < n < 100000$ ,
- Chercher tous les nombres premiers inférieurs ou égaux à n, les afficher à raison d'un nombre par ligne en mentionnant devant chaque nombre super premier la note « super premier ».

#### Exercice 50

On se propose d'écrire un programme Pascal intitulé **conversion** permettant de saisir un nombre binaire de longueur minimale 2 puis de le convertir en décimal. Pour ce faire, on multiplie la valeur de chaque bit \* 2 puissance «le poids du bits (ordre-1)».

**N.B** : On rappelle qu'un nombre binaire est composé uniquement par des 0 et des 1.

**Exemple** :

$$(1001101)_2 = 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 64 + 0 + 0 + 8 + 4 + 0 + 1 = (77)_{10}.$$

#### Exercice 51 Jeu « des allumettes »

Ecrire un programme Pascal permettant de simuler le jeu suivant :

Au départ on dispose quelques allumettes sur une table. Chaque joueur peut choisir de prendre 1, 2 ou 3 allumettes en même temps. Le perdant est celui qui retire la dernière allumette.

#### Exercice 52

Un nombre colombien, ou auto-nombre, est un entier naturel qui, dans une base donnée, ne peut pas s'écrire sous la forme d'un nombre ajouté à la somme des chiffres de ce nombre.

Exemples :

- 23 n'est pas un nombre colombien, puisqu'il peut être généré par la somme de 16 et de ses chiffres, c'est-à-dire,  $23 = 16 + 1 + 6$ .
- 20 est un nombre colombien car il n'existe pas une telle somme pour 20.

Ecrire un programme Pascal qui permet d'afficher tous les nombres colombiens < 1000.

#### Exercice 53

Un **nombre Vampire** est un nombre qui est égal à un produit de ses chiffres.

**Exemple** :  $126 = 21 \times 6$

Ecrire un programme Pascal qui permet de déterminer tous les nombres Vampires de trois chiffres.

#### Exercice 54

Le **PGCD** de deux nombres entiers **M** et **N** supérieurs ou égaux à 2 a pour décomposition en facteurs premiers le produit des facteurs premiers apparaissant à la fois dans la décomposition de **M** et de **N** munis du plus petit des exposants trouvés dans la décomposition de **M** et de **N**.

Ainsi, si  $M = 2^3 \times 3^4 \times 5^2 \times 7$  et  $N = 2^2 \times 3^5 \times 7^3 \times 11$  alors  $\text{pgcd}(m, n) = 2^2 \times 3^4 \times 7$

#### Exercice 55

Le **PPCM** de deux nombres entiers **M** et **N** supérieurs ou égaux à 2 a pour décomposition en facteurs premiers le produit des facteurs premiers apparaissant dans **M** ou dans **N** munis du plus grand des exposants trouvés dans la décomposition de **M** et de **N**.

Ainsi, si  $M = 2^3 \times 3^4 \times 5^2 \times 7$  et  $N = 2^2 \times 3^5 \times 7^3 \times 11$  alors  $\text{ppcm}(m, n) = 2^3 \times 3^5 \times 5^2 \times 7^3 \times 11$

**Exercice 56**

Ecrire un programme qui permet de calculer et d'obtenir l'affichage suivant :

```

1 * 8 + 1 = 9
12 * 8 + 2 = 98
123 * 8 + 3 = 987
1234 * 8 + 4 = 9876
12345 * 8 + 5 = 98765
123456 * 8 + 6 = 987654
1234567 * 8 + 7 = 9876543
12345678 * 8 + 8 = 98765432
123456789 * 8 + 9 = 987654321
    
```

**Exercice 57**

Soit à saisir deux entiers F et G tels que  $F > G > 0$ . Il s'agit d'écrire un programme nommé «Range» qui permet d'afficher les F premiers entiers naturels non nuls sur G colonnes et avec le minimum de lignes. La lecture des entiers se fera par colonnes, de gauche à droite et de haut en bas.

Exemple : Pour  $F=17$  et  $G=3$ , l'affichage devra être sous la forme :

```

1  7  13
2  8  14
3  9  15
4 10  16
5 11  17
6 12
    
```

**Exercice 58**

Votre sœur joue avec des perles, des blanches et des noires, et elle souhaite constituer des colliers. Dans un souci d'esthétique, elle désire alterner les couleurs, tout en préservant une certaine périodicité. Elle va donc placer X perles blanches, suivies de Y perles noires et elle répètera ce motif N fois.

Exemple : Supposons qu'elle possède  $X = 12$  perles blanches et  $Y = 6$  perles noires. Elle a plusieurs possibilités :

- une répétition (12 blanches, 6 noires)
- deux répétitions (6 blanches, 3 noires)
- trois répétitions (4 blanches, 2 noires)
- six répétitions (2 blanches, 1 noire)



Écrire un programme appelé «Perles» qui calcule, étant donné deux entiers strictement positifs X et Y, représentant respectivement les nombres de perles blanches et noires, le nombre maximum de répétitions que votre sœur va pouvoir faire avec ces perles.

**Exercice 59**

Écrire un programme Pascal qui affiche tous les nombres palindromes inférieurs à un entier N donné ( $N > 1$ ) et qui peuvent être écrits sous la forme d'une somme des carrés de nombres successifs.

Exemples :

5 est un nombre palindrome et vérifie la propriété car  $5 = 1^2 + 2^2$

595 est un nombre palindrome et vérifie la propriété car  $595 = 6^2 + 7^2 + 8^2 + 9^2 + 10^2 + 11^2 + 12^2$

Remarques :

- Un nombre palindrome est un nombre égal à lui-même s'il est lu de gauche à droite ou de droite à gauche.
- Vous devez afficher le nombre palindrome suivi de la somme de ses carrés successifs.

Exemple d'affichage :  $595 = 6^2 + 7^2 + 8^2 + 9^2 + 10^2 + 11^2 + 12^2$

### Exercice 60

Un **nombre de SMITH** s'il est un nombre dont la somme des chiffres est égale à la somme de tous les chiffres de ses facteurs premiers.

**Exemple :**

$$4937775 = 3 \cdot 5 \cdot 5 \cdot 65837$$

$$\Rightarrow \text{Somme des chiffres : } 4+9+3+7+7+7+5 = 42$$

$$\Rightarrow \text{Somme des chiffres de sa factorisation : } 3+5+5+6+5+8+3+7 = 42$$

Ecrire un programme Pascal qui permet de chercher et d'afficher tous les nombres de **smith** de l'intervalle  $[n, m]$  (avec  $10 < m < n < 100$ ).

### Exercice 61

Un **nombre de Zuckerman** est un nombre qui est divisible par le produit de ses chiffres. Les nombres contenant un 0 sont naturellement exclus. Alors qu'un **nombre de Harshad** est divisible par la somme.

Ecrire un programme Pascal qui permet de chercher et d'afficher tous les nombres qui sont à la fois **Zuckerman** et **Harshad** de l'intervalle  $[n, m]$  (avec  $100 \leq n < m$ ).

### Exercice 62

#### **SUITE DE STEINHAUS**

- On prend un nombre de deux chiffres, au hasard.
- On multiplie les deux premiers chiffres.
- On place le résultat à la suite des chiffres déjà trouvés.

**Exemple :** 23 ;  $2 \cdot 3 = 6$  ; 236 ;  $3 \cdot 6 = 18$  ; 23618 ;  $6 \cdot 1 = 6$  ; 236186 ;  $1 \cdot 8 = 8$  ; 2361868 ;  $8 \cdot 6 = 48$  ; 236186848 ...

Ecrire un programme Pascal qui permet de chercher et d'afficher les 100 premiers chiffres de cette suite.

### Exercice 63

#### **Suite de Padovan**

Les nombres de **Padovan** sont donnés par la récurrence :

$$P_n = P_{n-3} + P_{n-2} \quad \text{avec } P_0 = P_1 = P_2 = 1.$$

Ecrire un programme Pascal qui affiche les 20 premiers nombres de Padovan.

### Exercice 64

Un nombre est **riche** si un de ses facteurs au moins est répété deux fois ou plus.

**Exemples :**

$$4 = 2^2$$

$$90 = 2 \times 3^2 \times 5$$

$$540 = 2^2 \times 3^3 \times 5$$

Ecrire un programme Pascal qui affiche tous les nombres riches  $< 1000$ .

### Exercice 65

Un nombre est **harshad morphique**, s'il est la somme des chiffres d'un nombre de harshad et qui termine aussi ce nombre de harshad.

**Exemple :**  $910 = 9 + 1 + 0 = 10$

**910** est un **nombre de harshad** puisqu'il est divisible par la somme de ses chiffres.

**10** est **harshad morphique**.

Ecrire un programme Pascal qui affiche tous les nombres harshad de l'intervalle  $[100, 99999]$  et qui intègrent les harshad morphique.

### Exercice 66

Un **nombre narcissique** est un entier naturel N non nul qui est égal à la somme des puissances p-ièmes de ses chiffres, où p désigne le nombre de chiffres de N.

**Exemples :**

$$153 = 1^3 + 5^3 + 3^3$$

$$548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$$

Ecrire un programme Pascal qui affiche tous les nombres narcissiques < 10000.

### Exercice 67

Deux entiers **N1** et **N2** sont dits frères si chaque chiffre de **N1** apparaît au moins une fois dans **N2** et inversement.

Ecrire un programme Pascal qui saisit deux entiers **N1** et **N2**, vérifie et affiche s'ils sont frères ou non.

**Exemples :**

- Si N1 = 1164 et N2 = 614 alors le programme affichera : N1 et N2 sont frères
- Si N1 = 905 et N2 = 9059 alors le programme affichera : N1 et N2 sont frères
- Si N1 = 405 et N2 = 554 alors le programme affichera : N1 et N2 ne sont pas frères

### Exercice 68

Si **d** est un diviseur propre de **n** et s'il est **premier avec n/d** (pgcd (d, n/d)=1), c'est un **diviseur unitaire** de n.

Un **nombre unitairement parfait** est égal à la somme de ses diviseurs unitaires.

Ecrire un programme Pascal qui affiche tous les nombres **unitairement parfaits** < 100000.

### Exercice 69

Tout entier strictement positif peut s'écrire sous la forme d'une somme de puissance de, où les exposants sont distincts.

**Exemples :**  $31 = 2^0 + 2^1 + 2^2 + 2^3 + 2^4$

$$146 = 2^1 + 2^4 + 2^7$$

Ecrire un programme Pascal qui permet d'afficher la décomposition d'un entier **N**, en une somme de puissances de 2 distinctes.

### Exercice 70

On veut afficher à l'écran le calendrier d'un mois donné. On donne le mois, l'année et le jour de la semaine correspondant au premier du mois en question. Écrivez un programme Pascal qui réalise cette tâche.

**Exemple:**

Donner le n° du mois = 5

Donner le jour de début de mois (lundi=1, mardi=2, ...) = 3

Lundi	6	13	20	27	
Mardi	7	14	21	28	
Mercredi	1	8	15	22	29
Jeudi	2	9	16	23	30
Vendredi	3	10	17	24	31
Samedi	4	11	18	25	
Dimanche	5	12	19	26	

### Exercice 71

**Nombre sublime** : La quantité de diviseurs et la somme des diviseurs sont deux nombres parfaits.

**Exemple : 12 est un nombre sublime.**

Diviseurs : 1, 2, 3, 4, 6 et 12

Quantité : 6 Nombre parfait

Somme : 28 Nombre parfait

### Exercice 72

Un nombre "**extra-ordinaire**" est un entier naturel composé des chiffres distincts non nuls et divisible par chacun de ses chiffres.

On se propose d'écrire un programme Pascal permettant de chercher et d'afficher tous les nombres "**extra-ordinaires**" de l'intervalle [10, 1000].

### Exercice 73

On se propose d'écrire un programme Pascal permettant de chercher et d'afficher tous les nombres de 6 chiffres dont la concaténation de deux nombres consécutifs formant un carré.

**Exemple** :  $183\ 184 = 428^2$

### Exercice 74

On se propose d'écrire un programme Pascal permettant de chercher dans l'intervalle [1000,9999], combien de nombres sont-ils singuliers au point de n'avoir aucun chiffre répété, comme 1234 ? Et, en retour, combien ont des chiffres répétés, comme 1123.

### Exercice75

En arithmétique géométrique, un **nombre oblong**, ou **nombre pronique**, est le produit de deux entiers naturels consécutifs, c'est-à-dire,  $n(n + 1)$ .

**Exemple** :  $20 = 4 \times 5$

Ecrire un programme Pascal qui affiche les vingt premiers nombres oblongs.

### Exercice76

En mathématiques, un nombre **premier palindrome** est un nombre **premier** qui est aussi un nombre **palindrome**.

**Exemples** : 7, 11, 151, 919, ...

Ecrire un programme Pascal qui affiche les vingt premiers nombres **premiers palindromes**.

### Exercice77

En additionnant un nombre et son **renversé**, on peut trouver un nombre **palindrome**. Par exemple,  $23 + 32 = 55$ ,  $145 + 541 = 686$ . Si le nombre obtenu n'est pas palindrome, on peut exécuter la même opération sur le résultat jusqu'à ce qu'on obtienne un palindrome.

Par exemple,  $57 + 75 = 132$  et  $132 + 231 = 363$ .

Ecrire un programme Pascal qui affiche les nombres palindromes en appliquant cette méthode sur des nombres de trois chiffres.

### Exercice78

La **conjecture de Goldbach** stipule que tout nombre pair  $n$  plus grand que 2 est la somme de deux nombres premiers. Ces nombres premiers  $p$  et  $q$  sont appelés décomposant de Goldbach de  $n$ .

**Exemple** :  $10 = 3 + 7$  ;  $10 = 5 + 5$

Ecrire un programme Pascal qui teste, pour tout entier pair compris entre 4 et 400, teste si la conjecture est vérifiée ou non. Dans l'affirmation, le programme affichera  $n$ ,  $p$  et  $q$  tels que  $n=p+q$ .

### Exercice79

On note que  $4 = 2+2=2 \times 2$

$6 = 1+2+3=1 \times 2 \times 3$

$8 = 1+1+2+4=1 \times 1 \times 2 \times 4$

Ecrire un programme Pascal qui permet de trouver les nombres  $N$  tels que :

**$N = X_1 + X_2 + \dots + X_n = X_1 \times X_2 \times \dots \times X_n$**  avec  **$X_1 \leq X_2 \leq \dots \leq X_n$**

### Exercice80

En arithmétique, un **triplet pythagoricien** est un triplet  $(x, y, z)$  d'entiers naturels non nuls vérifiant la relation de Pythagore :  $x^2 + y^2 = z^2$ .

Un triplet pythagoricien  $(x, y, z)$  est dit **primitif** si les trois entiers  $x$ ,  $y$  et  $z$  sont premiers entre eux dans leur ensemble ( $\text{PGCD}(x,y,z)=1$ ).

**Exemples :** (3, 4, 5) (5, 12, 13) (8, 15, 17)

Tous les triplets primitifs de Pythagore s'obtiennent à partir des formules :

$x=a^2-b^2$   $y=2ab$   $z=a^2+b^2$  où  $a>b$ ,  $\text{PGCD}(a, b)=1$ ,  $a$  et  $b$  de parités distinctes.

Ecrire un programme Pascal qui affiche tous les triplets primitifs dont tous les termes sont inférieurs à 100.

Fenni Salah

# LES TABLEAUX

## Exercice 1

Ecrire un programme Pascal, qui fait remplir un tableau T par n ( $5 < n \leq 10$ ) entiers compris entre 1 et 20. Ensuite, il fait calculer et afficher la **somme**, le **produit** et la **moyenne arithmétique** des éléments de T.

## Exercice 2

Ecrire un programme Pascal, qui fait remplir un tableau T par n **lettres** ( $2 < n \leq 20$ ). Ensuite il fait afficher, **sans redondance**, les éléments de T.

## Exercice 3

Ecrire un programme Pascal qui fait réaliser les tâches suivantes :

- Remplir un tableau LET de 35 cases par des lettres majuscules au hasard. On utilisera la fonction prédéfinie Random.
- Compter la **fréquence d'apparition** (le nombre d'occurrence) de chaque lettre majuscule dans le tableau LET et ranger les résultats statistiques dans un tableau FE.

## Exercice 4

Ecrire un programme Pascal qui permet de convertir un entier naturel en son équivalent binaire (**conversion de la base 10 à la base 2**).

## Exercice 5

Ecrire un programme Pascal qui permet de convertir un nombre écrit dans une **base b1** en son équivalent dans une **base b2**.

## Exercice 6

Soit T un tableau contenant N entiers ( $10 \leq N \leq 50$ ). On propose d'écrire un programme Pascal qui permet d'**éclater T** en deux tableaux :

TN (contenant les éléments négatifs de T) et TP (contenant les éléments positifs de T).

## Exercice 7

Soit T un tableau contenant N entiers ( $10 \leq N \leq 50$ ). On propose d'écrire un programme Pascal qui permet d'**inverser** les éléments de T (permuter T [1] et T[n], puis T [2] et T [n-1],...).

## Exercice 8

Soit T un tableau contenant N entiers ( $10 \leq N \leq 50$ ). On propose d'écrire un programme Pascal qui permet de **regrouper** les éléments pairs au début et les éléments impairs à la fin de T, sans modifier l'ordre de saisie des valeurs paires et impaires.

**Exemple** : pour N = 10

4	2	7	8	20	5	11	6	3	30
---	---	---	---	----	---	----	---	---	----

Etat initial de T

4	2	8	20	6	30	7	5	11	3
---	---	---	----	---	----	---	---	----	---

Etat final de T

## Exercice 9

Soit T un tableau contenant N entiers ( $10 \leq N \leq 50$ ). On propose d'écrire un programme Pascal qui permet de déterminer et d'afficher la **valeur maximale** et la **valeur minimale** de T.

## Exercice 10

On se propose d'écrire un programme Pascal qui permet de saisir un entier N >1 et pair. Ensuite remplir un tableau T par n entiers égaux deux par deux. Enfin transformer T en un **tableau symétrique**.

**Exemple** : pour N = 10

4	4	0	0	-5	-5	8	8	3	3
---	---	---	---	----	----	---	---	---	---

Etat initial de T

4	0	-5	8	3	3	8	-5	0	4
---	---	----	---	---	---	---	----	---	---

Etat final de T

### Exercice 11

Soit un tableau de N ( $5 \leq N \leq 20$ ) entiers. On se propose de **ranger** les éléments de T de façon à mettre les éléments positifs ou nuls au début de T suivis des éléments négatifs.

Exemple :

T initial							T final						
2	-3	-9	0	15	-1	6	2	0	15	6	-9	-1	-3

### Exercice 12

Soit T un tableau contenant N entiers ( $10 \leq N \leq 50$ ). On propose d'écrire un programme Pascal qui permet de **chercher l'existence d'un élément** V donné, dans la liste de valeurs de T.

### Exercice 13

Soit le tableau T suivant :

10	7	9	7	10	6	7	4	8	8
----	---	---	---	----	---	---	---	---	---

Pour chaque élément de T on ne garde que sa première occurrence et on remplace les autres par 0.

10	7	9	0	0	6	0	4	8	0
----	---	---	---	---	---	---	---	---	---

On regroupe les éléments restant au début du tableau T.

10	7	9	6	4	8	0	0	0	0
----	---	---	---	---	---	---	---	---	---

Ecrire un programme Pascal qui fait le traitement ci-dessus pour un tableau T de n ( $2 \leq n \leq 20$ ) entiers positifs non nuls.

### Exercice 14

Ecrire un programme Pascal qui fait remplir un tableau T par les résultats de 20 lancements d'un dé. Le programme doit faire remplir par la suite un **tableau fréquence** F par le nombre de fois que chaque face est obtenue.

### Exercice 15

On dispose d'un tableau MOY qui contient la liste des moyennes de N élèves. On propose d'écrire un programme Pascal qui permet de déterminer et d'afficher le **rang** de chaque élève.

### Exercice 16

On dispose d'un tableau de N\_MAX éléments rempli par N caractères ( $N < N\_MAX$ ). Ecrire un programme Pascal permettant d'**insérer un caractère** C donné à la k\_ème position (avec  $K \leq N$ ).

### Exercice 17

Le **triangle de Pascal** est obtenu en considérant que pour une ligne donnée :

- Le premier élément et le dernier élément sont égaux à 1.
- Les autres éléments sont obtenus au moyen de la relation :  $T[l, c] = T[l-1, c] + T[l-1, c-1]$

Ecrire un programme Pascal permettant de générer et afficher le triangle de pascal de taille n.

Exemple d'exécution :

Taille du triangle : 5

Le triangle correspondant est :

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

### Exercice 18 Transposition d'une Matrice carrée

Une Matrice carrée est une matrice à n lignes et n colonnes. L'opération de transposition consiste à inverser les lignes et les colonnes en effectuant une symétrie par rapport à la diagonale principale de la matrice.

Exemple :

La Matrice	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> </table>	1	2	3	4	5	6	7	8	9	Devient	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>4</td><td>7</td></tr> <tr><td>2</td><td>5</td><td>8</td></tr> <tr><td>3</td><td>6</td><td>9</td></tr> </table>	1	4	7	2	5	8	3	6	9
1	2	3																			
4	5	6																			
7	8	9																			
1	4	7																			
2	5	8																			
3	6	9																			

Ecrire un programme Pascal qui permet de saisir une Matrice carrée puis il cherche et affiche son transposée.

### Exercice 19

L'algorithme suivant permet de calculer et d'afficher les moyennes des trois notes de  $n$  élèves.

- 0) Début Calcul\_moyennes
- 1) Lire ( $n$ )
- 2) Pour  $i$  de 1 à  $n$  faire  
    Lire (Note1, Note2, Note3)  
    Moy  $\leftarrow$  (Note1 + 2 x Note2 + 2 x Note3)/5  
    Ecrire (Moy)
- FinPour
- 3) Fin Calcul\_moyennes

#### Questions :

1. Traduire cet algorithme en Pascal.
2. Dans l'action 1, ajouter le contrôle sur la saisie pour assurer la condition suivante  $5 \leq n \leq 40$ .
3. Ajouter une fonction intitulée **RANG** qui permet de chercher le rang de l'élève  $n^{\circ}i$ .
4. Apporter les changements adéquats afin d'afficher pour chaque élève et sur une même ligne :  

<b>N°</b>	<b>Moyenne</b>	<b>Rang</b>
-----------	----------------	-------------

### Exercice 20

On se propose d'utiliser l'algorithme de tri par sélection pour **trier un tableau** de 20 chaînes de caractères.

### Exercice 21

Soient deux tableaux T1 et T2 contenant chacun  $n$  éléments distincts deux à deux ( $2 < n < 100$ ). On appelle **intersection** de T1 et T2 l'ensemble des éléments communs à ces deux tableaux. On se propose d'écrire un programme Pascal, qui range les éléments de l'intersection des deux tableaux dans un tableau INTER puis affiche les trois tableaux T1, T2 et INTER.

### Exercice 22

On se propose d'écrire un programme Pascal permettant :

1. De saisir les éléments d'un tableau T composé de  $n$  chaînes de caractères non vides.
2. De trier le tableau T dans un ordre croissant selon les deux critères suivants :
  - Longueur de la chaîne en premier lieu.
  - Ordre alphabétique en cas d'égalité pour les longueurs.

### Exercice 23

Une version de tri à bulles est le tri bidirectionnel. Elle consiste à parcourir le tableau de gauche à droite, puis de droite à gauche, le changement de direction ayant lieu chaque fois que l'une des extrémités est atteinte. Ainsi, les plus petits éléments du tableau descendant au même rythme que remontent les plus grands éléments.

On se propose d'écrire un programme Pascal permettant de saisir  $n$  entiers ( $10 < n < 20$ ) dans un tableau T et de le trier en utilisant le principe mentionné ci-dessus.

### Exercice 24

Nous disposons de deux tableaux T de  $n$  noms et C de  $n$  couleurs ('B' ou 'N').

Ecrire un programme Pascal qui permet de réarranger les éléments de T et C de manière à ce que les éléments de couleur 'B' précèdent les éléments de couleur 'N'. Si deux éléments ont des couleurs identiques, l'ordre alphabétique des chaînes intervient.

Exemple :

Ali	Salah	Sonia	Tounsi	Salma	Ahmed
B	N	B	N	N	B

Sont réarrangés comme suit :

Ahmed	Ali	Sonia	Salah	Salma	Tounsi
B	B	B	N	N	N

### Exercice 25

Ecrire un programme Pascal, qui permet de créer un tableau V3 d'entiers croissants à partir de deux tableaux V1 et V2 d'entiers.

Exemple :

V1	1	3	2	-6			
V2	0	4	-5				
V3	-6	-5	0	1	2	3	4

### Exercice 26

Ecrire un programme Pascal permettant de faire une étude comparative du temps d'exécution des algorithmes de tri (sélection, bulles et insertion)

### Exercice 27

Ecrire, en s'inspirant du tri par sélection, une procédure qui permet de construire à partir d'un tableau T de n entiers un tableau Rang tel que Rang[i] soit l'indice dans T de l'ième élément dans l'ordre croissant sans modifier le tableau T.

Exemple :

T	80	50	90	35	20
	1	2	3	4	5
Rang	5	4	2	1	3

### Exercice 28

Ecrire un programme Pascal qui permet de remplir un tableau T par n caractères, de rechercher dans le tableau la première plus longue suite de caractères identiques et d'afficher le caractère concerné ainsi que le nombre de fois qu'il est répété.

Exemple : Si on introduit (a a b c c e d e e e e f f g a a a).

Le programme doit afficher (e, 5).

### Exercice 29

Ecrire un programme Pascal permettant de remplir d'une manière automatique et aléatoire, un tableau T par N entiers (de 100 à 999), avec ( $5 \leq N \leq 200$ ), et d'afficher les éléments qui sont symétrique.

### Exercice 30

Ecrire un programme Pascal permettant de saisir, dans un tableau T, N entiers positifs triés dans l'ordre croissant, puis d'afficher les entiers manquants entre le premier et le dernier élément de ce tableau T ainsi que leur nombre.

Exemple : Si N = 7 et si T est le tableau suivant

T	5	6	8	9	10	12	15
i	1	2	3	4	5	6	7

Le programme affichera : Les entiers manquants sont : 7 11 13 14 ; leur nombre est : 4

### Exercice 31

Écrire un programme Pascal intitulé **Séquence** permettant de :

- Remplir un tableau T par N ( $2 \leq N \leq 24$ ) entiers non nuls.
- Chercher et afficher la première longue séquence d'éléments **juxtaposés** dont la somme est égale à zéro.

Exemple : Si N = 10 et si T est le tableau suivant :

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

Le programme affichera : La plus longue séquence est : 6 -2 -2 -1 -1

### Exercice 32

Écrire un programme Pascal intitulé **EI\_Frequent** permettant de remplir d'une manière automatique et aléatoire, un tableau T par N (avec  $5 \leq N \leq 20$ ) entiers (de 0 à 9) et d'afficher l'élément qui apparaît le plus dans le tableau T, ainsi que son nombre d'occurrences. Si plusieurs éléments différents répondent à la condition, le programme doit en fournir le premier.

### Exercice 33

Écrire un programme Pascal permettant de vérifier l'existence d'une chaîne donnée **Ch** dans un tableau T de n chaînes de caractères ( $2 \leq n \leq 10$ ). Les éléments du tableau T ainsi que la chaîne recherchée sont formés de n caractères. On admet que la chaîne **Ch** est dans le tableau T si elle est :

- l'un des éléments du tableau,
- l'inverse d'un élément du tableau.

**Exemple :** Pour  $n = 4$ , le tableau T contient 4 chaînes formées chacune de 4 caractères. On veut vérifier l'existence de la chaîne **Ch** dans le tableau T.

Si **Ch** = "HOUE"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

**Ch** est le troisième élément du tableau T.

Si **Ch** = "DOUE"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

**Ch** est l'inverse du deuxième élément du tableau T.

Si **Ch** = "ROUT"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

**Ch** n'existe pas dans le tableau T.

### Exercice 34

Soit T un tableau de N chaîne de caractères non vides et dont la taille maximale est 5 caractères. On se propose d'écrire un programme permettant de remplir le tableau T par N chaînes de caractères ( $2 \leq N \leq 30$ ), puis de calculer et d'afficher la somme des nombres extraits des chaînes de chaque élément du tableau T.

Le nombre extrait de la chaîne contenue dans la case i du tableau T, est formé par la concaténation de tous les chiffres de la chaîne parcourue de gauche à droite.

**N.B. :** Si une chaîne ne contient pas des chiffres, elle prend la valeur 0 dans le calcul de la somme finale.

**Exemple :** Si  $N = 9$  et que le tableau T contient les éléments suivants :

T	R4*s2	12hj5	5?7e	Ak!r	E9Y41	6754	3E-Z2	G(Y	U5Kx1
	1	2	3	4	5	6	7	8	9

Alors la somme  $S = 42 + 125 + 57 + 0 + 941 + 6754 + 32 + 0 + 51 = 8002$

### Exercice 35

Un élément X du tableau T est dit **majoritaire** si et seulement si T contient strictement plus de  $(n/2)$  occurrences de X.

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

T	25	9	9	9	7	9	12	12	9	9
	1	2	3	4	5	6	7	8	9	10

L'entier 9 possède 6 occurrences qui est  $> n/2$  donc **9 est majoritaire**.

Écrire un programme Pascal qui permet de saisir un entier N ( $5 \leq N \leq 25$ ), puis de remplir un tableau T par N entiers, de vérifier si T contient un élément majoritaire.

### Exercice 36 Intervalle de plus grande somme

Nous avons un tableau T de n entiers relatifs. Nous recherchons un sous-tableau de T dont la somme des éléments adjacents soit maximale.

**Exemple :** Pour n=9 et le tableau T suivant :

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

La plus grande somme est définie par les valeurs : 6, 7 et 1.

### Exercice 37 Segmentation d'un tableau

La segmentation d'un tableau T de N entiers par rapport à T [1] consiste à placer les éléments inférieurs ou égaux à T [1] à sa gauche et les éléments strictement supérieurs à T [1] à sa droite.

**Exemple :** Soient N=10 et le tableau T suivant :

125	10	34	192	125	1026	22	-365	411	48
1	2	3	4	5	6	7	8	9	10

La segmentation de T par rapport à l'élément numéro 1 donne le tableau suivant :

10	34	125	22	-365	48	125	192	1026	411
1	2	3	4	5	6	7	8	9	10
Eléments ≤ à 125						Eléments > à 125			

Ecrire un programme Pascal qui permet de saisir un entier N (5 ≤ N ≤ 20), de remplir un tableau T par N entiers, de segmenter le tableau T par rapport à T [1] et d'afficher le tableau T segmenté.

### Exercice 38

Soit T un tableau de N chaîne de caractères alphabétiques minuscules (2 ≤ N ≤ 10).

Ecrire un programme Pascal qui permet de saisir les éléments du tableau T et affiche les lettres communes à toutes les chaînes sans redondance.

**Exemple :** Si N=4 et le tableau T suivant :

T	palindrome	amour	amirale	marron
	1	2	3	4

Le programme affichera : Les caractères communs sont : a, m, r

### Exercice 39

Soit T un tableau de N chaîne de caractères constitués uniquement de 0 et 1 (2 ≤ N ≤ 10).

Ecrire un programme Pascal qui permet de saisir les éléments du tableau T, de chercher et d'afficher la première chaîne contenant la plus longue séquence des 1 adjacents.

**Exemple :** Si N=5 et le tableau T suivant :

T	100001	1011011	011110	0000	101111
	1	2	3	4	5

Le programme affichera : 011110

### Exercice 40 Les nombres premiers

Pour obtenir les nombres premiers inférieurs à n :

Écrire tous les entiers de 2 à n,

Enlever (ou barrer) les multiples de 2 sauf 2,

Récupérer le plus petit nombre non barré, c'est à dire 3, et barrer les multiples de 3 sauf 3, etc.

On s'arrête dès qu'on a atteint la racine carrée de n.

Faire un programme en Pascal permettant de trouver ces nombres premiers.

### Exercice 41 nombre polite

Dans la théorie des nombres, un nombre polite est un entier positif qui peut être écrit comme la somme de deux ou plusieurs nombres entiers positifs consécutifs.

**Exemple :**

$$0 + 1 = 1, 1 + 2 = 3, 2 + 3 = 5, 1 + 2 + 3 = 6, 3 + 4 = 7, 4 + 5 = 9, 1 + 2 + 3 + 4 = 10 \dots$$

Faire un programme en Pascal permettant de trouver ces nombres polites.

### Exercice 42 les nombres chanceux d'ULAM

Pour trouver les nombres chanceux d'ULAM, il faut procéder par étapes :

Nous commençons avec une liste d'entiers démarrant par 1 :

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, ...

Puis nous enlevons un nombre sur deux, ce qui ne laisse que les entiers impairs :

1, -, 3, -, 5, -, 7, -, 9, -, 11, -, 13, -, 15, -, 17, -, 19, -, 21, -, 23, -, 25, -, ...

Le deuxième terme de la suite est désormais 3. Maintenant, nous enlevons un nombre sur trois parmi ceux qui restent dans la liste : 1, 3, -, 7, 9, -, 13, 15, -, 19, 21, -, 25, 27, -, 31, 33, -, 37, 39, -, 43, 45, -, 49, 51, ...

Le troisième nombre survivant est 7. Maintenant, nous enlevons un nombre sur sept parmi ceux qui restent dans la liste : 1, 3, 7, 9, 13, 15, -, 21, 25, 27, 31, 33, 37, -, 43, 45, 49, 51, 55, 57, -, 63, 67, 69, 73, ...

Le quatrième nombre survivant est 9. Maintenant, nous enlevons un nombre sur neuf parmi ceux qui restent dans la liste, etc.

Si nous répétons cette procédure indéfiniment, les survivants sont les nombres chanceux d'ULAM :

1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, 87, 93, 99, ...

L'idée est de faire un programme en Pascal permettant de trouver ces nombres chanceux.

### Exercice 43

Soit T un tableau de N éléments ( $3 \leq N \leq 100$ ) contenant 3 types de couleurs : **V** (vert), **R** (rouge) et **B** (bleu). Ecrire un programme Pascal qui ordonne les couleurs comme suit :

**RRR...BBBBBB...VVVVV**

### Exercice 44

Un nombre X de trois chiffres est dit **premier absolu** s'il est premier et que tous les nombres formés par les combinaisons de ses trois chiffres sont aussi premiers.

**Exemple :** X=337 est un nombre premier absolu car 337, 373, 733 sont premiers.

Ecrire un programme Pascal qui permet de remplir un tableau T de N ( $5 \leq N \leq 30$ ) entiers positifs de trois chiffres, de chercher et d'afficher tous les entiers premiers absolus de T.

### Exercice 45

Ecrire un programme Pascal permettant de :

- Remplir aléatoirement un tableau T par N (avec  $4 < n \leq 20$ ) nombres premiers de l'intervalle [2,99].
- Afficher toutes les séquences de valeurs consécutives croissantes, ainsi que leur nombre.

Exemple : pour le tableau T suivant :

T	5	29	13	17	53	23	2	5	19
---	---	----	----	----	----	----	---	---	----

Le programme affichera :

5 29

13 17 53

23

2 5 19

Le nombre de séquences est 4

### Exercice 46

A l'approche de la naissance de leur enfant, un couple superstitieux contacte une voyante pour qu'elle lui recommande des lettres porte-bonheur pouvant être utilisées dans la nomination du futur bébé. Pour cela, la voyante effectue N tirages au hasard de P cartes (avec  $1 \leq P \leq 10$  et  $3 < N < 20$ ) où chaque carte comporte une lettre majuscule. A chaque tirage en résulte une chaîne de caractères formée par la concaténation des lettres tirées. Les lettres porte-bonheur sont celles les plus tirées dans les différents tirages.

**Exemple :** Pour  $N=4$  et  $P=5$

Les tirages effectués ont donné les quatre chaînes des caractères suivantes :

"HBAMX", "MSAIS", "HREMA" et "DRTYU"

D'où les lettres **porte-bonheur** sont : **A** et **M**, car elles sont les plus tirées (3 fois).

Ecrire un programme Pascal qui permet de saisir les deux entiers N et P et simuler les tâches effectuées par la voyante et d'afficher les lettres porte-bonheur correspondantes aux tirages effectués.

### Exercice 47

Un entier est dit **Zig-Zag** lorsque ses chiffres croissent et décroissent ou inversement d'une façon alternative.

**Exemples :**

- Le nombre 13254 est dit **Zig-Zag**, car  $1 < 3$ ,  $3 > 2$ ,  $2 < 5$  et  $5 > 4$
- Le nombre 8781 est dit **Zig-Zag**, car  $8 > 7$ ,  $7 < 8$  et  $8 > 1$
- Le nombre 8761 est dit **non Zig-Zag** car  $8 > 7$  et  $7 > 6$
- Le nombre 9667 est dit **non Zig-Zag** car  $9 > 6$  et  $6 = 6$

Ecrire un programme Pascal qui permet de remplir un tableau **T** par **N** ( $5 \leq N \leq 25$ ) entiers positifs composés d'au minimum trois chiffres et d'afficher les nombres Zig-Zag de T.

### Exercice 48

Ecrire un programme Pascal qui permet de déterminer et d'afficher si les éléments d'un tableau T de n ( $2 \leq n \leq 15$ ) entiers constituent ou non une **suite géométrique**.

**N.B :** Une suite U est dite géométrique si et seulement si il existe un réel q tel que pour tout entier n, on a :  $U_n = q * U_{n-1}$

### Exercice 49

Ecrire un programme Pascal qui permet de déterminer et d'afficher si les éléments d'un tableau T de n ( $2 \leq n \leq 15$ ) entiers constituent ou non une **suite arithmétique**.

**N.B :** Une suite U est dite arithmétique si et seulement si il existe un réel r tel que pour tout entier n, on a :  $U_n = U_{n-1} + r$

### Exercice 50

Ecrire un programme Pascal qui permet de saisir n ( $10 < n < 40$ ) entiers à mettre dans un tableau T et deux entiers non nuls p et s (p et s doivent être deux éléments de T). On demande d'afficher tous les blocs d'éléments de T placés entre p et s dans l'ordre (p et s peuvent figurer dans cet ordre plusieurs fois dans T).

**Exemple :** Si p = 5 et s = 3 et si on donne le tableau T suivant :

T	7	5	0	3	9	1	5	6	8	3
---	---	---	---	---	---	---	---	---	---	---

Alors le résultat de l'affichage sera : 0 6 8

### Exercice 51

Ecrire un programme Pascal permettant de remplir, d'une façon aléatoire, deux tableaux T1 et T2 de N ( $2 \leq N \leq 15$ ) entiers à deux chiffres chacun, puis de former un tableau T tel que un élément T[i] est le résultat de la fusion des deux éléments T1[i] et T2[i] selon le principe suivant :

- Insérer le chiffre des dizaines du plus petit nombre parmi T1[i] et T2[i], entre les deux chiffres du plus grand nombre parmi T1[i] et T2[i].
- Mettre le chiffre des unités du plus petit nombre parmi T1[i] et T2[i], à droite du nombre obtenu.

**Exemples :** - Pour T1[i] = 52 et T2[i] = 36 , T[i] sera égal à 5326  
- Pour T1[i] = 13 et T2[i] = 47 , T[i] sera égal à 4173

### Exercice 52

On se propose d'écrire un programme Pascal qui permet de réaliser le traitement suivant :

- Saisir un entier **N** ( $5 \leq N \leq 30$ ).
- Remplir un tableau **NOM** par N noms d'élèves, sachant qu'un nom est composé uniquement par des lettres alphabétiques et des espaces.
- Remplir un tableau **MOY** par les moyennes d'élèves de façon à ce que **MOY[i]** soit la moyenne de **NOM[i]**.
- Afficher le classement des élèves selon l'ordre décroissant des moyennes. Chaque suite des noms qui ont la même moyenne sur une ligne à part.

#### Exemple :

Pour  $N=5$  et les deux tableaux suivants :

<b>NOM</b>	Rim	Adel Hlou	Zied	Dorsaf	Folla
	1	2	3	4	5
<b>MOY</b>	16.5	13	16.5	18	8.5
	1	2	3	4	5

Le programme affiche :

Le classement est :

Rang 1 : Dorsaf

Rang 2 : Rim, Zied

Rang 3 : Adel Hlou

Rang 4 : Folla

### Exercice 53

Ecrire un programme en Pascal calculant la longueur de la plus longue suite croissante dans un tableau.

**Exemple :** pour le tableau **T** suivant :

<b>T</b>	5	3	1	2	7	9	5	6	8
----------	---	---	---	---	---	---	---	---	---

Le programme affichera :

La longueur de la plus longue suite croissante = 4

1 2 7 9

### Exercice 54

On se propose d'écrire un programme Pascal permettant de :

- Remplir un tableau **T** par **N** entiers (*avec  $3 \leq N < 100$  et multiple de 3*). Les éléments de **T** sont saisis dans un ordre strictement croissant.
- Calculer la somme de chaque triplet de valeurs successives de **T** et ranger le résultat dans le même tableau à partir du début sans utiliser un tableau intermédiaire (*voir exemple*).
- Compléter le reste du tableau par des zéros.
- Afficher le tableau résultat.

**Exemple :** Si  $N=12$  et les éléments du tableau **T** sont :

<b>T</b>	4	7	8	10	15	20	21	22	25	35	40	50
	1	2	3	4	5	6	7	8	9	10	11	12

Alors le programme affichera le tableau suivant :

<b>T</b>	19	45	68	125	0	0	0	0	0	0	0	0
	1	2	3	4	5	6	7	8	9	10	11	12

### Exercice 55

On se propose d'écrire un programme Pascal permettant de :

- Remplir un tableau **T** par **N** caractères alphabétiques majuscules (avec  $4 \leq N \leq 100$  et multiple de 4).
- Inverser symétriquement les caractères de chaque bloc de longueur **d** (avec  $d < n$  et diviseur de  $n$  donné).
- Afficher le tableau résultat.

**Exemple :** Si **N=12** et **d=4** et les éléments du tableau **T** sont :

T	I	N	F	O	R	M	A	T	I	Q	U	E
	1	2	3	4	5	6	7	8	9	10	11	12

Alors le programme affichera le tableau suivant :

T	O	F	N	I	T	A	M	R	E	U	Q	I
	1	2	3	4	5	6	7	8	9	10	11	12

### Exercice 56

Soit **T** un tableau de **N** caractères numériques (avec  $5 \leq N \leq 20$ ). Écrire un programme permettant de crypter les données figurant dans le tableau **T** comme suit :

1. Convertir les éléments de **T** compris entre deux positions **P1** et **P2** (avec  $1 \leq P1 < P2 \leq N$ ) en leurs représentations en code ASCII.
2. Permuter les chiffres des unités avec ceux des dizaines du code ASCII puis déterminer le caractère correspondant à ce nouveau code ASCII.
3. Faites une permutation de façon circulaire vers la droite d'ordre 1 sur le contenu du tableau **T**.

### Exercice 57

Une suite est dite arithmétique si, pour tout  $n > 0$ ,  $U_n = U_{n-1} + r$ .

On se propose d'écrire un programme Pascal permettant de réaliser les traitements suivants :

- Remplir un tableau **T** par **N** entiers (avec  $10 \leq n \leq 50$  et  $n$  est pair).
- Remplir un tableau **V** à partir des éléments du tableau **T**. Chaque case de **V** contiendra la valeur absolue de la différence d'un élément de **T** et de son symétrique comme l'illustre l'exemple ci-après.
- Chercher et afficher tous les blocs d'entiers successifs, du tableau **V**, qui forment les termes d'une suite arithmétique de raison **r**. On affichera chaque bloc suivi de la mention "est une suite arithmétique de raison **r** = " suivi de sa raison **r**.

**Exemple :**

A partir du tableau **T** ci-dessous, on obtient le tableau **V** suivant :

T	4	10	0	25	46	6	30	8	0	19	31	12	2	20	7	15	0	16	17	38	20	-2	5	6
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

V	2	5	2	5	8	11	14	8	15	12	11	10
	1	2	3	4	5	6	7	8	9	10	11	12

$$V[1] = | T[1] - T[24] |$$

$$V[2] = | T[2] - T[23] | , \text{ etc.}$$

Pour le tableau **V** précédent, on affichera :

**2, 5, 8, 11, 14 est une suite arithmétique de raison  $r = 3$**

**12, 11, 10 est une suite arithmétique de raison  $r = -1$**

# LES CHAINES DE CARACTERES

## Exercice 1

### **Question de cours :**

Soit la déclaration Pascal suivante : **Var ch : String [63] ;**

- Sur combien d'octets est codée ch ?
- Où est stockée la longueur courante de ch ?
- Quelles sont les 2 façons pour connaître la longueur courante de ch ?

## Exercice 2

Un **palindrome** est un mot, ou une phrase, lisible dans les deux sens, par exemple **kayak** et **radar**. Ecrire un programme Pascal qui vérifie si une chaîne de caractères est palindrome ou non.

## Exercice 3

Ecrire un programme Pascal qui permet d'**inverser** une chaîne de caractères (chaîne miroir).

## Exercice 4

Ecrire un programme Pascal qui convertit une chaîne de caractères, en **minuscule** puis en **majuscule**.

**Exemple :** si ch = 'Turbo PAscal' → 'turbo pascal' → 'TURBO PASCAL'

## Exercice 5

Ecrire trois programmes Pascal qui permettent de saisir une chaîne de caractères en majuscule et l'affiche sous les formes suivantes :

Par exemple, si la chaîne saisie est "TUNIS", on aura :

T	TUNIS	TS	TT
TU	TUNI	TUIS	UTTU
TUN	TUN	TUNNIS	NUTTUN
TUNI	TU	TUNIUNIS	INUTTUNI
TUNIS	T	TUNISTUNIS	SINUTTUNIS

## Exercice 6

On veut écrire un programme Pascal permettant de supprimer les **espaces superflus** dans une chaîne de caractères.

**Exemple :** si la chaîne est : ' Travaux \_pratique \_pascal '

Alors l'exécution du programme donnera la chaîne = 'Travaux\_pratique\_pascal'

## Exercice 7

Ecrire un programme Pascal qui saisit une phrase et l'affiche **renversée**.

Par exemple : 'RESOLUTION DE PROBLEMES' devient 'PROBLEMES DE RESOLUTION'

## Exercice 8

On veut écrire un programme Pascal permettant de :

- Saisir une chaîne de caractères CH
- Parcourir la chaîne CH et afficher l'**occurrence** de chacun de ses caractères.

## Exercice 9

Ecrire un programme Pascal permettant de lire un texte de longueur > à 20, puis déterminer et afficher le **nombre d'occurrence** d'un mot donné, dans ce texte.

## Exercice 10

Ecrire un programme Pascal permettant de lire deux mots ch1 et ch2 et d'afficher tous les caractères qui apparaissent dans les deux chaînes sans redondance.

Exemple : soit ch1 = "Coccinelle" et ch2 = "Cible" Résultat : "Cile"

### Exercice 11

Pour réaliser un bel affichage, on veut **aérer une chaîne** de caractères en insérant un espace entre les caractères de cette chaîne.

Par exemple : DEVOIR devient D\_E\_V\_O\_I\_R

### Exercice 12

Une **anagramme** est un mot obtenu par transposition des lettres d'un autre mot (par exemple **chien, chine** sont des anagrammes du mot **niche**).

Ecrire un programme Pascal qui permet de saisir deux mots non vides MOT1 et MOT2 puis de déterminer si MOT2 est une anagramme de MOT1.

### Exercice 13

Sans utiliser la fonction prédéfinie POS, écrire un programme Pascal qui détermine la **première position** d'une chaîne ch1 dans une autre ch2.

### Exercice 14

Sans utiliser la fonction prédéfinie COPY, écrire un programme Pascal qui **copie** N caractères d'une chaîne ch1 à partir d'une position p.

### Exercice 15

Le Jeu du **PENDU** consiste à :

Un joueur donne un mot en majuscule. Ce mot sera caché sous des tirets, conformément à l'exemple : BONJOUR → B-----R.

L'autre joueur devra alors deviner ce mot de la manière suivante : Il propose chaque fois une lettre. Si cette lettre existe dans le mot caché alors le tiret sera remplacé par la lettre proposée chaque fois où se trouve cette lettre précédemment sinon "Echec" sera affiché jusqu'à ce que le nombre d'essais soient égal à la longueur du mot caché ou le mot sera deviné.

### Exercice 16

Soit un tableau P de n ( $1 < n < 100$ ) chaîne de caractères. Tous les éléments de T doivent être constitués uniquement de chiffres (0..9) et non vides.

On se propose de remplir un tableau Q de la façon suivante :

Q[i] contiendra la chaîne de caractères P[i] écrite à l'envers.

Exemple : Si P[2] = "9025" alors Q[2] contiendra la chaîne "5209"

Ecrire un programme Pascal permettant de saisir les éléments de P, de remplir puis d'afficher les éléments du tableau Q.

### Exercice 17

Ecrire un programme Pascal qui permet de convertir un nombre écrit dans une **base b1** en son équivalent dans une **base b2**.

### Exercice 18

Un entier naturel est dit nombre **ROND** si son écriture binaire contient autant de 1 que de 0.

Exemple : 9 est représenté en binaire par 1001 donc 9 est un nombre ROND.

Ecrire un programme Pascal qui permet de trouver et d'afficher tous les entiers ROND de l'intervalle [1,1000].

### Exercice 19

Un "**totalogramme**" est une chaîne dont chacun de ses mots commence et se termine par la même lettre.

**Exemple** : La chaîne "**ALLALA EMPRUNTE TEMPORAIREMENT A DAOUD SES SOULIERS**" est un "totalogramme".

Ecrire un programme Pascal qui permet de saisir une chaîne de caractères composée uniquement de lettres majuscules et d'espaces (on suppose que deux mots consécutifs sont séparés par un seul espace) ; puis d'afficher un message indiquant si cette chaîne est "**totalogramme**" ou non.

### Exercice 20

Une chaîne est dite **distincte** si elle est composée de caractères distincts (différents).

Ecrire un programme Pascal qui permet de saisir une chaîne CH non vide, puis de vérifier et d'afficher si cette chaîne est distincte ou non.

### Exercice 21 Une suite mystérieuse

La suite correspond à l'énumération orale des chiffres successifs, lus de gauche à droite, en regroupant les chiffres identiques consécutifs.

*Exemple* : soit le nombre : 111221

Le nombre suivant sera ainsi : trois "1", deux "2", un "1", soit 312211

**Soit la suite :**

```
1
11
21
1211
111221
312211
13112221
1113213211
31131211131221
13211311123113112211
```

Ecrire un programme Pascal, qui affiche les n lignes de cette suite.

### Exercice 22

Ecrire un programme pascal qui permet de saisir une chaîne non vide CH de longueur impaire et de l'afficher sous la forme d'un sablier.

**Exemple :**

Si Ch="SABLIER"

Le programme affichera

```
SABLIER
ABLIE
BLI
L
BLI
ABLIE
SABLIER
```

### Exercice 23

On se propose de réaliser le traitement suivant sur une chaîne CH :

Construire une chaîne **RES** à partir de la chaîne CH dans laquelle on rangera toutes les consonnes de CH qui sont en majuscule, suivies de toutes les voyelles de CH qui sont en majuscule, suivies de toutes les consonnes de CH qui sont en minuscule et finalement toutes les voyelles de CH qui sont en minuscule en conservant à chaque fois le même ordre d'apparition des lettres de la chaîne CH.

Ecrire un programme Pascal qui permet de saisir une chaîne CH non vide composée de lettres alphabétiques et dont la taille ne dépasse pas 50 caractères et de construire puis d'afficher la chaîne RES.

**Exemple 1 :** Si CH = "aFABzKOikvMx"

Alors le programme affichera la chaîne "FKMAObzkvxai"

**Exemple 2 :** Si CH = "bonjour"

Alors le programme affichera la chaîne "bnjroou"

### Exercice 24 Plus long palindrome

Un mot est dit palindrome s'il est lu de droite à gauche de la même manière s'il est lu de gauche à droite. Il s'agit de saisir une chaîne de caractères CH et de trouver puis afficher le nombre de palindromes existants et le mot palindrome le plus long. En cas d'existence de palindromes de même taille, nous choisirons celui qui se trouve le plus à gauche.

#### Exemple

**Entrée :** popu45pouluop47Valiseè4568654xLxzezx  
**Sortie :** 10  
pouluop

### Exercice 25

On se propose de **crypter** une phrase de la manière suivante :

- Chaque lettre d'un mot sera remplacée par la lettre qui la suit de P positions dans l'alphabet français, où P est le rang du mot dans la phrase.
- On suppose que le caractère qui suit la lettre "Z" est le caractère "A" et celui qui suit la lettre "z" est le caractère "a".
- Le caractère espace ne subit aucune modification.

#### Exemple :

**Phrase initiale :** Examen Pratique En Informatique  
**Rang des mots :** 1 2 3 4  
**Phrase cryptée :** Fybnfo Rtcvkswg Hq Mrjsvqexmuyi

Ecrire un programme Pascal intitulé **Cryptage** qui permet de :

- Saisir une phrase, dont les mots doivent être formés uniquement de lettres alphabétiques et séparés par un seul espace.
- Crypter la phrase en utilisant le principe cité ci-dessus.
- Afficher le résultat.

### Exercice 26

A partir d'un entier N de p chiffres ( $2 < p < 6$ ), on veut former une chaîne représentant la concaténation de la fréquence de ses facteurs premiers, suivi du facteur premier lui-même. Enfin afficher la chaîne générée.

**Exemple :** pour **p=4** et **n=4592**, les facteurs premiers de N sont : **2, 2, 2, 2, 7, 41**

Donc la chaîne générée sera **"4217141"** pour dire que 4 est la fréquence du facteur premier 2, 1 est la fréquence du facteur premier 7 et 1 est la fréquence du facteur premier 41.

### Exercice 27

La notation des nombres romains est basée l'utilisation des M,D,C, L, X, V et I.

On se propose d'écrire un programme qui, à partir d'une chaîne de caractères formée uniquement de chiffres romains, donne son équivalent décimal selon le principe suivant :

- L'équivalent décimal de chaque chiffre romain est : M=1000, D=500, C=100, L=50, X=10, V=5 et I=1.
- L'équivalent décimal de la chaîne de chiffres romains est obtenu en additionnant les équivalents décimaux de ses chiffres. Le parcours de la chaîne se fait de gauche à droite et dans le cas où un chiffre est inférieur à son successeur, il sera précédé du signe moins (-).

#### Exemples :

CDXL = -100+500-10+50 = 440  
CXVI = 100+10+5+1 = 116  
CXIV = 100+10-1+5 = 114  
MMCIX = 1000+1000+100-1+10 = 2109

### Exercice 28

Soit **Ch** une chaîne de caractères. Écrivez un programme Pascal permettant de dupliquer chaque caractère de la chaîne autant de fois que sa position dans la chaîne.

Attention à bien répéter le 1er caractère 1 fois le second 2 fois, ...

**Exemple :** bonjour → boonnjjjjoooooooouuuuurrrrrrr

### Exercice 29

Le problème consiste à écrire un programme de **codage d'un message** écrit entièrement en majuscules. Ce message pourra comporter aussi des espaces et des caractères de ponctuation. La ponctuation et les espaces ne sont pas codés. Le principe de ce codage consiste à remplacer chaque lettre du message par son image après une rotation de  $k$  caractères des 26 lettres majuscules de l'alphabet. Le sens de cette rotation est celui des aiguilles d'une montre. L'entier  $k$  ( $0 <= k <= 25$ ) sera appelé la clé du codage.

**Exemple :** Pour une rotation avec  $k = 3$ , la lettre «**A**» se transforme en «**D**», la lettre «**B**» en «**E**», la lettre «**Z**» en «**C**», etc...

Dans ce cas,

- le message «**BAC**» sera codé «**EDF**»
- le message «**UN ZEBRE.**» sera codé «**XQ CHEUH.**»

### Exercice 30

Pour sécuriser l'envoi des messages, deux chercheurs cryptent leurs messages en utilisant une clé de cryptage selon le principe suivant :

1. Saisir le message à crypter `msg`, sachant qu'il est composé par des lettres majuscules et des espaces.
2. Saisir la clé de cryptage qui est une chaîne de caractères `chcle` composée uniquement par des chiffres et ayant la même longueur que le message à crypter.
3. Remplacer chaque lettre du message `msg`. d'ordre alphabétique  $i$ , par la lettre d'ordre alphabétique  $j$  avec  $j=i+c$ , sachant que  $c$  est le chiffre de la chaîne `chcle` ayant le même indice que la lettre à crypter.

**NB. :**

- L'espace ne sera pas crypté,
- Si  $j$  dépasse 26, on reprend les lettres alphabétiques dès le début.

**Exemple :**

Soit le message "**EXCELLENTE PERFORMANCE**" et soit la clé "**1954632738401653628451**"

<b>Message initial :</b>	E	X	C	E	L	L	E	N	T	E	P	E	R	F	O	R	M	A	N	C	E	
<b>La clé de cryptage :</b>	1	9	5	4	6	3	2	7	3	8	4	0	1	6	5	3	6	2	8	4	5	1
<b>Message codé :</b>	F	G	H	I	R	O	G	U	W	M	P	F	X	K	R	X	O	I	R	H	F	

En effet :

- La lettre "E" est d'ordre alphabétique 5, elle sera remplacé par la lettre d'ordre alphabétique  $5+1=6$  c'est-à-dire "F"
- La lettre "X" est d'ordre alphabétique 24, elle sera remplacé par la lettre d'ordre alphabétique  $24+9=33$ ,  $33 \text{ MOD } 26 = 7$  c'est-à-dire "G"
- etc.

Ecrire un programme Pascal qui permet de saisir un message et une clé de cryptage en respectant les contraintes citées ci-dessus, puis d'afficher le message crypté en utilisant le principe décrit précédemment.

### Exercice 31

On veut crypter une chaîne de caractères donnée `CH` dont la taille ne dépasse pas 50 caractères, 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.

Ecrire un programme Pascal qui permet de saisir la chaîne `CH` qui doit être non vide et formée uniquement par des lettres alphabétiques, puis de former et d'afficher la chaîne `RES` selon le principe décrit précédemment.

Exemple :

Si `CH="aaafyyBssssssssazz"` Alors la chaîne `RES` qui sera affichée est "`3a1F2y1B9s1a2z`"

### Exercice 32

On appelle poids d'un mot la somme des produits de la position de chaque voyelle contenue dans le mot par son rang dans l'alphabet français.

Une lettre a le même rang qu'elle soit écrite en majuscule ou en minuscule.

Ecrire un programme Pascal qui calcule et affiche le poids d'un mot donné.

#### Exemple :

Le mot « Epreuve » a pour poids 165 car :  $(1*5) + (4*5) + (5*21) + (7*5) = 165$

### Exercice 33

Ecrire un programme Pascal qui permet :

- Remplir un tableau T de N ( $5 \leq n \leq 9$ ) lettres alphabétiques minuscules.
- Saisir un chaîne MOT non vide formée de lettres alphabétiques minuscules de longueur  $\leq 7$
- Vérifier si on peut former cette chaîne à partir des caractères du tableau. Dans le cas où on peut la former, le programme affichera les indices respectifs des caractères dans le tableau T qui composent la chaîne MOT.

**Exemple1** : pour MOT = "admis" et pour le tableau T suivant :

T	s	r	d	a	i	e	p	d	m
	1	2	3	4	5	6	7	8	9

Le programme affichera : 4 3 9 5 1

**Exemple2** : pour MOT = "bac" et pour le tableau T suivant :

T	f	c	k	a	e	h
	1	2	3	4	5	6

Le programme affichera : On ne peut pas former la chaîne à partir des caractères de T.

### Exercice 34

Le terme **Bigramme** désigne la combinaison de deux lettres consécutives et distinctes dans un mot.

On se propose d'écrire un programme Pascal qui permet d'afficher le nombre d'occurrences de chaque bigramme d'une chaîne de caractères **CH** non vide et formée uniquement de lettres minuscules.

#### Exemple :

Pour ch = "mamapappacma"

Le programme affiche :

Le nombre d'occurrences de **ma** est 3

Le nombre d'occurrences de **am** est 1

Le nombre d'occurrences de **ap** est 2

Le nombre d'occurrences de **pa** est 2

Le nombre d'occurrences de **ac** est 1

Le nombre d'occurrences de **cm** est 1

# LES ENREGISTREMENTS ET LES FICHIERS

## Exercice 1:

Soient les tableaux de déclarations des types et des objets suivants :

Type
Jours = (Lun, Mar, Mer, Jeu, Ven, Sam, Dim)
Enreg = Enregistrement
Jrs : Jours
CA : Chaîne [10]
QV : Réel
Fin Enreg
Article = Fichier de Enreg
Fiche = Fichier d'Entier
Vect = Tableau [lun . . ven] de Enreg

Objet	Nature/Type	Rôle
Va	Vect	Vecteur d'articles vendus
Fa	Article	Fichier d'articles vendus
Fe	Fiche	Fichier d'entiers
Avend	Enreg	Les données d'un article vendu
Test	Booléen	Variable booléenne
R	Réel	Variable réelle
Cj	Jours	Compteur
Ft	Texte	Fichiers contenant les noms des articles

## Questions :

Compléter le tableau ci-dessous en mettant dans la 2<sup>ème</sup> colonne la lettre "V" si l'instruction est valide ou par "F" dans le cas contraire, tout en justifiant la réponse si l'instruction est non valide.

Instructions	Validité de l'instruction	Justification (Si Faux)
Ecrire (Fa, Avend.CA)		
Lire (Avend.Jrs)		
Pour Cj de Lun à Jeu Faire Va[cj].jrs ← cj Fin Pour		
Test ← Fin_Fichier (Ft) = Faux		
Ecrire (Fe, R)		
Test ← Va[Lun] > Va[Mar]		

## Exercice 2

Déclarez en pascal les enregistrements relatifs aux descriptions suivantes :

- une matière est caractérisée par son nom, son coefficient et le nombre d'heure par semaine.
- une voiture est caractérisée par sa marque, sa couleur, sa puissance.

## Exercice 3

Soit la déclaration en Pascal suivante :

```
Type point = record
    X: Real;
    Y: Real;
end;
```

```
Var A, B : point ;
```

- 1) Créer par affectation, les points A (-1, 0.9) et B (2.5, 6).
- 2) Lire et afficher les points A et B.

#### Exercice 4

Écrire un programme Pascal qui demande à l'utilisateur les coordonnées de deux points distincts du plan et qui affiche les coordonnées du point milieu.

#### Exercice 5

Écrire un programme Pascal qui permet de calculer et d'afficher, au moyen de type enregistrement, la somme, le produit et la division de deux nombres complexes C1 (a + b i) et C2 (c + d i) tout en utilisant les formules suivantes :

$$(a + b i) + (c + d i) = (a + c) + (b + d) i$$

$$(a + b i) * (c + d i) = (ac - bd) + (ad + bc) i$$

$$(a + b i) / (c + d i) = (ac + bd) / (c^2 + d^2) + (bc - ad) i / (c^2 + d^2)$$

#### Exercice 6

Écrire un programme Pascal qui permet :

- De créer un tableau **Emp** qui contiendra les informations sur les 50 employés d'une entreprise :
  - Matricule (un entier)
  - Nom (chaîne de caractères)
  - Salaire (un réel)
  - Etat\_Civil (M ou C)
- D'afficher le nombre d'employés mariés dont le salaire est  $\geq 800$  Dinars.

#### Exercice 7

Écrire un programme Pascal qui lit au clavier les informations : nom, prénom et âge, relatives à 10 personnes et qui les affiche d'une façon ordonnée suivant le nom.

#### Exercice 8

Écrire un programme Pascal qui permet de saisir un fichier séquentiel des caractères (contenant 26 lettres alphabétique), puis affiche le caractère du milieu.

#### Exercice 9

Écrire un programme Pascal permettant de :

- Créer et remplir un fichier "FP" qui contient les informations sur le personnel d'une entreprise (matricule, nom, grade et salaire).
- Afficher la liste des employés de cette entreprise dont le salaire est compris entre 500D et 800D.
- Rechercher un employé dans le fichier "FP" en connaissant son matricule.

#### Exercice 10

Écrire un programme Pascal permettant :

- de remplir un fichier nommé "Nombres.dat" par N entiers, formé chacun de P chiffres (avec  $2 < N < 100$  et  $2 < P < 6$ ).
- de créer, pour chaque entier du fichier "Nombres.dat", une chaîne représentant la concaténation de la fréquence de chacun de ses facteurs premiers suivi du facteur premier lui-même et l'enregistrer dans une ligne d'un fichier texte nommé "Facteurs.txt".
- d'afficher le contenu du fichier "Facteurs.txt".

**Exemple :**

Pour N = 3 et P = 4 :

<i>Le contenu du fichier "Nombres.dat"</i>	<i>Les facteurs premiers de chaque nombre du fichier "Nombres.dat"</i>	<i>Le contenu du fichier "Facteurs.txt"</i>
1912	2, 2, 2, 239	321239
4592	2, 2, 2, 2, 7, 41	4217141
3525	3, 5, 5, 47	1325147

### **Exercice 11**

Ecrire un programme Pascal qui supprime un composant d'un fichier de données.

### **Exercice 12**

Nous disposons d'un fichier d'entiers enregistré sous le nom physique "Valeurs.dat". Ce fichier est déjà trié en ordre croissant. Nous désirons insérer dans ce fichier et sa bonne place une nouvelle valeur V donnée.

Ecrire l'algorithme d'une procédure nommée INSERTION réalisant cette tâche.

N.B: L'insertion doit se faire directement dans le fichier et non pas à travers un transfert dans un tableau ou dans un autre fichier.

### **Exercice 13**

Ecrire un programme en pascal qui permet lors de son exécution d'afficher lui-même (le programme source).

### **Exercice 14**

Écrire un programme en Pascal permettant de :

- Saisir un ensemble des lignes d'un texte qui se termine par un point.
- Déterminer le nombre des mots par ligne.
- Déterminer le nombre de voyelles de chaque ligne.

### **Exercice 15**

Le contenu de plusieurs pages de texte est enregistré dans un fichier nommé "PAGES.TXT". On désire connaître le nombre total de lettres de ce fichier, l'occurrence et le pourcentage de présence de chaque une de ces lettres.

### **Exercice 16**

Nous désirons, à partir d'un fichier de texte nommé 'c:\modele.txt', lire son contenu et déterminer le code ASCII de chacun de ses caractères, convertir chaque code trouvé en son équivalent binaire et le sauvegarder dans un fichier texte nommé 'c:\binaire.txt'.

N.B.: On termine la saisie des lignes de texte par une ligne vide, qui ne sera pas enregistrée dans le fichier.

Ecrire les analyses et les algorithmes d'un programme modulaire permettant l'affichage du contenu du fichier 'c:\binaire.txt'.

### **Exercice 17**      **Fusion de deux fichiers**

On dispose de 2 fichiers f1 et f2 qui contiennent chacun des chaînes de caractères. Ces fichiers sont triés dans l'ordre croissant. On se propose de fusionner les contenus des deux fichiers f1 et f2 dans un troisième fichier f3 pour obtenir une liste triée par ordre croissant.

**Exemple :**

F1	F2	F3
des	bien	bien
exemples	choisir	choisir
fait	germer	des
la		exemples
solution		fait
		germer
		la
		solution

### **Exercice 18**

On se propose d'écrire un programme qui saisit un fichier texte, puis recopie à partir du fichier, les lignes d'ordre impair dans un 2ème fichier et les lignes d'ordre pair dans un 3ème fichier.

### **Exercice 19**

Ecrire un programme qui permet de tester si deux fichiers d'entiers sauvegardés sur disque dur sont égaux ou non.

### **Exercice 20**

Les informations concernant les adhérents à un club de sport sont enregistrées dans un fichier texte nommé adherent.txt et enregistré dans le dossier « d:\club ». Chaque ligne de ce fichier concerne un adhérent et comporte 4 informations séparées par des virgules à savoir :  
Le N° de la carte d'adhérent, le nom, la date de naissance, le nombre d'heures de sport.  
La longueur d'une ligne de ce fichier ne dépasse pas 60 caractères.

On se propose de :

**A)** Créer à partir de ce fichier, un deuxième fichier de données nommé « d:\club\adherent.dat » où chaque enregistrement correspond à un adhérent et a la structure suivante :

- Le N° de la carte : un entier
- Le nom : une chaîne de 40 caractères
- La date de naissance : une chaîne de 10 caractères
- Les nombres d'heures de sport : un entier

**B)** Informatiser la gestion des adhérents au club, par la création d'une application offrant un menu qui permet les actions ci-dessous selon la valeur d'une lettre saisie :

- La valeur "A", pour ajouter les données relatives à un nouvel adhérent. L'ajout se fera à la fin du fichier.
- La valeur "M", pour modifier les données relatives à un adhérent dont on saisit le numéro de la carte.
- La valeur "S", pour supprimer un adhérent dont on saisit le numéro de la carte.
- La valeur "T", pour trier le fichier dans l'ordre alphabétique croissant des noms des adhérents. Pour trier le fichier, on copiera les données nécessaires dans la mémoire centrale. Une fois triées, les données seront recopiées dans le fichier d'origine.
- La valeur "Q" pour quitter l'application.

### **Exercice 21**

Un médecin veut informatiser la gestion de son cabinet. On se propose de l'aider à travers un programme Pascal permettant d'effectuer les tâches suivantes :

1. Ajouter un nouveau patient
2. Modifier les informations relatives à un patient existant
3. Attribuer un RDV pour un patient donné
4. Afficher les informations relatives à un patient donné
5. Afficher la liste des patients
6. Afficher la liste des patients ayant un RDV à une date donnée
7. Quitter l'application.

Un patient est caractérisé par :

- Un numéro de dossier
- Un nom prénom
- Une date de naissance (jour, mois, année)
- Une date du prochain RDV (jour, mois, année)

### **Exercice 22**

Pour mieux gérer et organiser ses films, le responsable d'un vidéo club décide d'informatiser sa vidéothèque par la création d'un fichier faisant objet d'une base de données de ses films.

Ecrire un programme permettant les actions offertes par les menus suivants:

<b>Menu Fichiers</b>	<b>Menu Edition</b>	<b>Menu Quitter</b>
- Créer un nouveau fichier - Ouvrir un fichier existant - Supprimer un fichier - Fermer un fichier	- Ajouter un film - Modifier un film - Chercher un film - Supprimer un film	- Sortir

## Exercice 1

Écrire une procédure qui affiche les entiers par ordre décroissant, de 10 jusqu'à 1. Proposer une solution itérative et une autre récursive.

## Exercice 2

Soit la procédure itérative suivante :

```
procedure affiche;  
var a, b: integer;  
begin  
  for a := 0 to 3 do  
    for b := 0 to 9 do  
      writeln(a * 10 + b);  
    end;  
end;
```

Transformer cette procédure en une procédure récursive.

## Exercice 3

Écrire une procédure récursive permettant de saisir un entier N pair ( $1 < N < 31$ ).

## Exercice 4

Écrire une procédure récursive permettant de remplir un tableau T par N entiers positifs.

## Exercice 4.2

Écrire une procédure récursive permettant d'afficher un tableau T.

## Exercice 5

Écrire une fonction récursive permettant de calculer la **factorielle** d'un entier N donné.

## Exercice 6

On appelle "**palindrome**" un mot ou une phrase qui se lit de la même façon à l'endroit comme à l'envers, sans tenir compte des espaces.

*Exemple : le mot "ABCBA" est un palindrome.*

Écrire une fonction récursive permettant de vérifier si une chaîne de caractères CH est un palindrome.

## Exercice 7

Écrire une fonction récursive permettant de vérifier la **parité** d'un entier N.

## Exercice 8

Écrire une fonction récursive permettant d'effectuer la **multiplication** de deux entiers positifs (notés p et q) donnés, en utilisant uniquement l'addition entière.

En effet,  $pxq = p+p+p+\dots+p$  (q fois).

## Exercice 9

Écrire une fonction récursive permettant de calculer la **somme** des N premiers entiers.

## Exercice 10

Écrire une fonction récursive permettant de calculer  $X^n$  pour X réel et n entier relatif.

## Exercice 11

Écrire une fonction récursive permettant de calculer le **PGCD** de deux entiers A et B par la méthode d'EUCLIDE puis la méthode de différence.

## Exercice 12

Écrire une fonction récursive permettant de calculer le **PPCM** de deux entiers A et B.

### Exercice 13

La suite de **Fibonacci** est définie par :  $U_n = U_{n-1} + U_{n-2}$  avec  $U_1 = 1$  et  $U_2 = 1$ . Ecrire une fonction récursive permettant de calculer le  $N^{\text{me}}$  terme de la suite de Fibonacci.

### Exercice 14

Écrire une procédure récursive permettant de décomposer un entier N en **facteurs premiers**.  
(Exemple :  $432 = 2 * 2 * 2 * 2 * 3 * 3 * 3$ ).

### Exercice 15

Un nombre **parfait** est un nombre qui est égale à la somme de ses diviseurs sauf lui-même  
exemple : 6 est parfait car  $6 = 1 + 2 + 3$

Écrire une fonction récursive qui calcule la somme de diviseurs d'un entier N sauf lui-même.

### Exercice 16

Un entier supérieur à 1 est dit **premier** s'il n'est divisible que par 1 et par lui-même.

Écrire une fonction récursive qui vérifie si un entier N est premier ou non.

### Exercice 17

Soit l'exponentielle : 
$$e^x \simeq 1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Faire une fonction fact(n) qui renvoie n!

Faire une fonction puiss(x, n) qui renvoie  $x^n$ .

Ecrire une fonction récursive qui calcule la valeur approchée de  $e^x$  en faisant appel aux fonctions fact et puiss.

### Exercice 18

Calculer par récursivité  $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$  pour tout  $n > 0$ .

### Exercice 19

Calculer par récursivité  $S(n) = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + (-1)^n \frac{1}{2n+1}$  pour tout  $n \geq 0$ .

### Exercice 20

Écrire une fonction récursive qui permet de chercher le **maximum** dans un tableau T de n entiers.

Écrire une fonction récursive qui permet de chercher le **minimum** dans un tableau T de n entiers.

### Exercice 21

Écrire une fonction récursive qui détermine la valeur la plus proche d'un entier m donné dans un tableau T de n entiers.

### Exercice 22

Écrire une fonction récursive qui teste l'existence d'un élément donné dans un tableau donné en utilisant une **recherche séquentielle**.

### Exercice 23

Écrire une fonction récursive qui teste l'existence d'un élément donné dans un tableau donné en utilisant une **recherche dichotomique**.

### Exercice 24

Écrire une procédure récursive qui permet de **décaler** tous les éléments d'un tableau d'une position à droite à partir de la position p.

### Exercice 25

Écrire une procédure récursive qui permet **d'inverser** les éléments d'une partie d'un tableau compris entre la position p et n.

### Exercice 26

Écrire une procédure récursive qui permute les  $n$  éléments consécutifs deux à deux d'un tableau  $T$ .

### Exercice 27

Écrire une fonction récursive qui détermine la valeur maximale de deux entiers positifs sans utiliser  $<$  et  $>$ .

### Exercice 28

Soit une chaîne de caractères ; supposons qu'on veuille faire aussi bien la fonction que la procédure qui nous renvoie **l'inverse** de cette chaîne.

### Exercice 29

Écrire une procédure récursive qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri par sélection**.

### Exercice 30

Écrire une procédure récursive qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri par insertion**.

### Exercice 31

Écrire une procédure récursive qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri à bulles**.

### Exercice 32

Écrire une procédure récursive qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri Shell**.

### Exercice 33

Écrire une procédure récursive qui permet de trier un tableau de  $n$  entiers en utilisant la méthode de **tri par fusion**.

### Exercice 34

Écrire une procédure récursive qui supprime toutes les occurrences d'un caractère donné d'une chaîne de caractère.

### Exercice 35

Écrire une fonction qui compte le nombre d'occurrences d'un caractère  $c$  dans une chaîne  $ch$ .

### Exercice 36

Écrire une fonction récursive qui calcule la somme des chiffres d'un nombre entier.

### Exercice 37

Écrire une fonction récursive qui détermine le nombre des chiffres d'un entier.

### Exercice 38

Deux mots sont des anagrammes si l'un est une permutation des lettres de l'autre.

Par exemple les mots suivants sont des anagrammes :

- aimer et maire
- chien et niche

Par définition, on considère que deux mots vides sont des anagrammes.

Proposez une fonction récursive qui permet de savoir si deux mots sont des anagrammes.

### Exercice 39

Écrire une fonction récursive qui **teste** la présence d'un caractère dans une chaîne de caractère.

#### **Exercice 40**

Écrire une fonction récursive nommée **suppr\_car** qui permet de supprimer la première occurrence d'un caractère d'une chaîne.

#### **Exercice 41**

En utilisant les deux fonctions précédentes (**teste** et **suppr\_car**), proposez une fonction récursive **constructible** qui permet de savoir si un mot est constructible à partir d'un ensemble de lettres. Le mot et l'ensemble de lettres sont représentés à l'aide d'une chaîne de caractères. Si une lettre apparaît deux fois dans le mot, il faut qu'elle apparaisse au moins deux fois dans l'ensemble de lettres.

Par exemple :

**Constructible** ("BONJOUR","BJNORUYZ") retournera faux (le 'O' n'apparaissant qu'une seule fois dans "BJNORUYZ")

**Constructible** ("TRALALA","LLAAAAART") retournera vrai

**Constructible** ("","ABC") retournera vrai

#### **Exercice 42**

Le triangle de Pascal est le tableau des coefficients qui sont utilisés pour le développement de certaines expressions comme  $(a+b)^2$  ou  $(a+b)^n$ .

Ce triangle est le suivant :

$$\begin{array}{l} 1 \qquad \qquad \qquad (a+b)^0 = 1 \\ 1 \ 1 \qquad \qquad \qquad (a+b)^1 = 1a + 1b \\ 1 \ 2 \ 1 \qquad \qquad \qquad (a+b)^2 = 1a^2 + 2ab + 1b^2 \\ 1 \ 3 \ 3 \ 1 \qquad \qquad \qquad (a+b)^3 = 1a^3 + 3a^2b + 3ab^2 + 1b^3 \\ 1 \ 4 \ 6 \ 4 \ 1 \qquad \qquad \qquad (a+b)^4 = 1a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + 1b^4 \end{array}$$

Écrire une fonction récursive permettant de déterminer les valeurs du triangle pascal.

#### **Exercice 43**

Écrire une fonction récursive MacCarthy qui calcule MacCarthy(n) selon la définition suivante :

$$\text{Si } n > 100 \quad \text{MacCarthy}(n) = n - 10$$

$$\text{Si } n \leq 100 \quad \text{MacCarthy}(n) = \text{MacCarthy}(\text{MacCarthy}(n+11))$$

#### **Exercice 44**

Écrire une fonction récursive calculant la fonction d'**Ackermann** définie comme suit :

$$Ack(m, n) = \begin{cases} n+1 & \text{si } m = 0 \\ Ack(m-1, 1) & \text{si } m > 0 \text{ et } n = 0 \\ Ack(m-1, Ack(m, n-1)) & \text{si } m > 0 \text{ et } n > 0 \end{cases}$$

Calculer Ack(0,3) et Ack(2,2)

#### **Exercice 45 Évaluation d'une chaîne de caractère**

Soit une chaîne de caractères du type  $s = "5+123-4+67-2"$  ; Écrire une fonction récursive qui évalue cette chaîne de caractères.

#### **Exercice 46 Vers une mini-calculatrice**

Soit une chaîne de caractères du type  $s = "5+3*4/2-5*3+4*7/2"$  ; faisons le programme qui évalue cette chaîne de caractères.

### Exercice 47

Écrire une procédure récursive qui affiche les combinaisons (les différentes permutations de n objets) d'une chaîne de caractères.

Par exemple, les **anagrammes** des lettres de "abc" sont :

"abc", "acb", "bac", "bca", "cab" et "cba".

### Exercice 48

Écrire un programme permettant d'évaluer un nombre romain en son équivalent en décimal. Sachant que les chiffres romains :

- M = 1000
- D = 500
- C = 100
- L = 50
- X = 10
- V = 5
- I = 1

On constate que les nombres s'arrêtaient aux milliers.

**Exemples d'écriture des nombres romains :**

- 4 s'écrit IV.
- 6 s'écrit VI.
- 9 s'écrit IX.
- 15 s'écrit XV.
- 47 s'écrit XLVII.
- 149 s'écrit CXLIX (et non CIL, comme on pourrait le penser) On constate ici la décomposition  $100+40+9 = C + XL + IX$
- 1490 s'écrit MCDXC =  $1000 + 400 + 90 = M + CD + XC$

### Exercice 49

Écrire un programme qui utilise une procédure récursive permettant d'afficher les caractères d'une chaîne sous la forme indiquée dans l'exemple suivant :

**Exemple :** Soit la chaîne "TURBO"

```
TURBO
TURB
TUR
TU
T
```

### Exercice 50

Écrire une fonction booléenne récursive qui teste si deux fichiers d'entiers sauvegardés sur disque dur, sont égaux ou non.

### Exercice 51

Écrire un programme qui utilise une fonction récursive pour trouver le plus petit nombre d'un fichier d'entiers sauvegardés sur disque dur, puis l'affiche à l'écran.

### Exercice 52

On considère la procédure suivante écrite en Pascal :

```
Procédure P(n : integer);
begin
  if (n>0)
  then begin
    P(n div 4);
    writeln(n);
    P(n div 3);
  end;
end;
```

- A) Pour n=5, la sortie de P(n) affiche 5 1 1.
- B) Pour n=5, la sortie de P(n) affiche 1 5 1.
- C) Pour n=7, la sortie de P(n) affiche 1 1 5.
- D) Pour n=9, la sortie de P(n) affiche 2 3 1.
- E) P(n) se termine pour toutes les valeurs entières de n.