

Exercices en Turbo Pascal

Salah Fenni

LES STRUCTURES SIMPLES

Exercice 1

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.

- $N \bmod P * Q$
- $N \bmod P \text{ div } Q$
- $N = P \text{ Or } N \leq Q$
- $\text{SUCC}('N' + 'P')$

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.

- $\text{PRED}(C) + 'e'$
- $\text{ORD}(C) + 2.5$
- $\text{CHR}(\text{ORD}(C) - 32)$
- $\text{CHR}(\text{SUCC}(\text{ORD}(C)))$

c) Si N est une variable entière et X une variable réelle, quelles sont les affectations possibles :

- $X := N$;
- $N := X + 1$;
- $N := \text{TRUNC}(X) + 1$;
- $N := \text{INT}(X) + 1$;
- $N := \text{ROUND}(X) + 1.5$;
- $N := \text{ROUND}(X + 1.5)$;

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

☐ Lire (A)	☐ Lire (45)	☐ Lire (A+B)
☐ Lire ("A")	☐ Lire ("A =", A)	☐ Ecrire (X+2*Y)
☐ Ecrire ("A =", A)	☐ Ecrire (A, " ", B)	☐ Ecrire (A:6:2)
☐ Ecrire (5 mod 7 div 2)	☐ Ecrire ("Saisir un réel")	☐ Ecrire (45)

Exercice 2

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 3

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 4

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 5

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 6

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

Exercice 7

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

Exercice 8

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 9

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

Exercice 10

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 11

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 12

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 13

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 14

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 15

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 16

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 17

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 18

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui calcule la **résistance** équivalente de 3 résistances montées en parallèle.

On rappelle : $1/R = 1/R_1 + 1/R_2 + 1/R_3$

Exercice 19

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 20

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.

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

La surface d'un triangle de côtés a, b et c est donnée par :

$$Surf = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{Où } p = (a+b+c)/2$$

Ecrire un programme Pascal intitulé **Aire_triangle** qui lit les valeurs de a, b et c, calcule et affiche SURF.

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 **ax²+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_LETTRE**, 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.

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              1
            232            121
          34543          12321
        4567654        1234321
      567898765      123454321
    67890109876    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, 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 : 5, 8, 3

```

          B
          B
          B
    A      B
    A      B
    A      B      C
    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** ($X*Y=X+X+X+...$).

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

Exercice 21

On remarque que :

$$\begin{aligned} 12 \times 42 &= 21 \times 24 \\ 12 \times 63 &= 21 \times 36 \\ 12 \times 84 &= 21 \times 48 \end{aligned}$$

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 Π** par la méthode de Wallis, définie par la formule suivante :

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

Ecrire un programme Pascal qui utilise la formule ci-dessus pour déterminer et afficher une valeur approchée de Π 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** qui repose sur le principe suivant :

Si $B=0$ alors $\text{PGCD}(A, B) = A$ sinon $\text{PGCD}(A, B) = \text{PGCD}(B, A \bmod B)$

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** qui repose sur le principe suivant :

Si $A=B$ alors $\text{PGCD}(a, b) = a$ (ou b)

Sinon $\text{PGCD}(A, B) = \text{PGCD}(a-b, b)$ 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 A et B.

Exercice 26

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

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

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

Exercice 27

Un entier supérieur à 1 est dit **premier** s'il n'est divisible que par 1 et par 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 dis **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 $\leq 10^{-4}$. 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 $\left| \frac{x^n}{n!} \right| \leq \epsilon$. 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

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

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 base10 à la base2**).

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.

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 20 entiers. On se propose de **ranger** les éléments de T dans un tableau R de façon à mettre les éléments positifs ou nuls de T au début de R suivis des éléments négatifs.

Exemple :

T						
2	-3	15	6	-9	-1	0

R						
2	15	6	0	-3	-9	-1

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

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 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

1	2	3
4	5	6
7	8	9

Devient

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 20

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

Exercice 21

Une classe est composée de 30 élèves. Le professeur d'arabe Mr Najib veut utiliser l'ordinateur pour faire la moyenne trimestrielle et le rang de ses élèves.

Sachant que la moyenne = $(\text{note1} + 2 * \text{note2}) / 3$, Mr Najib veut afficher les résultats sous forme d'un tableau comportant :

Nom	Prénom	Note 1	Note2	Moyens	Rangs
.....					
.....					

Exercice 22

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 23

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 24

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 25

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 26

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 27

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

Exercice 28

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 du 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 29

Ecrire un programme Pascal qui permet de remplir un tableau T par n caractères, de rechercher dans le tableau la 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).

LES CHAINES DE CARACTERES

Soit S une variable du type chain. Sur combien d'octets est codée S ? Où est stockée la longueur courante de S ? Quelles sont les 2 façons pour connaître la longueur courante de S ?

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.

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

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

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

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

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'

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

Par exemple :

'RESOLUTION DE PROBLEMES' devient 'PROBLEMES DE RESOLUTION'

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.

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

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"

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 sera 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 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 19

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 1:

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

Exercice 2:

Écrire un algorithme, puis 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 :

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

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

$$(a + b i) / (c + d i) = (a c + b d) / (c^2 + d^2) + (b c - a d) i / (c^2 + d^2)$$

Exercice 3:

Écrire un algorithme, puis 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 ≥ 800 Dinars.

Exercice 4:

Écrire un algorithme, puis 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 5:

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

Exercice 6:

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 7:

Écrire un programme Pascal qui permet de :

- Créer un 1er fichier texte f3ligne et saisir 3 lignes
- Créer un 2ème fichier texte f1ligne et saisir une ligne
- Créer un 3ème fichier texte f4ligne et générer son contenu de 4 lignes par recopie du fichier1 et du fichier2.

Exercice 8:

Écrire un programme en Pascal permettant de :

- Saisir et enregistrer des noms des villes dans un fichier dont le nom physique est "c:\ villes.dat"
- Afficher toutes les villes commençant par la lettre "A"
- Pour chaque lettre, afficher la liste des villes dont le nom commence par cette lettre.

Exercice 9:

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

Exercice 10:

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 11:

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

Exercice 12:

On dispose d'un fichier texte, intitulé **source.txt**.

Un tautogramme est un texte dont tous les mots commencent par la même lettre (sans distinction entre majuscule et minuscule).

Exemple : Le lion lape le lait lentement

On se propose de sauvegarder toutes les lignes tautogrammes de ce fichier texte, dans un deuxième fichier texte intitulé **tauto.txt**.

Exercice 13:

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 14:

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 15:

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 16:

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 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:

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

Exercice 19:

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
- Le 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 20:

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 21:

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

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

Menu Fichiers	Menu Edition	Menu Quitter
- 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 22

Les informations relatives à une personne sont :

- Nom, prénom (chaîne) ;
- Téléphone (10 caractères) ;
- Numéro fiscal (entier) ;
- Adresse (rue (entier) ; ville (chaîne) ; code postal (entier non signé)) ;

Ecrivez les analyses et les algorithmes des différents modules d'un programme qui permet de :

- ☐ Remplir un fichier intitulé « **data.dat** » sous la racine **C:/** par des personnes. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Afficher à l'écran les noms et les prénoms des personnes résidant dans une ville donnée.
- ☐ Saisir un numéro de téléphone **TEL** puis chercher et afficher le nom, le prénom et le numéro fiscal de la personne ayant ce numéro de téléphone.
- ☐ Saisir le nom d'une ville puis calculer et afficher le nombre des personnes résidant dans cette ville.

Exercice 23

Une société veut informatiser la gestion de ses employés. Les informations relatives à un employé sont représentées par la structure FICHE suivante :

- Nom_prénom (30 caractères) ;
- Le grade (uniquement **G1**, **G2**, **G3** ou **G4**) ;
- Code fiscal (entier non signé) ;
- Assurance de maladie (lettre **O** pour oui et la lettre **N** pour non) ;

Ecrivez un programme modulaire nommé **GESTION** qui permet :

- ☐ Remplir un fichier intitulé « **info.dat** » sous la racine **C:/** par des Fiches. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ L'affichage de toutes les fiches (sur l'écran) une par une avec une attente jusqu'à l'appui de la touche entrée.
- ☐ La saisie d'un grade et l'affichage de nombre d'employés ayant ce grade ainsi que leur pourcentage par rapport au nombre total des employés.
- ☐ L'affichage de nombre des employés qui possèdent une assurance de maladie et leur pourcentage par rapport au nombre total des employés.
- ☐ L'affichage des noms et prénoms et les grades des employés qui n'ont pas d'assurance de maladie.

Exercice 24

La structure d'un étudiant est :

- Nom, Prénom : chaîne ;
- Numéro de carte d'étudiant NCE : entier non signé ;
- Niveau : (5 caractères) ;
- Date de naissance (jj / mm / aaaa) ;
- Admission : (**O** pour oui et **N** pour non) ;

Ecrivez un programme modulaire nommé **STATISTIQUES** qui permet de :

- ☐ Remplir un fichier intitulé « **institut.dat** » sous la racine **C:/** par des étudiants. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Calculer et afficher le nombre des étudiants qui ont réussi et leur pourcentages par rapport au nombre total des étudiants.
- ☐ Calculer et afficher le nombre des étudiants appartenant à un niveau **NIV** donné et qui ont réussi.
- ☐ Saisir le **NCE** d'un étudiant et afficher son résultat d'admission. Si ce NCE n'existe pas dans le fichier alors le programme doit afficher un message indiquant que cet étudiant n'existe pas.
- ☐ Recopier ce fichier dans l'emplacement « C:/Bac201N/ ».
- ☐ Afficher à l'écran l'enregistrement numéro **NUM**. Où NUM est un entier saisi au clavier représentant le numéro d'ordre de l'enregistrement dans le fichier. Attention au cas échéant lorsque (NUM < 0 !!) ou lorsque (NUM ≥ taille de fichier !!).
- ☐ Supprimer l'enregistrement numéro **NE**. Où NE est un entier saisi au clavier représentant le numéro d'ordre de l'enregistrement que l'on veut le supprimer. Attention au deux cas (NE < 0 !!) et (NE ≥ taille de fichier !!).
- ☐ Afficher à l'écran le nombre d'enregistrements.

Exercice 25

Les caractéristiques d'une voiture sont :

- Marque (10 caractères) ;
- Couleur (10 caractères) ;
- Matricule (20 caractères) ;
- Nombre de chevaux ;

Ecrivez un programme modulaire qui permet de :

- ☐ Remplir un fichier intitulé « **véhicules.dat** » sous **C:/Bac201N** par des voitures. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Chercher puis afficher les marques des voitures possédant **5** chevaux.
- ☐ Afficher les matricules des voitures une couleur saisie au clavier.
- ☐ Saisir une voiture **V** et vérifier si elle existe dans le fichier ou non en se basant sur sa matricule.
- ☐ Effacer le fichier totalement du disque.

Exercice 26

Les informations relatives à une élève sont les suivantes :

- Nom (15 caractères) ;
- Prénom (15 caractères) ;
- Sexe ("garçon" ou "fille") ;
- Moyenne générale ;
- Date de naissance ;

Ecrivez un programme modulaire qui permet de :

- ☐ Remplir un fichier intitulé « **classe.dat** » sous **C:/Bac201N** par des élèves. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Calculer et afficher la moyenne arithmétique de cette classe.
- ☐ Afficher le nombre des garçons et le nombre des filles de cette classe.
- ☐ Dégager le nom et le prénom du meilleur élève de cette classe.
- ☐ Transférer les moyennes des élèves dans un tableau **T**. (on supposera que le nombre d'enregistrements dans le fichier est inférieur à 100).
- ☐ Supprimer tous les enregistrements qui suit l'enregistrement d'ordre **NUM** (on va tronquer le fichier). Où **NUM** est un entier saisi au clavier. Attention au cas échéant lorsque (**NUM** < 0 !!) ou lorsque (**NUM** > taille de fichier !!).

Exercice 27

Dans une clinique, la fiche d'un patient contient les informations suivantes :

- Nom de patient (15 caractères) ;
- Prénom de patient (15 caractères) ;
- Numéro de carte d'identité ;
- Numéro de la salle où il est logé.
- Médecin de patient (nom, prénom, spécialité) ;
- Date de consultation (15 caractères) ;

Ecrivez un programme modulaire qui permet de :

- ☐ Remplir un fichier intitulé « **clinique .dat** » sous **C:/Bac201N** par des patients. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Afficher les noms et les prénoms des patients du médecin « **xxxx** » « **yyyy** ».
- ☐ Modifier le numéro de salle des patients qui sont consulté le "02/10/2007" avec le numéro **S** saisi au clavier.
- ☐ Chercher et d'afficher les noms et les prénoms des patients qui ont été consultés à une date **T** donnée.
- ☐ Saisir un patient **P** et vérifier s'il existe dans ce clinique ou non en se basant sur sa carte d'identité.

Exercice 28

Un hôtel est caractérisé par :

- Nom de l'hôtel ;
- Classe ("*", "***", "****", "*****", "*****") ;
- Ville (15 caractères) ;
- Directeur de l'hôtel (nom, prénom, téléphone) ;

Ecrivez un programme modulaire qui permet de :

- ☐ Ecrire un module qui permet de remplir un tableau **T** par **N** hôtels ($4 \leq N \leq 100$).
- ☐ Transférer les éléments de tableau **T** dans un fichier intitulé « **stat.dat** » sous la racine **C:/ Bac201N**.
- ☐ Afficher les noms d'hôtels et leurs classes d'une ville **V** saisie au clavier.
- ☐ Compter le nombre des villes contenant des hôtels.
- ☐ Dégager à partir du tableau **T** les noms d'hôtels de la ville "**sousse**" ayant comme classe "*****".

Exercice 29

Une bibliothèque emprunte des livres aux élèves. Un livre emprunté est caractérisé par :

- Identifiant (5 caractères) ;
- Titre (20 caractères) ;
- Genre (20 caractères) ;
- Auteur (nom, prénom) ;
- Date d'emprunt ;
- Elève (nom, prénom, classe) ;

Ecrivez un programme modulaire qui permet de :

- ☐ Remplir un fichier intitulé « **biblio.dat** » sous **C:/Bac201N** par des livres. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- ☐ Chercher et d'afficher les noms et les prénoms des élèves qui ont emprunté des livres aujourd'hui.
- ☐ Chercher et d'afficher les noms et les prénoms des élèves qui ont emprunté des livres de genre « informatique ».
- ☐ Le livre ayant pour identifiant « info4 » est emprunté de nouveau à l'élève :
Nom : « **xxxx** »
Prénom : « **yyyy** »
Classe : « **4si** »

Ecrire un module qui effectue les modifications appropriées dans le fichier.

- ☐ Compter et afficher le nombre des élèves de la classe « 4si » qui ont emprunté des livres de la bibliothèque.
- ☐ Saisir le nom, le prénom et la classe d'un élève puis modifier l'enregistrement numéro NUM par ces nouvelles informations (NUM est saisi au clavier).
- ☐ Supprimer le fichier totalement de disque.

Exercice 30

Ecrivez un programme modulaire qui permet de :

- ☐ Créer un fichier texte intitulé « **paragraphe.txt** » sous **C:/Bac201N** et le remplir par des chaînes de caractère saisie au clavier. La saisie s'arrête lorsqu'on tape la chaîne 'FIN' (minuscule ou majuscule). La chaîne de caractère, 'FIN' en l'occurrence ne sera pas stockée dans le fichier.
- Compter et afficher le nombre de lignes **NBL** enregistré dans ce fichier.
Compter et afficher le nombre de caractères **NBC** enregistré dans ce fichier.
Afficher le contenu de F à l'écran.

Exercice 31

Ecrivez un programme modulaire qui permet de :

- ☐ Créer un fichier texte intitulé « **texte.doc** » sous **C:/Bac201N** et le remplir par des chaînes de caractère saisie au clavier. Le nombre de chaînes saisies ne doit pas dépasser 100. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N ».
- Afficher le contenu de F à l'écran.
Transférer le contenu du fichier dans un tableau T.

Exercice 32

Ecrivez un programme modulaire qui permet de :

- ☐ Remplir un fichier texte intitulé « **nombres.txt** » sous **C:/Bac201N** et le remplir par des entiers positifs. La saisie s'arrête selon le choix de l'utilisateur en répondant à la question « voulez-vous continuer ? O/N » ou lorsque le nombre d'entiers saisi atteint 100.
 - ☐ Afficher le contenu de F à l'écran.
 - ☐ Calculer la somme S et le Produit P des entiers de F.
- Question** : Expliquer Comment peut-on **trier** les entiers qui existent dans F
- ☐ Transférer les entiers de fichier F vers un tableau T.
 - ☐ Trier les éléments de T par ordre décroissant.
 - ☐ Recopier les entiers de nouveau dans le fichier F.

Exercice 33

Ecrivez un programme modulaire intitulé **FUSION** qui permet de

- ☐ Fusionner le fichier texte « **texte2.txt** » qui existe sous **C:/Bac201N** dans le fichier « **texte1.txt** » qui existe sous **C:/**. La fusion consiste à copier le contenu de fichier « **texte2.txt** » dans le fichier « **texte1.txt** » et à la suite de son contenu.
- ☐ Afficher le contenu de F à l'écran.

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 $N^{\text{ème}}$ terme de la suite de Fibonacci.

Exercice 14

Ecrire 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 $6=1+2+3$
Ecrire 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.
Ecrire 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

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

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

Exercice 21

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

Exercice 22

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

Exercice 23

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

Exercice 24

Ecrire 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

Ecrire 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 caractères.

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 caractères.

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 :

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

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 :

Si $n > 100$ MacCarthy(n) = $n - 10$

Si $n \leq 100$ MacCarthy(n) = MacCarthy(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 une fonction booléenne récursive qui teste si deux fichiers d'entiers sauvegardés sur disque dur, sont égaux ou non.

Exercice 50

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.

LES STRUCTURES SIMPLES

Exercice 1-d)

[V] Lire (A)
[F] Lire ("A")
[V] Ecrire ("A =", A)
[V] Ecrire (5 mod 7 div 2)

[F] Lire (45)
[F] Lire ("A =", A)
[V] Ecrire (A, " ", B)
[V] Ecrire ("Saisir un réel")

[F] Lire (A+B)
[V] Ecrire (X+2*Y)
[V] Ecrire (A : 6 : 2)
[V] Ecrire (45)

Exercice 2 0) Début Sortie_Inverse 1) Ecrire ("A = "), Lire (A) 2) Ecrire ("B = "), Lire (B) 3) Ecrire ("C = "), Lire(C) 4) Ecrire (C, " ", B, " ", A) 5) Fin Sortie_Inverse	Exercice 3 0) Début Cylindre 1) Ecrire ("Donner le rayon : "), Lire(R) 2) Ecrire ("Donner la hauteur : "), Lire (H) 2) $V \leftarrow \pi R^2 H$ 3) Ecrire ("Volume = ", V) 4) Fin Cylindre
Exercice 4 0) Début Surface_rectangle 1) Ecrire ("Largeur = "), Lire (la) 2) Ecrire ("Longueur = "), Lire (lo) 3) $S \leftarrow la * lo$ 4) Ecrire ("La surface du rectangle dont la longueur mesure ", lo, " m et la largeur mesure ", la, " m, a une surface égale à ", s, " mètres carrés.") 5) Fin Surface_rectangle	Exercice 5 0) Début Piscine 1) Ecrire ("Donner les dimensions de la piscine"), Lire (LO, LA, PR) 2) $V \leftarrow LO * LA * PR$ 3) $EAU \leftarrow V * 1000$ 4) Ecrire ("Le volume = ", V) 5) Ecrire ("Quantité d'eau = ", EAU, " litres") 6) Fin Piscine
Exercice 6 0) Début Aire_Trapeze 1) Ecrire ("Donner les dimensions du trapèze "), Lire (H, B1, B2) 2) $S \leftarrow H * (B1 + B2)$ 3) Ecrire ("La surface = ", S) 4) Fin Aire_trapeze	Exercice 7 0) Début Permut 1) Lire (A, B) 2) $AUX \leftarrow A$ 3) $A \leftarrow B$ 4) $B \leftarrow AUX$ 5) Ecrire ("La nouvelle valeur de A est : ", A) 6) Ecrire ("La nouvelle valeur de B est : ", B) 7) Fin Permut
Exercice 8 0) Début Permut_Circulaire 1) Lire (A, B, C) 2) $AUX \leftarrow A$ 3) $A \leftarrow C$ 4) $C \leftarrow B$ 5) $B \leftarrow AUX$ 6) Ecrire (A, " ", B, " ", C) 7) Fin Permut_Circulaire	Exercice 9 0) Début Permut 1) Lire (X, Y) 2) $X \leftarrow X + Y$ 3) $Y \leftarrow X - Y$ 4) $X \leftarrow X - Y$ 5) Ecrire (X, " ", Y) 6) Fin Permut
Exercice 10 0) Début Division 1) Ecrire ("A = "), Lire (A) 2) Ecrire ("B = "), Lire (B) 3) $Q \leftarrow A \text{ div } B$ 4) $R \leftarrow A \text{ mod } B$ 5) Ecrire ("Le quotient est ", q, " et le reste est ", r) 6) Fin Division	Exercice 11 0) Début Temperature 1) Ecrire ("Donner une température en °C : "), Lire (D) 2) $F \leftarrow 9/5 * D + 32$ 3) Ecrire (D, " °C = ", F, " Fahrenheit") 4) Fin Temperature
Exercice 12 0) Début Mile_marin 1) Ecrire ("Donner le nombre de Km : "), Lire (km) 2) $Mm \leftarrow km / 1.852$ 3) Ecrire (km, " km = ", mm, " miles marins") 4) Fin mile_marin	Exercice 13 0) Début Conversion 1) Ecrire ("Nombres de bits = "), Lire (bit) 2) $Octet \leftarrow bit / 8$ 3) $Kilo \leftarrow octet / 1024$ 4) $Mega \leftarrow kilo / 1024$ 5) $Giga \leftarrow mega / 1024$ 6) Ecrire (octet, kilo, mega, giga) 7) Fin conversion
Exercice 14 0) Début Temps 1) Ecrire ("Donner une durée en secondes "), Lire (T) 2) $H \leftarrow T \text{ div } 3600$ 3) $M \leftarrow (T \text{ mod } 3600) \text{ div } 60$ 4) $S \leftarrow T \text{ mod } 60$ 5) Ecrire (H, " : ", M, " : ", S) 6) Fin Temps	Exercice 16 0) Début Interet_Simple 1) Ecrire ("Donner la somme initiale : "), Lire (SOM) 2) Ecrire ("Donner le taux d'intérêt : "), Lire (TAUX) 3) $INTERET \leftarrow (SOM * TAUX / 100) * 5$ 4) $VA \leftarrow SOM + INTERET$ 5) Ecrire ("Après 5 ans la somme sera = ", VA) 6) Fin Interet_Simple
Exercice 17 0) Début Consommation 1) Ecrire ("Nombre de Km parcouru : "), Lire(Y) 2) Ecrire ("Nombre de litres d'essence : "), Lire (L) 3) $TAUX \leftarrow (L * 100) / Y$ 4) Ecrire ("Taux de consommation est = ", TAUX, " %") 5) Fin Consommation	Exercice 18 0) Début Résistance 1) Ecrire ("Donner les trois résistances : "), Lire (R1, R2, R3) 2) $R \leftarrow 1 / (1/R1 + 1/R2 + 1/R3)$ 3) Ecrire ("La résistance équivalente est = ", R) 4) Fin Résistance

Exercice 15	Exercice 19
0) Début Futur 1) Ecrire ("Donner un verbe du 1er groupe : "), Lire (verbe) 2) Ecrire ("Je ", verbe, "ai") 3) Ecrire ("Tu ", verbe, "as") 4) Ecrire ("Il ou elle ", verbe, "a") 5) Ecrire ("Nous ", verbe, "ons") 6) Ecrire ("Vous ", verbe, "ez") 7) Ecrire ("Ils ou elles ", verbe, "ont") 8) Fin Futur	0) Début NBR_3 1) Ecrire ("Saisir un entier formé de 3 chiffres non nuls : "), Lire (N) 2) $c \leftarrow n \text{ div } 100$ 3) $d \leftarrow n \text{ mod } 100 \text{ div } 10$ 4) $u \leftarrow n \text{ mod } 10$ 5) $r1 \leftarrow c*100+u*10+d$ 6) $r2 \leftarrow u*100+d*10+c$ 7) $r3 \leftarrow u*100+c*10+d$ 8) $r4 \leftarrow d*100+c*10+u$ 9) $r5 \leftarrow d*100+u*10+c$ 10) Ecrire (N, r1, r2, r3, r4, r5) 11) Fin NBR_3
Exercice 20	
0) Début Sup_Inf 1) Ecrire ("A = "), Lire (A) 2) Ecrire ("B = "), Lire (B) 3) $SUP \leftarrow (A + B + \text{abs}(A - B)) \text{ div } 2$ 4) $INF \leftarrow (A + B - \text{abs}(A - B)) \text{ div } 2$ 5) Ecrire ("Valeur sup = ", SUP, " Valeur inf = ", INF) 6) Fin Sup_Inf	

LES STRUCTURES CONDITIONNELLES

Exercice 1

```
Program Min2 ;
Uses Wincrt ;
Var a, b, min : Integer ;
Begin
  Writeln ('Saisir deux entiers : ') ;
  Readln (a, b) ;
  IF a<b Then min := a
    Else min := b ;
  Writeln ('La plus petite valeur est : ', min) ;
End.
```

Exercice 3

```
Program Racine ;
Uses Wincrt ;
Var x : Real ;
Begin
  Write ('Saisir un réel ') ; Readln (x) ;
  IF x >= 0
    Then Writeln ('Racine carrée ', x, ' = ', sqrt(x))
    Else Writeln ('Donnée incorrecte') ;
End.
```

Exercice 6

```
Program Parite ;
Uses Wincrt ;
Var N : Integer ;
Begin
  Writeln ('Donner un entier ') ; Readln (n) ;
  IF n mod 2 = 0
    Then Writeln (n, ' est pair')
    Else Writeln (n, ' est impair') ;
End.
```

Exercice 7

```
Program Chez_la_fourmi;
Uses Wincrt;
label 1,2 ;
Var na, nb, err : Integer;
a, b, z : Char;

Begin
  1: Writeln ('Nombre de doigts montrés par le joueur A');
  a:=readkey;
  Writeln ('Nombre de doigts montrés par le joueur B');
```

Exercice 9

```
Program Sup_Inf;
Uses Wincrt ;
Var a, b : Integer ;
sie : String ;

Begin
  Writeln ('Saisir deux entiers : ') ;
  Readln (a, b) ;
  IF a>b
    Then sie := ' est supérieur à '
    Else IF a<b
      Then sie := ' est inférieur à '
      Else sie := ' est égal à ' ;
  Writeln (a, sie, b) ;
End.
```

Exercice 2

```
Program Max3 ;
Uses Wincrt ;
Var a, b, c, maxi : Integer ;
Begin
  Writeln ('Saisir trois entiers : ') ;
  Readln (a, b, c) ;
  maxi := a ;
  IF b>maxi Then maxi := b ;
  IF c>maxi Then maxi := c ;
  Writeln ('La plus grande valeur est : ', maxi) ;
End.
```

Exercice 4

```
Program Aire_Triangle ;
Uses Wincrt ;
Var a, b, c, Surf, P : Real;
Begin
  Writeln ('Donner 3 réels positifs :'); Readln (a, b, c);
  IF (a+b=c) Or (a+c=b) Or (b+c=a)
    Then Writeln ('Il ne s'agit pas d'un triangle')
    Else Begin
      P := (a+b+c)/2;
      Surf := sqrt (P*(P-a)*(P-b)*(P-c));
      End;
  Writeln ('Aire de triangle = ', Surf:4:2);
End.
```

Exercice 5

```
Program Abs_diff;
Uses Wincrt;
Var a, b, z : Integer;
Begin
  Write ('Donner deux entiers : '); Readln (a, b);
  IF (a-b) < 0 Then z:=b-a
    Else z:=a-b;
  Writeln ('valeur absolue de a-b = ', z);
End.
```

```
b:=readkey;
VAL (a, na, err);
VAL (b, nb, err);
IF (na+nb) mod 2 =0
  Then Writeln ('Le joueur A gagne.')
  Else Writeln ('Le joueur B gagne.');
```

Writeln ('Voulez vous jouer encore ? (o/n)'); Readln (z);
IF z='n' Then goto 2 Else goto 1 ;
2:End.

Exercice 8

```
Program Invite ;
Uses Wincrt ;
Var titre, foulén, term1, term2 : String ;
Begin
  Write ('Titre = ') ; Readln (titre) ;
  Write ('Votre nom = ') ; Readln (foulén) ;
  IF titre = 'Mr'
    Then Begin
      term1 := 'e' ;
      term2 := '' ;
    End
  Else IF (titre = 'Mlle') Or (titre='Mme')
    Then Begin
      term1 := 'a' ;
      term2 := 'e' ;
    End ;
  Writeln (titre, ' ', foulén, ' ', soyez l', term1,
    ' bienvenu', term2) ;
End.
```

Exercise 10

```

Program Ordre ;
Uses Wincrt ;
Var e1, e2, petit, grand : Integer ;
Begin
  Writeln ('Saisir deux entiers : ') ; Readln (e1, e2) ;
  petit := e1 ;
  grand := e2 ;
  IF e1>e2 Then Begin
    petit := e2 ;
    grand := e1 ;
  End;
  Writeln (petit, grand:3) ;
End.

```

Exercise 12

```

Program Equa_1d ;
Uses Wincrt ;
Var a, b : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  IF a <> 0
    Then Writeln ('x = ', -b/a)
    Else IF b = 0
      Then Writeln ('IR')
      Else Writeln ('{ }') ;
End.

```

Exercise 11

```

Program Tri;
Uses Wincrt;
Var a, b, c : Integer;
(*****)
Procedure permut (Var x, y : Integer);
Var aux : Integer;
Begin
  aux:=x;
  x:=y;
  y:=aux;
End;
(*****)

```

```

Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  Write ('c = ') ; Readln (c) ;
  IF a>b Then permut (a, b);
  IF b>c Then permut (b, c);
  IF a>b Then permut (a, b);
  Writeln (a, b:4, c:4);
End.

```

Exercise 13

```

Program Equa_2d ;
Uses Wincrt ;
Var a, b, c, delta : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  Write ('c = ') ; Readln (c) ;
  IF a = 0 {équation 1er degré}
    Then IF b = 0
      Then IF c = 0
        Then Writeln ('IR')
        Else Writeln ('{ }')
      Else Writeln ('x = ', -c/b)
    Else delta := sqr (b) - 4*a*c ;
  IF delta = 0 {solution réelle double}
    Then Writeln ('x1=x2= ', -b/ (2*a))
    Else IF delta > 0 {deux solutions réelles}
      Then Begin
        Writeln ('x1= ', (-b-sqrt (delta))/ (2*a)) ;
        Writeln ('x2= ', (-b+sqrt (delta))/ (2*a)) ;
      End
    Else Writeln ('Deux solutions complexes') ;
End.

```

Exercise 14

```

Program Inequation ;
Uses Wincrt ;
Var a, b : Real ;
Begin
  Write ('a = ') ; Readln (a) ;
  Write ('b = ') ; Readln (b) ;
  IF a>0
    Then Writeln ('x < ', -b/a)
    Else IF a<0
      Then Writeln ('x > ', -b/a)
      Else IF b<0
        Then Writeln ('IR')
        Else Writeln ('Impossible') ;
End.

```

Exercise 15

```

Program Touche ;
Uses Wincrt;
Var c : Char ;
nature : String;
Begin
  Writeln ('Taper sur une touche'); Readln (c);
  Case c of
    'a'..'z','A'..'Z' : IF UPCASE(c) in ['A','E','I','U','O','Y']
      Then nature := 'Voyelle'
      Else nature := 'Consonne';
    '0'..'9' : nature := 'Chiffre';
    Else nature := 'Symbole';
  End;
  Writeln (nature);
End.

```

Exercise 16

```

Program Calculette ;
Uses Wincrt ;
Var a, b : Real ;
op : Char ;
Begin
  Readln (a) ; Readln (op) ; Readln (b) ;
  Case op of
    '+' : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a+b:3:2) ;
    '-' : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a-b:3:2) ;
    '*' : Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a*b:3:2) ;
    '/' : IF b = 0
      Then Writeln ('impossible')
      Else Writeln (a:3:2,' ',op,' ',b:3:2,' = ',a/b:3:2);
    Else Writeln ('Opérateur incorrect');
  End ;
End.

```

Exercice 17

```

Program Bulletin ;
Uses Wincrt ;
Var moy : Real ;
    me, dec : String ;

Begin
  Write ('Donner la moyenne annuelle : ') ; Readln (moy) ;
  IF moy < 10
  Then Begin
    dec := 'Redouble' ;
    me := '' ;
  End
Else Begin
  dec := 'Admis' ;
  IF moy < 12
  Then me := 'Passable'
  Else IF moy < 14
  Then me := 'Assez bien'
  Else IF moy < 16
  Then me := 'Bien'
  Else IF moy < 18
  Then me := 'Très bien'
  Else me := 'Excellent' ;
End ;
Writeln ('Moyenne = ', moy, ' Décision = ', dec,
  ' Mention = ', me) ;

End.

```

Exercice 18

```

Program nbr_jours_mois;
Uses wincrt;
Var nbj, mm, an : integer;

Begin
  Write ('N° du mois : ') ; readln (mm) ;

  nbj :=31;

  if mm in [4,6,9,11]
  then nbj :=30
  else begin
    write ('Année : '); readln (an);
    nbj := 28;
    if (an mod 4=0)and((an mod 100<>0)or(an mod 400=0))
    then nbj := 29;
  end;

  writeln ('Le nombre de jours du mois saisi est : ', nbj);
End.

```

Exercice 20

```

Program Anciennete ;
Uses Wincrt ;
Var ji, mi, ai, jf, mf, af, jj, mm, aa : Integer ;
Begin
  Write ('Donner la date initiale : ') ;
  Readln (ji, mi, ai) ;
  Write ('Donner la date finale : ') ;
  Readln (jf, mf, af) ;
  IF ji > jf
  Then Begin
    jf := jf + 30 ;
    mf := mf - 1 ;
  End ;
  IF mi > mf
  Then Begin
    mf := mf + 12 ;
    af := af - 1 ;
  End ;
  jj := jf - ji ;
  mm := mf - mi ;
  aa := af - ai ;
  Writeln (aa, ' années ', mm, ' mois ', jj, ' jours') ;

End.

```

Exercice 19

```

Program Date ;
Uses Wincrt ;
Var mm, err : Integer ;
    date, jj, aa, mois : String [10] ;
Begin
  Writeln ('Saisir une date sous la forme jj/mm/aaaa') ;
  Readln (date) ;
  jj := COPY (date, 1, 2) ;
  aa := COPY (date, 7, 4) ;
  VAL (COPY (date, 4, 2), mm, err) ;
  Case mm of
    1 : mois := 'Janvier' ;
    2 : mois := 'Février' ;
    3 : mois := 'Mars' ;
    4 : mois := 'Avril' ;
    5 : mois := 'Mai' ;
    6 : mois := 'Juin' ;
    7 : mois := 'Juillet' ;
    8 : mois := 'Août' ;
    9 : mois := 'Septembre' ;
    10 : mois := 'Octobre' ;
    11 : mois := 'Novembre' ;
    12 : mois := 'Décembre' ;
  End ;
  Writeln (jj, mois:2, aa:2) ;

End.

```

Exercice 21

```

Program Lendemain;
Uses Wincrt;
Var j, m, a : Integer;
Begin
  Writeln ('Donner une date sous forme jj mm aa');
  Readln (j, m, a);
  Case m Of
    1, 3, 5, 7, 8, 10,12 : IF j<31
      Then j:=j+1
      Else IF m = 12
        Then Begin
          j:=1;
          m:=1;
          a:=a+1;
        End
        Else Begin
          j:=1;
          m:=m+1;
        End;
    4, 6, 9,11: IF j<30
      Then j:=j+1

```

```

      Else Begin
        j:=1;
        m:=m+1;
      End;
    2 : IF a mod 4 =0
      Then IF j<29
        Then j:=j+1
        Else Begin
          j:=1;
          m:=3;
        End
      Else IF j<28
        Then j:=j+1
        Else Begin
          j:=1;
          m:=3;
        End;
      End;
    Writeln ('La date du lendemain est : ',j,'/',m,'/',a);
  End.

```

Exercise 22

```

Program  Toute_Lettre;
Uses    Wincrt;
Const
    Chiffres : Array[0..19] of String = ('','un','deux','trois','quatre','cinq','six','sept','huit','neuf','dix',
                                           'onze','douze','treize','quatorze','quinze','seize','dix-sept','dix-huit','dix-neuf');
    Dizaines : Array[2..9] of String = ('vingt','trente','quarante','cinquante','soixante','','quatre-vingt','');
Var
    n : Integer;
    result : String;
Begin
    Writeln ('Donner un entier entre 0 et 99'); Readln (n);
    Case n of
        0..19 : result := Chiffres [n];
        20..69,80..89: IF ((n mod 10 = 1) and (n<>81))
            Then result := Dizaines [n div 10] + ' et ' + Chiffres [n mod 10]
            Else result := Dizaines [n div 10] + ' ' + Chiffres [n mod 10];
        70..79,90..99: IF (n = 71)
            Then result := Dizaines [n div 10 -1] + ' et ' + Chiffres [n mod 10 + 10]
            Else result := Dizaines [n div 10 -1] + ' ' + Chiffres [n mod 10 + 10];

    End;

    IF n=0 Then Writeln ('zéro')
    Else Writeln (result);
End.

```

Exercise 23

```

Program  jour_semaine;
Uses    wincrt;
Var
    day, month, year, dayyear, daymonth, weekday, cm:integer;
    jj:string;
Begin
    writeln ('Donner le jour'); readln (day);
    writeln ('Donner le mois'); readln (month);
    writeln ('Donner l''année'); readln (year);
    dayyear:=(year-1)*365 + ((year-1) div 4);
    daymonth:=0;
    for cm:=1 to (month-1) do
        case cm of
            1, 3, 5, 7, 8, 10, 12 : daymonth:=daymonth+31;
            4, 6, 9, 11 : daymonth:=daymonth+30;
            2 : if (year mod 4=0) and ((year mod 100<>0) or (year mod 400 =0))
                then daymonth:=daymonth+29
                else daymonth:=daymonth+28;
        end;
    end;
    weekday:=(dayyear+daymonth+day) mod 7;
    case weekday of
        0:jj:='Dimanche';
        1:jj:='Lundi';
        2:jj:='Mardi';
        3:jj:='Mercredi';
        4:jj:='Jeudi';
        5:jj:='Vendredi';
        6:jj:='Samedi';
    end;
    writeln ('Le jour correspondant est ', jj);
End.

```

LES STRUCTURES ITERATIVES

Exercice 1

```
Program Alphabet;
Uses Wincrt ;
Var c : Char ;
Begin
  FOR c:= 'A' To 'Z' Do Write (c:2);
  Writeln ;
  FOR c:= 'Z' Downto 'A' Do Write (c:2);
End.
```

Exercice 3

```
Program Somme_Produit ;
Uses Wincrt ;
Var s, i : Integer ; p : Real ;
Begin
  S := 0 ; P := 1 ;
  FOR i:= 1 To 20 Do
    Begin
      s := s + i ;
      p := p * i ;
    End ;
  Writeln ('Somme = ', s);
  Writeln ('Produit = ', p:2:2);
End.
```

Exercice 4

```
Program Suite ;
Uses Wincrt ;
Var som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  i := 1 ;
  Repeat
    som := som + u ;
    u := u + 3 ;
    i := i+1 ;
  Until (i>100) ;
  Writeln (som) ;
End.
```

Exercice 5

```
Program Pythagore ;
Uses Wincrt ;
Const n = 9 ;
Var i, j : Byte ;
Begin
  FOR i:=1 To n Do
    Begin
      FOR j:=1 To n Do Write (i * j : 4) ;
      Writeln ;
    End ;
End.
```

Exercice 7

```
Program Moy_Notes;
Uses Wincrt;
Var i, n : Integer;
    note, s : Real;
Begin
  Write ('Combien de notes : '); Readln (n);
  s:=0;
  FOR i:=1 To n Do
    Begin
      Write ('Note ', i, ' : ');
      Readln (note);
      s := s+note;
    End;
  Writeln ('Moyenne de ces ', n, ' notes : ', s/n:2:2);
End.
```

Exercice 2

```
Program Table3;
Uses Wincrt ;
Const n = 10 ;
Var i : Integer ;
Begin
  FOR i:= 1 To n Do
    Writeln ('3*',i,' = ',3*i);
End.
```

Exercice 4

```
Program Suite ;
Uses Wincrt ;
Var som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  FOR i := 1 To 100 Do
    Begin
      som := som + u ;
      u := u + 3 ;
    End ;
  Writeln (som) ;
End.
```

Exercice 4

```
Program Suite ;
Uses Wincrt ;
Var som, i, u : Integer ;
Begin
  som := 0 ;
  u := 2 ;
  i := 1 ;
  While (i<=100) Do
    Begin
      som := som + u ;
      u := u+3 ;
      i := i+1 ;
    End ;
  Writeln (som) ;
End.
```

Exercice 6

```
Program Pyramide;
uses wincrt;
const N=4;
var i, j : integer;
begin
  for i:=0 to n do
    begin
      for j:=i+1 to n do write (' ');
      for j:=-i to i do write ('*');
      writeln;
    end;
end.
```

Exercice 8

```
Program Factoriel ;
Uses Wincrt ;
Var i, n : Byte ;
    fact : Real ;
Begin
  Repeat
    Writeln ('Saisir un entier');
    Readln (n) ;
  Until n IN [0..255] ;

  fact := 1 ;
  FOR i := 2 To n Do fact := fact * i ;

  Writeln (n, ' ! = ', fact) ;
End.
```

Exercise 10

```

Program Diviseurs;
Uses Wincrt;
Var n, m, r : Integer;
Begin
  Writeln ('Donner un entier');
  Readln (n);
  m:=n;
  Repeat
    r:= m mod 10;
    m:= m div 10;
    IF (n mod r = 0) Then Write (r, ' ');
  Until m=0;
End.

```

Exercise 11

```

Program Som_15;
Uses Wincrt;
Var i, j, k : Integer;
Begin
  FOR i:=1 TO 9 DO
    FOR j:=1 TO 9 DO
      FOR k:=1 TO 9 DO
        IF (i+j+k=15)
          Then Begin
            Writeln (i, ' ', j, ' ', k);
            Readln ;
          End;
      End;
    End;
  End.

```

Exercise 9

```

Program Jeu ;
Uses Wincrt ;
Label 1, 2 ;
Var np, nc, essai : Integer ;
    z : Char ;
Begin
  1: Clrscr ;
  Randomize ;
  nc := Random (100) +1 ;
  essai := 0 ;
  Repeat
    essai := essai+1 ;
    Write ('Essai numéro ', essai, 'Votre nombre : ':20);
    Readln (np) ;

```

```

    IF np > nc Then Writeln ('C'est grand')
    Else IF np < nc Then Writeln ('C'est petit')
    Else Writeln ('Bravo vous avez gagné !!') ;
  Until (np = nc) Or (essai = 7) ;

  IF np<>nc
    Then Writeln ('Perdu, le nombre cherché est : ', nc);

  Writeln ('Voulez vous jouer encore ? (o/n)');
  Readln (z);
  IF z='n' Then goto 2 Else goto 1 ;
2:End.

```

Exercise 12

```

Program Histogramme;
Uses Wincrt;
Var a, b, c, max, i : Integer;
Begin
  Writeln ('Entrer trois entiers compris entre 0 et 15');
  Readln (a, b, c);
  max:=a;
  IF b>max Then max:=b;
  IF c>max Then max:=c;
  FOR i:= max Downto 1 Do
    Begin
      IF i>a Then Write (' ')
      Else Write ('A');
      IF i>b Then Write (' ':4)
      Else Write ('B':4);
      IF i>c Then Writeln (' ':4)
      Else Writeln ('C':4);
    End;
  End.

```

```

Program histogramme;
Uses wincrt;
Var n1, n2, n3:integer;
Procedure lecture (var n:integer);
Begin
  writeln ('Entrer trois entiers compris entre 0 et 15');
  repeat readln(n) until n in [0..15] ;
End;
Procedure affiche (n,c:integer; ca:char);
Var l,i:integer ;
Begin
  l:=21; {numéro de ligne}
  for i:=1 to n do
    begin
      gotoxy(c,l);
      writeln(ca);
      l:=l-1;
    end;
  End;
Begin
  lecture(n1); lecture(n2); lecture(n3);
  affiche(n1,10,'A'); affiche(n2,14,'B'); affiche(n3,18,'C');
End.

```

Exercise 13

```

Program Som_Chiffres;
Uses Wincrt;
Var n, som, r : Integer;
Begin
  Writeln ('Donner un entier'); Readln (n);
  som:=0;
  Repeat
    r:= n mod 10;
    som:=som+r;
    n:= n div 10;
  Until n=0;
  Writeln ('La somme de chiffres est : ', som);
End.

```

Exercise 14

```

Program Nbr_Cube;
Uses Wincrt;
Var k, c, d, u : Integer;
Begin
  FOR k:=100 To 999 Do
    Begin
      c:=k div 100 ;
      d:=(k div 10) mod 10;
      u:=k mod 10 ;
      IF (u*u*u+d*d*d+c*c*c) = K
        Then Writeln (k, ' est un nombre cubique');
    End;
  End.

```

Exercise 15

```

Program Somme ;
Uses Wincrt;
Var n, i : Integer ;
    s1, s2, s3 : Real ;
Begin
  Repeat
    Write ('Saisir un entier impair : '); Readln (n);
  Until odd (n);
  s1:=0; s2:=0; s3:=0;
  FOR i:=1 To n Do
    IF odd (i) Then s2 := s2 + 1/i
    Else s3 := s3 - 1/i ;
    s1:= s2 - s3 ;
    Writeln (s1:8:2, s2:8:2, s3:8:2);
  End.

```

Exercise 17

```

Program Probabilite;
Uses Wincrt ;
Const n = 12; essai = 100;
Var d1, d2, d3, cumul, i : Byte;
Begin
  Randomize;
  cumul :=0;
  FOR i:=1 To essai Do
    Begin
      d1 := 1 + Random (6);
      d2 := 1 + Random (6);
      d3 := 1 + Random (6);
      IF (d1+d2+d3 = n) Then cumul := cumul +1;
    End;
  Writeln ('Probabilité est : ', cumul/essai :5:2);
End.

```

Exercise 19

```

Program Multiplication_Addition ;
Uses Wincrt ;
Var x, y, s, aux, i : Integer ;
Begin
  Writeln ('Donner deux entiers'); Readln (x, y) ;
  Write (x, ' * ', y, ' = ');
  IF abs(y)>abs(x) Then Begin
    aux := x ;
    x := y ;
    y := aux ;
  End ;
  IF y<0 Then Begin
    y:= -y ;
    x:= -x ;
  End ;
  s := 0 ;
  FOR i:=1 To y Do s := s+x ;
  Writeln (s) ;
End.

```

Exercise 21

```

Program produits;
uses wincrt;
var a, b, c, d:integer;
begin
  for a:=1 to 9 do
    for c:=a to 9 do
      for b:=c downto a do
        for d:=c downto a do
          if ((10*a+b)*(10*c+d) = (10*b+a)*(10*d+c))
            and (a<>b) and (b<>c)
          then Writeln (a, b, ' * ', c, d, ' = ', b, a, ' * ', d, c) ;
        end.
      end.
    end.
  end.

```

Exercise 16

```

Program Syracuse ;
Uses Wincrt;
Var n, i, s : Integer ;
Begin
  Writeln ('Saisir un entier > 0 '); Readln (n);
  S:=n;
  FOR i:=1 To 50 Do
    Begin
      Write (S, ' ');
      IF S mod 2 = 0
        Then S:= S div 2
        Else S:= 3*S+1;
    End;
  End.

```

Exercise 18

```

Program Puissance_n ;
Uses Wincrt ;
Var n, k : Integer ;
    y, x : Real ;
Begin
  Write ('Saisir un nombre réel : '); Readln (x) ;
  Write ('Saisir la puissance n : '); Readln (n) ;
  y := 1 ;
  FOR k := 1 To abs (n) Do y := y * x ;
  IF n<0 Then y := 1/y ;
  Writeln (x:5:2, ' puissance ', n, ' = ', y:5:2) ;
End.

```

Exercise 20

```

Program Suite ;
Uses Wincrt ;
Var i, n : Integer ; s, invfact : Real ;
Begin
  Writeln ('Donner un entier'); Readln (n) ;
  s := 1 ;
  invfact := 1 ;
  FOR i := 1 To n Do
    Begin
      invfact := invfact/i ;
      s := s + invfact ;
    End ;
  Writeln (s:5:2) ;
End.

```

Exercise 22

```

PROGRAM PI_WALLIS;
uses wincrt;
var p, r, diff : real;
    i : longint;
begin
  p:=1; i:=0;
  repeat
    i:=i+2;
    r:=i/(i-1)*i/(i+1);
    diff:= (p*r) - p;
    p:=p*r;
  until abs (diff) < 1e-8;
  writeln ('par la formule de wallis pi = ', 2*p:2:7);
end.

```

Exercise 23

```

Program PGCD_Euclide;
Uses Wincrt;
Var a, b, r : Integer;
Begin
  Repeat
    Writeln ('Saisir deux entiers > 0'); Readln (a, b);
  Until (a>0) and (b>0) ;
  While b<>0 Do
    Begin
      r := a mod b;
      a := b;
      b := r;
    End;
  Writeln ('PGCD = ', a);
End.

```

Exercise 24

```

Program PGCD_Diff;
Uses Wincrt;
Var a, b : Integer;
Begin
  Repeat
    Writeln ('Saisir deux entiers >0 '); Readln (a, b);
  Until (a>0) and (b>0) ;
  While a<>b Do
    IF a>b Then a:= a-b
    Else b:= b-a ;

  Writeln ('PGCD = ', a) ;
End.

```

Exercise 25

```

Program PPCM_AB;
uses wincrt ;
var ppcm, a, b, aux : integer;
begin
  repeat
    writeln ('saisir deux entiers > 0');
    readln (a, b) ;
  until (a>0) and (b>0);
  if a < b then begin
    aux:= a;
    a := b;
    b := aux;
  end;
  ppcm := a;
  while (ppcm mod b <> 0) do ppcm := ppcm + a;
  writeln ('ppcm = ', ppcm);
end.

```

Exercise 25

```

Program PPCM_AB;
uses wincrt ;
var c, a, b, aux: integer;
begin
  repeat
    writeln ('saisir deux entiers > 0');
    readln (a, b) ;
  until (a>0) and (b>0);
  if a < b then begin
    aux:= a;
    a := b;
    b := aux;
  end;
  c:=0;
  repeat
    c:=c+1;
  until (a*c) mod b = 0;
  writeln ('ppcm = ', a*c);
end.

```

Exercise 26

```

Program Fibonacci ;
uses wincrt ;
var k, f0, f1, f2 : integer ;
begin
  f0 := 1 ; f1 := 1 ;
  write (f0, ' ', f1, ' ');

  for k := 2 to 19 do
    begin
      f2 := f1+f0 ;
      f0 := f1 ;
      f1 := f2 ;
      write (f2, ' ');
    end ;
end.

```

Exercise 27

```

Program Nbre_Premiers ;
uses wincrt ;
var nb, i : integer ;
begin
  for nb := 2 to 400 do
    begin
      i := 2 ;
      while (nb mod i <> 0) and (i <= nb div 2) do i:= i+1 ;
      if (i > nb div 2) then write (nb:4) ;
    end ;
end.

```

Exercise 28

```

Program Parfait ;
uses wincrt;
var nb, d, som, a, b : integer;
begin
  repeat
    Readln(a,b);
  until (2<a) and (a<b);
  for nb:=a to b do
    begin
      som:=0;
      for d:=1 to (nb div 2) do
        if (nb mod d = 0) then som:=som+d ;
      if nb=som then writeln (nb, ' est parfait');
    end;
end.

```

Exercise 32

```

Program Exponentiel;
uses wincrt;
var x, s, epsilon, p, f:real;
    i:integer;
begin
  write ('epsilon = '); readln (epsilon);
  write ('x = '); readln (x);
  s:=1; i:=1; p:=1; f:=1;
  repeat
    p:=p*x;
    f:=f*i;
    s:=s+p/f;
    i:=i+1;
  until abs(p/f) <= epsilon;
  writeln ('expn = ', s:2:10);
end.

```

Exercise 29

```

Program Amis;
uses wincrt;
var m, n, sdn, sdm : integer;
(*****)
function diviseurs (x : integer) : integer;
var sdx, i : integer;
begin
  sdx:=1;
  for i:=2 to (x div 2) do
    if (x mod i) = 0 then sdx := sdx+i;
  diviseurs:=sdx;
end;
(*****)
begin
  for m:=1 to 1000 do
    for n:=1 to 1000 do
      begin
        sdn := diviseurs (n);
        sdm := diviseurs (m);
        if (sdm=n) and (sdn=m)
          then writeln (n, ' et ', m, ' sont amis') ;
      end;
    end;
  end.

```

Exercise 31

```

Program calcul_sinus;
uses wincrt;
var x:real;
(*****)
function sinus (x:real) : real;
var som, term1, term2, i:real;
begin
  som:=x;
  term2:=x;
  i:=1;
  repeat
    i:=i+2;
    term1:=term2;
    term2:=term1 * -sqr(x) / (i*(i-1));
    som:=som+term2;
  until abs (term2-term1) <= 0.0001;
  sinus:=som;
end;
(*****p.p*****)
begin
  repeat
    write ('donner un réel x ');
    readln(x);
  until (0<= x) and (x <= 2*Pi);
  write ('sin(', x:1:2,') = ', sinus(x):10:10);
end.

```

Exercise 34

```

Program Combinaison ;
Uses Wincrt ;
Var cnp : Real ;
    n, p : Integer ;
(*****)
Function Fact (x : Integer) : LongInt ;
Var f : LongInt ; i : Integer ;
Begin
  f := 1 ;
  FOR i := 2 To x Do f := f * i ;
  fact := f ;
End ;
(*****)
Begin
  Repeat
    Writeln ('Donner deux entiers : ');
    ReadLn (p, n);
  Until (0<p) and (p<n) ;
  cnp := fact (n) / (fact (p) * fact (n-p)) ;
  Writeln ('Combinaison = ', cnp :4:2) ;
End.

```

Exercise 30

```

Program Facteur_Premier ;
uses wincrt ;
type tab = array [1..100] of integer ;
var fp : tab ;
    n, i, f : integer ;
begin
  repeat
    writeln ('donner un entier entre 2 et 1000') ; readln (n) ;
  until (n>=2) and (n<=1000) ;
  write (n, ' = ');
  i := 2 ; f:=0;
  repeat
    if (n mod i = 0)
      then begin
        n:= n div i;
        f:=f+1;
        fp[f] := i;
      end
      else i:=i+1;
  until (n=1);
  write (fp[1]);
  for i:=2 to f do write (' * ', fp[i]);
end.

```

Exercise 33

```

Program Somme;
Uses Wincrt;
Var i, n : Integer; s : Real;
(*****)
Procedure saisie (Var m : Integer);
Begin
  Repeat
    Writeln ('Donner un entier positif');
    Readln (m);
  Until m>0;
End;
(*****)
Function puissance (x : Integer):Longint;
Var j : Integer ; p : Longint;
Begin
  p:=1;
  FOR j:=1 To x Do p:=p*x;
  puissance:=p;
End;
(*****)
Begin
  saisie (n);
  s:=0;
  FOR i:=1 To n Do s:= s + (2*i-1) / puissance (i);
  Writeln ('la somme = ', s:2:3);
End.

```

Exercise 34

```

Program Combinaison ;
Uses Wincrt ;
Var cnp : Real ; n, p, i : Integer ;
    Nf, pf, npf : Longint ;
Begin
  Repeat
    Write ('p = '); Readln (p);
    Write ('n = '); Readln (n);
  Until (0<p) and (p<n) ;
  Nf :=1;
  Pf :=1;
  Npf :=1;
  FOR i:=2 To n Do
    Begin
      Nf := nf*i ;
      IF i<=p Then pf := pf*i;
      IF i<=n-p Then npf := npf*i;
    End;
  Cnp := nf / (pf*npf) ;
  Writeln ('Combinaison = ', cnp :4:2) ;
End.

```

Exercise 35

```

Program Ordered ;
Uses Wincrt;
Var n : Integer;
    i, mn, mx, p : Longint;
(*****)
Procedure min_max (m : Integer; Var min, max:Longint);
Var i : Integer;
Begin
    min:=0; max:=0;
    FOR i:=1 To m Do
        Begin
            min:=10*min + i;
            max:=10*max + (9-m+i);
        End;
    End;
End;
(*****)
Function ordre (m:Longint) : Boolean;
Var c1, c2 : Integer; valide : Boolean;
Begin
    Repeat
        c1:= m MOD 10;
        m:= m DIV 10;
        c2:= m MOD 10;
        valide:= (c1>c2);
    Until Not (valide) Or (m<10) ;

```

```

        ordre:=valide;
    End;
    (*****)
    Procedure saisie (Var m : Integer);
    Begin
        Repeat
            Write ('Donner un entier n compris entre 2 et 8 : ');
            Readln (m);
            Until m in [2..8];
        End;
        (*****)
        Begin
            saisie (n);
            min_max (n, mn, mx);
            p:=0;
            FOR i:=mn To mx DO
                IF ordre (i) Then
                    Begin
                        p:=p+1;
                        Writeln (p:10, ' - ', i); Readln ;
                    End;
                End;
            End.

```

Exercise 36

```

Program Nbre_Impairs;
Uses Wincrt;
Var i, n : Integer;
Begin
    n:=0;
    FOR i:=1 To 99 Do
        IF ODD (i) and (i mod 7 <>0)
            Then Begin
                n:=n+1;
                IF n mod 5 <>0
                    Then Write (i:4)
                    Else Writeln (i:4);
            End;
    End.

```

Exercise 37

```

program sommes_entiers;
uses wincrt;
var n, i, j, k, s:integer;
begin
    writeln ('entrer la valeur de N :'); readln (n);
    for i:=1 to n div 2 do
        begin
            S := i; j:= i;
            Repeat
                J := j +1;
                S := S + j;
            until s >= n;
            if S=N then
                begin
                    write (n,' = ',i);
                    for k:=i+1 to j do write(' + ',k);
                    writeln;
                end;
            end;
        end;
    end.

```

Exercise 38

```

Program Reine ;
Uses Wincrt;
Var x, y, i, j : Byte ;
Begin
    Write ('Les coordonnées de la dame: X = ') ; Readln (X);
    Write ('                Y = ') ; Readln (Y);
    FOR i:=1 To 8 Do
        Begin
            FOR j:=1 To 8 Do
                IF (i=x) and (j=y)
                    Then Write (' R ')
                Else IF (i=x) Or (j=y) Or (abs(x-i)=abs(y-j))
                    Then Write (' * ')
                    Else Write (' ');
            Writeln ;
        End;
    End.

```

LES TABLEAUX

Exercise 1

```

Program Som_Produit_MoyArith ;
Uses Wincrt ;
Type Tab = Array [1..10] of Integer ;
Var T : Tab ; n, i, st : Integer ; mt, pt : Real ;
Begin
  Repeat
    Write ('N = ');
    Readln (n) ;
  Until (n>5) and (n<=10) ;
  FOR i := 1 To n Do
    Repeat
      Write ('T', i, ' = ');
      Readln (T[i]) ;
    Until (1<= T[i]) and (T[i] <= 20) ;
  st := 0 ;
  pt := 1 ;
  FOR i := 1 To n Do
    Begin
      St := st + T[i] ;
      Pt := pt * T[i] ;
    End ;
  Writeln ('Somme = ', st);
  Writeln ('Produit = ', trunc (pt));
  Writeln ('Moyenne arithmétique = ', st/n:2:2);
End.

```

Exercise 3

```

Program Freq_Lettre ;
Uses Wincrt ;
Const n=35 ;
Var LET : Array [1..n] of Char ;
    FE : Array ['A'..'Z'] of Byte ;
    i : Byte ; j : Char;
Begin
  Randomize ;
  FOR i := 1 To n Do
    Begin
      LET[i] := CHR (65+ Random (26)) ;
      Write (LET[i] :2) ;
    End;
  FOR j := 'A' To 'Z' Do FE[j] := 0 ;
  FOR i := 1 To n Do
    FE[LET[i]] := FE[LET[i]] + 1;

  Writeln ; Writeln;
  FOR j := 'A' To 'Z' Do Write (j:2);
  Writeln ;
  FOR j := 'A' To 'Z' Do Write (FE[j]:2) ;
End.

```

Exercise 2

```

Program Affich_sans_redondance;
Uses Wincrt;
Var T : Array [1..20] of Char;
    n, i, j : Integer;
Begin
  Repeat
    Writeln ('Donner un entier'); Readln (n);
  Until n in [3..20] ;
  FOR i:=1 To n Do
    Repeat
      Writeln ('Saisir la case d"ordre ', i);
      Readln (T[i]);
    Until upcase (T[i]) in ['A'..'Z'] ;
  FOR i:=1 To 20 Do Write (T[i], ' ');
  Writeln ;
  Writeln ;
  Write (T[1], ' ');
  FOR i:=2 To 20 Do
    Begin
      j:=i;
      While (j>2) and (T[j]<>T[j-1]) Do j:=j-1;
      IF T[i] <>T[j-1] Then Write (T[i], ' ');
    End;
  End.

```

Exercise 4

```

Program Conversion_base10_base2;
Uses Wincrt;
Var rest : Array [1..50] of 0..1;
    n, i, j : Integer;
Begin
  Repeat
    Writeln ('Donner un entier positif'); Readln (n);
  Until (n > 0);
  i:=0;
  Repeat
    i:=i+1;
    rest[i]:=n mod 2;
    n:=n div 2;
  Until n=0;
  FOR j:=i Downto 1 Do Write (rest[j]);
End.

```

===== Solution 2 =====

```

Program Conversion_base10_base2;
Uses Wincrt;
var n:integer;
(*****
procedure saisir(var n:integer);
begin
  Repeat
    Writeln ('Donner un entier positif'); Readln (n);
  Until (n > 0);
end;
(*****
function dec_bin (n:integer):string;
var chb,chr:string;
    r:0..1;
begin
  chb:="";
  Repeat
    r:=n mod 2;
    str(r, chr);
    insert (chr, chb, 1);
    n:=n div 2;
  Until n=0;
  dec_bin:=chb;
end;
(*****pp*****
Begin
  Saisir (n);
  writeln (('(',n,')10 = ('dec_bin(n),')2');
End.

```

Exercise 5

```

Program Conversion_b1_b2;
Uses Wincrt;
Var nb, reste : Array [1..50] of 0..15;
    b1, b2, n, i, j, err, nb10 : Integer;
    nch : String;
Begin
    Repeat
        Write ('Base b1 = '); Readln (b1);
        Write ('Base b2 = '); Readln (b2);
    Until (b1 in [2..16]) and (b2 in [2..16]);

    Writeln ('Donner le nombre à convertir'); Readln (nch);
    n:=Length (nch);
    FOR i:=1 To n Do
        IF ORD (nch[i]) <65
            Then VAL (nch[i], nb[i], err)
            Else nb[i] :=(ORD (nch[i]) - ORD ('A') + 10) ;

```

```

{conversion de la base b1 au décimal}
nb10 :=0;
FOR i:=1 To n-1 Do nb10:=(nb10+nb[i])*b1;
nb10:=nb10+nb[n];

{conversion de nb10 du décimal à la base b2}
i:=0;
Repeat
    i:=i+1;
    reste[i]:=nb10 mod b2;
    nb10:=nb10 div b2;
Until nb10=0;

{affichage du résultat}
FOR j:=i Downto 1 Do
    IF reste[j] < 10
        Then Write (reste[j])
        Else Write (CHR (reste[j]-10 + ORD ('A')));
End.

```

Exercise 6

```

Program Eclater_tab ;
Uses Wincrt ;
Type Tab = Array [1..50] of Integer ;
Var T, TN, TP : Tab ;
    n, i, j, k : Integer ;
Begin
    Repeat
        Writeln ('Saisir un entier') ;
        Readln (n) ;
    Until (n>=10) and (n<=50) ;
    Writeln ('Saisir les ', n, ' éléments de T') ;
    FOR i:=1 To n Do Readln (T[i]) ;

```

```

j := 0 ; k := 0 ;
FOR i := 1 To n Do
    IF T[i] < 0 Then Begin
        j := j+1 ;
        TN[j] := T[i] ;
    End
    Else Begin
        k := k+1 ;
        TP[k] := T[i] ;
    End ;
FOR i := 1 To j Do Write (TN[i]:4) ;
Writeln ;
FOR i := 1 To k Do Write (TP[i]:4) ;
End.

```

Exercise 7

```

Program Inverser_tab ;
Uses Wincrt ;
Type Tab = Array [1..50] of Integer ;
Var T : Tab ;
    i, n, aux : Integer ;
Begin
    Repeat
        Writeln ('Saisir un entier') ;
        Readln (n) ;
    Until n in [10..50] ;
    Writeln ('Saisir les ', n, ' éléments de T') ;
    FOR i := 1 To n Do Readln (T[i]) ;

```

```

FOR i := 1 To (n div 2) Do
    Begin
        aux := T[i] ;
        T[i] := T[n-i+1] ;
        T[n-i+1] := aux ;
    End ;
Writeln ; Writeln ;
Writeln ('Tableau inversé :') ;
FOR i := 1 To n Do Write(T[i]:4) ;
End.

```

Exercise 8

```

Program Regrouper_tab ;
Uses Wincrt ;
Var T : Array [1..50] of Integer ;
    i, j, k, n, tmp : Integer ;
Begin
    Repeat
        Write ('N = '); Readln (n) ;
    Until (n>=10) and (n<=50) ;
    Randomize ;
    FOR i := 1 To n Do
        Begin
            T[i] := -20+Random (41) ;
            Write (T[i]:4) ;
        End;

```

```

k:=0 ;
FOR i := 1 To n Do
    IF (T[i] mod 2) = 0
        Then Begin
            k := k+1 ;
            tmp := T[i] ;
            FOR j:=i Downto k+1 Do T[j]:=T[j-1] ;
            T[k] := tmp ;
        End ;
Writeln ; Writeln ;
FOR i := 1 To n Do Write (T[i]:4) ;
End.

```

Exercise 9

```

Program Min_Max_tab ;
Uses Wincrt;
Var T : Array [1..50] of Integer;
    i, min, max, n : Integer;
Begin
    Repeat Readln (n) Until (n>=10) and (n<=50);
    FOR i:=1 TO N DO Readln (T[i]);

```

```

min:=T[1];
max:=T[1];
FOR i:=2 TO n DO
    Begin
        IF T[i]<min Then min:=T[i];
        IF T[i]>max Then max:=T[i];
    End;
Writeln ('Valeur maximale = ', max);
Writeln ('Valeur minimale = ', min);
End.

```

Exercice 11

```

Program Ranger_tab ;
Uses Wincrt ;
Type Tab = Array [1..10] of Integer ;
Var T, R : Tab ;
    i, j : Integer ;
Begin
    Writeln ('Saisir les 10 éléments de T') ;
    FOR i:=1 To 10 Do
        Readln (T[i]) ;
    j:=0 ;
    FOR i:= 1 To 10 Do
        IF T[i] >= 0 Then Begin
            j := j+1 ;
            R[j] := T[i] ;
        End;
    FOR i:= 1 To 10 Do
        IF T[i] < 0 Then Begin
            j := j+1 ;
            R[j] := T[i] ;
        End;
    FOR i:= 1 To 10 Do Write (T[i]:4) ;
    Writeln ;
    FOR i:= 1 To 10 Do Write (R[i]:4) ;
End.

```

Exercice 10

```

Program Symetri_tab ;
Uses Wincrt ;
Type Tab = Array [1..50] of Integer ;
Var T : Tab ;
    i, j, n : Integer ;
Begin
    Repeat
        Writeln ('Saisir un entier') ;
        Readln (n) ;
    Until (n>1) and (n mod 2 =0) ;
    Writeln ('Saisir ', n div 2, ' éléments de T') ;
    FOR i := 1 To (n div 2) Do
        Begin
            Readln (T[2*i-1]) ;
            T[2*i] := T[2*i-1] ;
        End;
    FOR i := 1 To (n div 2)-1 Do
        Begin
            FOR j:=i+1 To n-i Do T[j] := T[j+1] ;
            T[n-i+1] := T[i] ;
        End ;
    Writeln ('Tableau symétrique :') ;
    FOR i := 1 To n Do Write (T[i]:4) ;
End.

```

Exercice 12

```

Program Recherche_Dichotomique_tab ;
Uses Wincrt;
type Tab = Array [1..50] of Integer;
Var T : Tab;
    N, V : Integer;
(*****)
Procedure Saisie (Var T : Tab ; Var n, v : Integer);
Var i : Integer;
Begin
    Repeat
        Writeln ('Donner un entier'); Readln (n);
    Until n in [10..50];

    Writeln ('Saisir les éléments de T en ordre croissant');
    Readln (T[1]);
    FOR i:=2 To n Do
        Repeat
            Readln (T[i])
        Until T[i] >= T[i-1];
    Writeln ('Donner la valeur à chercher'); Readln (v);
End;

```

```

(*****)
Function Recherche (v, n : Integer ; T : Tab) : Integer;
Var d, g, m, pos : Integer;
Begin
    g:=1 ; d:=n ; pos:=0;
    Repeat
        m:=(g+d) div 2 ;
        IF V=T[m] Then pos:=m
        Else IF V>T[m]
            Then g:=m+1
            Else d:=m-1;

        Until (pos=m) Or (g>d);
    recherche:= pos ;
End;
(*****)
Begin
    Saisie (T, N, V);
    IF Recherche (V, N, T) = 0
        Then Writeln (V, ' ne figure pas dans le tableau')
        Else Writeln (V, ' se trouve à la position ',
            recherche (V, N, T));
End.

```

Exercice 12

```

Program Recherche_sequentielle_tab ;
Uses Wincrt ;
Var T : Array [1..50] of Integer ;
    i, v, n : Integer ;
Begin
    Repeat
        Write ('N = '); Readln (n);
    Until (10<=n) and (n<=50);
    Writeln ('Saisir les ', n, ' éléments de T ');
    FOR i:= 1 To n Do Readln (T[i]) ;
    Writeln ('Donner la valeur à chercher') ; Readln (v) ;
    i := 0 ;
    Repeat
        i :=i+1 ;
    Until (v=T[i]) Or (i=n) ;
    IF v=T[i]
        Then Writeln (v, ' se trouve à la position ', i)
        Else Writeln (v, ' ne figure pas dans le tableau');
End.

```

Exercice 13

```

Program Regrouper_Tab ;
Uses Wincrt ;
Var T : Array [1..20] of Integer ;
    i, j, k, n, tmp : Integer ;
Begin
    Repeat
        Write ('N = ') ; Readln (n) ;
    Until (n>=2) and (n<=20) ;
    Writeln ('Saisir les éléments de T') ;
    FOR i:=1 To n Do Readln (T[i]) ;
    FOR i:=1 To n-1 Do
        FOR j:=i+1 To n Do
            IF (T[j] = T[i]) Then T[j]:=0 ;
        k:=0 ;
        FOR i:=1 To n Do
            IF T[i]<>0 Then Begin
                k:=k+1 ;
                IF T[i] <> T[k] Then Begin
                    tmp := T[k] ;
                    T[k] := T[i] ;
                    T[i] := tmp ;
                End ;
            End ;
        FOR i:=1 To n Do Write(T[i]:3) ;
    End.

```

Exercise 14

```

Program Frequence ;
Uses Wincrt ;
Const n=20 ;
Var T : Array [1..n] of 1..6 ;
    F : Array [1..6] of 0..20 ;
    i : 1..20 ;

Begin
  Randomize ;
  FOR i := 1 To n Do
    Begin
      T[i] := 1+ Random (6) ;
      Write (T[i] : 2) ;
    End ;

  FOR i:=1 To 6 Do F[i] := 0 ;

  FOR i:=1 To n Do
    F[T[i]] := F[T[i]] + 1 ;

  Writeln ;
  FOR i := 1 To 6 Do Write (F[i] : 4) ;
End.

```

Exercise 16

```

Program Insert_Tab;
Uses Wincrt ;
Const n_max = 100;
Var T : Array [1..n_max] of Char;
    c : Char;
    i, k, n : Integer;

Begin
  Repeat
    Writeln ('Donner un entier '); Readln (n);
  Until (n>=1) and (n<n_max);
  Writeln ('Saisir les éléments de T');
  FOR i:=1 To n Do Readln (T[i]);
  Writeln ('Donner le caractère à insérer');
  Readln (c);
  Repeat
    Writeln ('Donner la position d"insertion');
    Readln (k)
  Until k in [1..n];

  {décalage des éléments vers droite}
  FOR i:= n Downto k Do T[i+1] := T[i];
  T[k]:=c;
  FOR i:=1 To n+1 Do Write(T[i]:4);
End.

```

Exercise 17

```

Program Triangle_Pascal;
Uses Wincrt;
Type Tab = Array [1..15] of Integer;
Var T : Tab; N : Integer;
Procedure init (n : Integer ; Var T : Tab);
Var i : Integer;
Begin
  T[1]:=1;
  FOR i:=2 To n Do T[i]:=0;
End;
Procedure triangle (n : Integer ; Var T : Tab);
Var i, j : Integer;
Begin
  Writeln (T[1]);
  FOR i:=2 To n Do
    Begin
      FOR j:=i Downto 2 Do
        Begin
          T[j]:=T[j]+T[j-1];
          Write (T[j], ' ');
        End;
      Writeln (T[1]);
    End;
End;

```

Exercise 15

```

Program Moy_Rang;
Uses Wincrt;
Const n=30;
Var A, R : Array [1..n] of Real;
    j, i : Integer;

Begin
  FOR i:=1 To n Do
    Repeat
      Write ('Note élève ', i, ' : ');
      Readln (A[i]);
    Until (A[i]>=0) and (A[i]<=20);

  FOR i:=1 To n Do
    Begin
      R[i]:=1;
      FOR j:=1 To n Do
        IF A[i]<A[j] Then R[i]:=R[i]+1;
      End;

  Writeln ('Moyens':25, 'Rangs':8);
  FOR i:=1 To n Do
    Writeln (A[i]:25:2, trunc (R[i]):5);
End.

```

Exercise 18

```

Program Nbre_Rond;
Uses Wincrt;
Var reste, c : String;
    n, m, i, j, n1, n0 : Integer;

Begin
  FOR i:=1 To 1000 Do
    Begin
      n:=i;
      m:=i;
      reste:="";
      Repeat
        STR (n mod 2, c);
        reste:= c+reste;
        n:=n div 2;
      Until n=0;
      n1:=0; n0:=0;
      FOR j :=1 To Length (reste) Do
        Begin
          IF reste[j]='1' Then n1:=n1+1;
          IF reste[j]='0' Then n0:=n0+1;
        End;
      IF n1=n0 Then Writeln (m, ' est ROND')
        Else Writeln (m, ' n'est pas ROND');
      Readln ;
    End;
  End.

```

Exercise 17

```

Program Triangle_Pascal;
Uses Wincrt;
Type matrice = Array [1..15, 1..15] of Integer;
Var T : matrice; N : Integer;
Procedure triangle (n : Integer ; Var T:matrice);
Var l, c : Integer;
Begin
  T[1,1]:=1;
  FOR l:=2 To n Do
    Begin
      T[l,1]:=1;
      FOR c:=2 To l-1 Do
        T[l,c]:=T[l-1,c]+T[l-1,c-1];
      T[l,l]:=1;
    End;
  End;
Procedure Afficher (n : Integer ; T:matrice);
Var l, c : Integer;
Begin
  FOR l:=1 To n Do
    Begin
      FOR c:=1 To l Do Write (T[l,c], ' ');
      Writeln ;
    End;
  End;

```

<pre> (*****) Begin Repeat Writeln ('Donner la taille du triangle : '); Readln (n); Until n in [2..15]; Init (n, T); Triangle (n, T); End. </pre>	<pre> End; (*****) Begin Repeat Writeln ('Donner la taille du triangle : '); Readln (n); Until n in [2..15]; Triangle (n, T); Afficher (n, T); End. </pre>
Exercice 19 <pre> Program transpose_matrice; uses wincrt; const nmax=10; type Mat=array[1..nmax,1..nmax] of integer; var M:Mat; n:integer; Procedure saisie (Var n:integer); begin repeat writeln('Donner n :'); readln(n); until n in [1..nmax]; end; Procedure remplir (Var M:Mat; n:integer); var i, j:integer; begin For i:=1 to n do For j:=1 to n do begin Writeln('Donner M['',i'',',j'',']'); readln(M[i,j]); end; end; end; </pre>	<pre> Procedure Transpose (Var M:Mat;n:integer); var a ux,i,j:integer; begin For i:=1 to n do For j:=1 to i-1 do begin aux:=M[i,j]; M[i,j]:=M[j,i]; M[j,i]:=aux; end; end; end; Procedure Affiche (M:Mat; n:integer); var i, j:integer; begin For i:=1 to n do begin For j:=1 to n do Write(M[i,j], ' '); writeln; end; end; {Programme Principal} begin saisie (n); remplir (M, n); Transpose (M, n); Affiche (M, n); end. </pre>
Exercice 20 <pre> Program TRI_SELECTION ; Uses Wincrt ; Const n = 20 ; Type Tab = Array [1.. n] of String ; Var T : Tab ; i , j , posmin : Integer ; tmp : String; Begin Writeln ('Remplir le tableau par des chaînes :'); FOR i := 1 TO n DO Readln (T[i]) ; FOR i := 1 TO n-1 DO Begin posmin := i ; FOR j := i+1 TO n DO IF T[j] < T[posmin] Then posmin := j ; IF i<> posmin Then Begin tmp := T[i] ; T[i] := T[posmin] ; T[posmin] := tmp ; End ; End ; Writeln ('Tableau trié :'); FOR i := 1 TO n DO Writeln (T[i]) ; End. </pre>	Exercice 21 <pre> Program Moy_Rang; Uses Wincrt; Const n=30; Var Nom : Array [1..n, 1..2] of String [50]; Note : Array [1..n, 1..4] of Real; j, i, a, b : Integer; Begin FOR i:=1 To n Do Begin Writeln ('Elèves n° ', i); Write ('Nom : '); Readln (nom[i,1]); Write ('Prénom : '); Readln (nom[i,2]); FOR j:=1 To 2 Do Repeat Write ('Note ', j, ' : '); Readln (note[i,j]); Until (note[i,j]>=0) and (note[i,j]<=20); note[i,3] := (note[i,1]+note[i,2]*2)/3; End; FOR i:=1 To n Do Begin note[i,4]:=1; FOR j:=1 To n Do IF note[i,3]<note[j,3] Then note[i,4]:=note[i,4]+1; End; Writeln ('Nom', 'Prénom':15, 'Note 1':10, 'Note 2':10, 'Moyens':10, 'Rangs':10); FOR i:=1 To n Do Begin a:=Length (nom[i,1]); b:=Length (nom[i,2]); Writeln (nom[i,1], nom[i,2]:(12-a+b), note[i,1]:(15-b):2, note[i,2]:10:2, note[i,3]:10:2, trunc(note[i,4]):9); End; End; End. </pre>

Exercise 22

```

Program Intersection_Tab;
Uses Wincrt;
Type vect = Array [1..99] of Integer;
Var T1, T2, inter : vect;
    n, m : Integer;
(*****)
Procedure saisie_int (Var nf : Integer);
Begin
    Repeat
        Write ('N = ');
        Readln (nf);
    Until nf in [3..99];
End;
(*****)
Procedure remplir_tab (nf : Integer ; Var A:vect);
Var i, j : Integer;
Begin
    Writeln ('Saisir la case 1');
    Readln (a[1]);
    FOR i:=2 To nf Do
        Repeat
            Writeln ('Saisir la case ', i);
            Readln (A[i]);
            j:=i-1;
            While (j>1) and (A[i] <> A[j]) Do j:=j-1;
            Until A[i]<>A[j];
        End;
    End;

```

```

(*****)
Procedure intersection (nf : Integer ; A1, A2:vect;
    Var p : Integer ; Var B:vect);
Var i, j : Integer;
Begin
    p:=0;
    FOR i:=1 To nf Do
        Begin
            j:=0;
            Repeat
                j:=j+1;
            Until (j=nf) Or (A1[i]=A2[j]);
            IF A1[i]=A2[j]
                Then Begin
                    p:=p+1;
                    B[p]:=A1[i];
                End;
            End;
        End;
    End;
(*****)
Procedure affiche_tab (nf : Integer ; A:vect);
Var i : Integer;
Begin
    FOR i:=1 To nf Do Write (A[i], ' ');
End;
(***** P.P. *****)
Begin
    saisie_int (n); remplir_tab (n, T1); remplir_tab (n, T2);
    intersection (n, T1, T2, m, inter);
    affiche_tab (n, T1); Writeln ; affiche_tab (n, T2);
    Writeln ;
    affiche_tab (m, inter);
End.

```

Exercise 23

```

program tri_2_criteres;
uses wincrt,ecran;
const n=10;
type tab=array[1..n] of string;
var t:tab;
    i,j,pos:integer;
    aux:string;
begin
    writeln('Remplir T :');
    for i:=1 to n do
        repeat
            write('ch = ');
            readln(t[i]);
            until t[i]<>'';
        (*****)
    for i:=1 to n-1 do
        begin
            pos:=i;
            for j:=i+1 to n do
                if (length(t[j])<length(t[pos])) OR
                    ((length(t[j])=length(t[pos]))AND(t[j]<t[pos]))
                    then pos:=j ;
            if i<>pos then begin
                aux:=t[i] ;
                t[i]:=t[pos];
                t[pos]:=aux;
            end;
        end;
    (*****)
    for i:=1 to n do writeln (t[i]);
end.

```

Exercise 25

```

Program tri_2_criteres;
uses wincrt,ecran;
type tch=array[1..10] of string[20]; tc=array[1..10] of char;
var n:integer; t:tch; c:tc;
(*****)
procedure saisie(var n:integer;var t:tch;var c:tc);
var i:integer;
begin
    write ('N = ');readln(n);
    writeln ('remplir les tableaux T et C :');
    for i:=1 to n do
        begin
            write('nom = '); readln(t[i]);
            repeat write ('couleur = '); readln (c[i]); until c[i] in ['B','N'];
        end;
    end;
(*****)
procedure tri(n:integer;var t:tch;var c:tc);
var i:integer; permut:boolean; aux:string; tmp:char;
begin
    repeat
        permut:=false;
        for i:=1 to n-1 do
            if (c[i]>c[i+1])or((c[i]=c[i+1])and(t[i]>t[i+1]))
                then begin
                    aux:=t[i] ; t[i]:=t[i+1]; t[i+1]:=aux;
                    tmp:=c[i] ; c[i]:=c[i+1]; c[i+1]:=tmp;
                    permut:=true
                end;
        n:=n-1
    until (permut=false) or (n=1);
end;
(*****)
procedure affiche (n:integer;t:tch;c:tc);
var i:integer;
begin
    for i:=1 to n do writeln(t[i], ' ',c[i]);
end;
(*****P.P*****)
Begin
    Saisie (n,t,c); tri (n,t,c); affiche (n,t,c);
End.

```

Exercise 24

```

Program tri_bulles_bidirectionnel;
uses wincrt,ecran;
type tab=array[1..25] of integer;
var t:tab;
    n:integer;
(*****)
procedure saisir(var n:integer);
begin
    repeat
        writeln('Donner un entier entre 5 et 25');
        readln(n);
    until n in [5..25];
end;
(*****)
procedure remplir (var t:tab ; n:integer);
var i:integer;
begin
    randomize;
    for i:=1 to n do T[i]:=1+random(100);
end;
(*****)
procedure trier (var T:tab ; n:integer);
var i,aux,debut,fin:integer;
    permut:boolean;
(*****)
begin
    debut:=1;fin:=n;
    repeat
        permut:=false;

        for i:=debut to fin-1 do
            if t[i]>t[i+1]
            then begin
                aux:=T[i];
                T[i]:=T[i+1];
                T[i+1]:=aux;
                permut:=true;
            end;
        fin:=fin-1;

        for i:=fin downto debut+1 do
            if t[i]<t[i-1]
            then begin
                aux:=T[i];
                T[i]:=T[i-1];
                T[i-1]:=aux;
                permut:=true;
            end;

        debut:=debut+1;
    until (permut=false) or (debut>=fin);
end;
(*****)
procedure afficher(T:tab ; n:integer);
var i:integer;
begin
    for i:=1 to n do write(T[i], ' ');
end;
(***** P.P. *****)
BEGIN
    saisir(n);
    remplir(t,n);
    writeln('Tableau avant le tri :');
    afficher(t,n);
    trier(t,n);
    writeln;
    writeln('Tableau après le tri :');
    afficher(t,n);
END.

```

Exercise 26

```

program fusion;
uses wincrt,ecran;
type tab=array [1..20] of integer;
var v1,v2,v3:tab;
    n,m,c:integer;
(*****)
procedure lecture (var taille:integer);
begin
    repeat
        readln(taille);
    until taille in [2..20];
end;
(*****)
procedure remplir(var t:tab; taille:integer);
var i:integer;
begin
    for i:= 1 to taille do readln(t[i]);
end;
(*****)
procedure trier (taille:integer;var t:tab);
var i,tmp, min,j:integer;
begin
    for i:=1 to taille-1 do
        begin
            min:=i;
            for j:=i+1 to taille do
                if t[j]<t[min] then min:=j;
            if i<>min then begin
                tmp:=t[i];
                t[i]:=t[min];
                t[min]:=tmp;
            end;
        end;
end;
(*****)
procedure fusionner(v1,v2:tab;var v3:tab;n,m:integer;var c:integer);
var i,c1,c2:integer;
begin
    c1:=1; c2:=1; c:=0;
    repeat
        c:=c+1;
        if v1[c1]<v2[c2]
        then begin
            v3[c]:=v1[c1];
            c1:=c1+1;
        end
        else begin
            v3[c]:=v2[c2];
            c2:=c2+1;
        end
    until (c1>n) or (c2>m);
    if c1>n then
        for i:=c2 to m do
            begin
                c:=c+1;
                v3[c]:=v2[i];
            end
        else
            for i:=c1 to n do
                begin
                    c:=c+1;
                    v3[c]:=v1[i];
                end;
    end;
(*****)
procedure afficher(t:tab; taille:integer);
var i:integer;
begin
    writeln('Tableau fusion :');
    for i:= 1 to taille do
        write(t[i]:4);
    end;
(*****P.P*****)
begin
    write('Taille V1 : ');lecture(n);
    write('Taille V2 : ');lecture(m);
    writeln('Remplir V1 :');remplir(v1,n);

```

```
writeln('Remplir V2 :');remplir(v2,m);
trier(n,v1);
trier(m,v2);
fusionner(v1,v2,v3,n,m,c);
afficher(v3,c);
end.
```

Exercice 27

```
Program temps_tris;
uses wincrt,windos,ecran;
type tab=array[1..1000] of real;
var t,t1,t2:tab;
    n:integer;
    hi1,hi2,mi1,mi2,si1,si2,csi1,csi2,hs1,hs2,
    ms1,ms2,ss1,ss2,css1,css2,ts1,ti1:word;
(***** lecture et duplication *****)
procedure lecture_duplic(var n:integer;var t,t1,t2:tab);
var i:integer;
begin
    Writeln('Saisir un entier pour la taille des tableaux');
    Readln(n);
    randomize;
    for i:=1 to n do
        begin
            t[i]:=100*random;    { réel aléatoire entre ]0..100[ }
            t1[i]:=t[i];
            t2[i]:=t[i];
        end;
    end;
(***** TRI SELECTION *****)
Procedure tri1 (n:integer;var t1:tab);
var pm,i:integer;
(*****)
Function posmin(d,f:integer;t:tab):integer;
var i,pmin,j:integer;
begin
    pmin:=d;
    for j:=d+1 to f do
        if t[j] < t[pmin] then pmin:=j;
    posmin:=pmin;
end;
(*****)
Procedure permut (var x,y:real);
var aux:real;
begin
    aux:=x;
    x:=y;
    y:=aux;
end;
(*****)
begin
    for i:=1 to n-1 do
        begin
            pm:=posmin(i,n,t1);
            if pm<>i then permut(t1[pm],t1[i]);
        end;
    end;
end;
```

```
(***** TRI INSERTION *****)
procedure tri2 (n:integer;var t2:tab);
var j,i:integer;
    tmp:real;
(*****)
procedure decaler (var t2:tab;var j:integer;i:integer);
begin
    j:=i;
    WHILE (j>1)and(t2[j-1]>tmp) DO
        Begin
            t2[j]:=t2[j-1];
            j:=j-1;
        End ;
    end;
(*****)
Begin
    for i:=2 to n do
        if t2[i]<t2[i-1]
            then Begin
                tmp:=t2[i];
                Decaler(t2,j,i);
                t2[j]:=tmp;
            End ;
    End;
(***** Affichage *****)
procedure affiche(n:integer;t:tab);
var i:integer;
begin
    for i:=1 to n do write(t[i]:2:2,' ');
end;
(***** Programme principal *****)
BEGIN
    lecture_duplic(n,t,t1,t2);
    gettime(hs1,ms1,ss1,css1);
    tri1(n,t1);
    gettime(hs2,ms2,ss2,css2);
    ts1:=(hs2-hs1)*3600*100+(ms2-ms1)*60*100+(ss2-s1)*100+css2-
css1;
    gettime(hi1,mi1,si1,csi1);
    tri2(n,t2);
    gettime(hi2,mi2,si2,csi2);
    ti1:=(hi2-hi1)*3600*100+(mi2-mi1)*60*100+(si2-si1)*100+csi2-
csi1;
    affiche(n,t1); readln;
    affiche(n,t2); readln;
    writeln('tri selection   : ',ts1, ' Centième de seconde');
    writeln('tri insertion  : ',ti1, ' Centième de seconde');
END.
```

Exercice 28

```
Procedure Trier (n:integer ; T:tab;var rang,s:tab);
Var    i,j : integer;
BEGIN
    FOR i:=1 TO n DO s[i]:=1;

    FOR i:=1 TO n-1 DO
        FOR j:=i+1 TO n DO
            IF T[i]>T[j]
                THEN s[i]:=s[i]+1
                ELSE s[j]:=s[j]+1;

    FOR i:=1 TO n DO rang[s[i]]:=i;
END;
```

Exercice 29

```
program long_suite;
uses wincrt, ecran;
const n=20;
type tab=array[1..n] of char;
var t:tab; max,suite:string; i:integer;
begin
    for i:=1 to n do readln(t[i]);
    max:=t[1];
    suite:=t[1];
    for i:=2 to n do
        if t[i]=t[i-1]
            then suite:=suite+t[i]
            else begin
                if length(suite)>length(max) then max:=suite;
                suite:=t[i];
            end;

    writeln (max[1], length(max));
end.
```

LES CHAINES DE CARACTERES

Exercise 2

```

Program Palindrome;
Uses Wincrt ;
Var ch, inv : String;
    i : Integer;

Begin
  Writeln ('Saisir une chaîne'); Readln (ch);
  inv := '';
  FOR i := Length (ch) Downto 1 Do
    inv := inv + ch[i];
  IF ch = inv
  Then Writeln (ch, ' est palindrome')
  Else Writeln (ch, ' n'est pas palindrome');
End.

```

Exercise 4

```

Program Chaine_Majus_Minus;
Uses Wincrt;
Var ch : String;
    i : Integer;

Begin
  Writeln ('Saisir une chaîne de caractères'); Readln (ch);
  FOR i:=1 To Length (ch) Do
    IF ch[i] in ['a'..'z']
    Then ch[i]:=ch[i]
    Else ch[i]:=CHR (ORD (ch[i]) + 32);
  Writeln (ch);
  Writeln ;
  FOR i:=1 To Length (ch) Do ch[i]:=upcase (ch[i]);
  Writeln (ch);
End.

```

Exercise 5

```

Program Chaine1 ;
Uses Wincrt ;
Var i : Integer ; ch: String ;
    test : Boolean;

Begin
  Repeat
    Writeln ('Donner un mot en majuscule') ; Readln (ch) ;
    test:=True;
    i:=0 ;
    Repeat
      i:=i+1;
      IF Not(ch[i] in ['A'..'Z']) Then test:=False;
    Until (test=False) Or (i=Length (ch));
  Until test=True ;
  FOR i := 1 To Length (ch) Do
    Writeln (COPY (ch, 1, i));
End.

```

Exercise 7

```

Program Renverser_ch ;
Uses Wincrt ;
Var p : Integer ;
    chr, chd : String ;

Begin
  Writeln ('Saisir une phrase') ; Readln (chd) ;
  chr := '';
  p := POS(' ', chd) ;
  While p <> 0 Do
    Begin
      chr := ' ' + COPY (chd, 1, p-1) + chr ;
      DELETE (chd, 1, p) ;
      p := POS (' ', chd) ;
    End ;
  chr := chd + chr ;
  Writeln ('Phrase renversée est : ', chr) ;
End.

```

Exercise 3

```

Program chaine_inverse;
Uses wincrt;
Var ch : string;
  (*****)
Function miroir (ch:string):string;
var i, l : integer; c : char;
begin
  l:=length(ch);
  for i:=1 to l div 2 do
    begin
      c:=ch[i];
      ch[i]:=ch[l-i+1];
      ch[l-i+1]:=c;
    end;
  miroir:=ch;
end;
  (***** p.p *****)
BEGIN
  write('ch = ');readln(ch);
  writeln('l'inverse de ',ch,' est : ', miroir(ch));
END.
===== solution 2 =====
Function miroir (ch:string) : string;
var i, l : integer; mirch : string;
begin
  l:=length(ch);
  mirch:=ch;
  for i:=1 to l do mirch[i] := ch[l-i+1];
  miroir:=mirch;
end;

```

Exercise 5

```

Program Chaine2;
Uses Wincrt;
Var ch : String; i, lg : Integer;
    test : Boolean;

Begin
  Repeat
    Writeln ('Donner un mot en majuscule') ; Readln (ch) ;
    test:=True; i:=0 ;
    Repeat
      i:=i+1;
      IF Not (ch[i] in ['A'..'Z']) Then test:=False;
    Until (test=False) Or (i=Length (ch));
  Until test=True ;
  lg:=Length (ch);
  FOR i:=1 To lg Do
    Writeln (COPY (ch, 1, i), COPY (ch, lg-i+1, i));
End.

```

Exercise 6

```

Program Espace_superflus ;
Uses Wincrt ;
Var ch : String;
    i, p : Integer ;

Begin
  Writeln ('Donner une chaîne'); Readln (ch);
  Repeat
    P := POS (' ', ch); {position de 2 espaces dans ch}
    IF p<>0 Then DELETE (ch, p, 1);
  Until p=0 ;
  IF ch[1]=' '
  Then DELETE (ch, 1, 1);
  IF ch[Length(ch)]=' '
  Then DELETE (ch, Length (ch), 1);

  Writeln ('La chaîne devient : ', ch);
End.

```

Exercise 8

```

Program Occurrence_car;
Uses Wincrt;
Var ch : String;
    i, j, n : Integer;
Begin
  Writeln ('Saisir une chaîne'); Readln (ch);
  FOR i:=1 To Length (ch) Do
    Begin
      n:=0;
      FOR j:=1 To Length (ch) Do
        IF ch[i]=ch[j]
          Then n:=n+1;
      IF i = POS (ch[i], ch)
        Then Writeln ('Occurrence de ', ch[i], ' = ', n);
    End;
  End.

```

Exercise 10

```

Program Sans_Redondance ;
Uses Wincrt;
Var ch1, ch2, ch3, aux : String; i : integer;
Begin
  Write('Chaîne 1 = ');Readln (ch1);
  Write('Chaîne 2 = ');Readln (ch2);
  if length(ch1)>length(ch2)
    then begin
      aux:=ch1;
      ch1:=ch2;
      ch2:=aux;
    end;
  ch3:="";
  FOR i:=1To Length(ch1) Do
    IF (POS(ch1[i],ch2)<>0)and(POS(ch1[i],ch3)=0)
      Then ch3:=ch3+ch1[i];
  Writeln(ch3);
End.

```

Exercise 12

```

Program Anagrammes;
uses wincrt;
var mot1, mot2 : string;
(*****)
procedure saisie_ch (var m1, m2 : string);
begin
  repeat
    writeln ('donner deux mots : ');
    readln (m1);
    readln (m2);
  until (m1 > "") and (m2 > "");
end;
(*****)
function trie (mot : string) : string;
var i, j, n : integer;
procedure permut (var a, b : char);
var aux : char;
begin
  aux:=a; a:=b; b:=aux;
end;
begin
  n:=length (mot);
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if mot[i]>mot[j]
        then permut (mot[i], mot[j]);
  trie := mot;
end;
(*****)
begin
  saisie_ch (mot1, mot2);
  if trie (mot1) = trie (mot2)
    then writeln (mot2, ' est une anagramme de ', mot1)
    else writeln (mot2, ' n''est pas une anagramme de ', mot1);
end.

```

Exercise 9

```

Program Occurrence_mot ;
uses wincrt;
var ch, mot : string; nb, i, k: integer;
begin
  repeat
    writeln ('saisir un texte'); readln (ch);
  until length (ch)>20;
  writeln ('saisir un mot'); readln (mot);
  k:=length (mot); nb:=0; i:=1;
  repeat
    if (ch[i] = mot[1]) and (mot = copy (ch, i, k))
      then begin
        nb:=nb+1;
        i:=i+k;
      end
    else i:=i+1;
  until i>length(ch);
  writeln (mot, ' figure dans le texte ', nb, ' fois');
end.

```

Exercise 11

```

Program Aerer_ch ;
Uses Wincrt ;
Var k : Byte ;
    ch : String ;
Begin
  Writeln ('Saisir une chaîne') ;
  Readln (ch) ;

  k := 0 ;
  repeat
    k:=k+2;
    Insert (' ', ch, k) ;
  Until k = length(ch)-1;

  Writeln ('Chaîne aérée = ', ch) ;
End.

```

Exercise 12

```

Program anagrammes;
uses wincrt;
var mot1, mot2:string;
(*****)
procedure saisie_ch(var m1,m2:string);
begin
  repeat
    writeln ('donner deux mots : ');
    readln (m1);
    readln(m2);
  until (m1 > "") and (m2 > "");
end;
(*****)
function anagram (mot1,mot2:string):boolean;
var p:integer;
begin
  anagram:=false;
  repeat
    p:=pos(mot1[1],mot2);
    if p>0
      then begin
        delete(mot1,1,1);
        delete(mot2,p,1);
      end;
    until (p=0) or (mot1="");
    if (mot1="") and (mot2="") then anagram:=true;
  end;
(*****)
begin
  saisie_ch (mot1, mot2);
  if anagram (mot1, mot2)
    then writeln (mot1, ' est une anagramme de ', mot2)
    else writeln (mot1, ' n''est pas une anagramme de ', mot2);
end.

```

Exercise 13

```

Program Pos_ch;
Uses Wincrt;
Var ch1, ch2 : String;
    i, p : Integer;
Begin
    Write ('ch1 = '); Readln (ch1);
    Write ('ch2 = '); Readln (ch2);
    i:=1 ;
    p:=0;
    Repeat
        IF ch1 = COPY (ch2, i, Length (ch1)) Then p:=i;
        i:=i+1;
    Until (Length (ch2)-i < Length (ch1)) Or (p<>0) ;
    Writeln ('La chaîne ', ch1, ' occupe la position ', p,
        ' dans la chaîne ', ch2);
End.

```

Exercise 15

```

Program Jeu_pendu ;
Uses Wincrt;
type chain=String [50];
Var se, ma:chain;
    rep, let : Char;
(*****)
Function controle_saisie (se:chain) : Boolean;
Var i : Integer;
    r : Boolean;
Begin
    r:=True; i:=0 ;
    Repeat
        i:=i+1;
        IF Not (se[i] in ['A'..'Z']) Then r:=False;
    Until (r=False) Or (i=Length (se));
    controle_saisie := r;
End;
(*****)
Procedure masquer (se:chain; Var ma : chain);
Var i : Integer;
Begin
    ma:=se;
    FOR i:= 2 To (Length (se)-1) Do ma[i]:='-';
End;
(*****)
Procedure saisie_let (Var let : Char);
Begin
    Writeln ('Donner une lettre ');
    Readln (let);
    let:=upcase (let);
End;
(*****)
Procedure dévoiler (se : chain; let : Char; Var ma : chain);
Var i : Integer; r : Boolean;
Begin
    r:=False;
    FOR i:=2 To (Length (se)-1) Do
        IF se[i]=let
            Then Begin
                ma[i]:=let;
                r:=True;
            End;
    IF r=False Then Writeln ('Echec');
End;
(*****)
Procedure partie_jeu (se, ma : chain ; let : Char);
Var nb : Integer;
Begin
    ClrScr;
    Writeln (ma);
    nb:=0;
    Repeat
        nb:=nb+1;
        saisie_let (let);
        dévoiler (se, let, ma);
        Writeln (ma);
    Until (nb=Length (se)) Or (ma=se);
    IF ma=se
        Then Writeln ('Bravo, trouvé en ', nb, ' propositions')

```

Exercise 14

```

Program Copie_ch;
Uses Wincrt;
Var ch1, chr : String;
    i, p, n, l : Integer;
Begin
    Write ('ch1 = '); Readln (ch1);
    l:=Length (ch1);
    Repeat
        Write ('Position = '); Readln (p);
        Write ('Nbre caractères = '); Readln (n);
    Until (p in [1..l]) and (n in [1..l]) and (p+n<=l+1);
    chr:="";
    FOR i:=p To p+n-1 Do
        chr:=chr+ch1 [i];
    Writeln ('La chaîne copiée est : ', chr);
End.

```

Exercise 16

```

Program Chaines_inverses;
Uses Wincrt;
Type Tab=Array [1..100] of String [50];
Var p, q : Tab;
    i, n : Integer;
(*****)
Procedure saisie_entier (Var m : Integer);
Begin
    Repeat
        Writeln ('Donner un entier'); Readln (m);
    Until (1<m) and (m<100);
End;
(*****)
Procedure saisie_tab (m : Integer; Var T : Tab);
Var i, j : Integer; test : Boolean;
Begin
    FOR i:=1 To m Do
        Repeat
            Writeln ('Donner l'élément d'ordre ', i); Readln (T[i]);
            j:=0; test:=True;
            While (test=True) and (j<Length (T[i])) Do
                Begin
                    j:=j+1;
                    IF Not (T [i, j] in ['0'..'9']) Then test:=False;
                End;
            Until (test=True) and (T[i] <>'');
        End;
    End;
(*****)
Function inverse (ch : String) : String;
Var i : Integer; chinv : String;
Begin
    chinv:="";
    FOR i:=Length (ch) Downto 1 Do
        chinv:=chinv+ch[i];
    inverse:=chinv;
End;
(*****)
Procedure Affiche_tab (m : Integer; T : Tab);
Var i : Integer;
Begin
    FOR i:=1 To m Do Write (T[i], ' ');
End;
(***** P.P *****)
Begin
    saisie_entier (n);
    saisie_tab (n, p);
    FOR i:=1 To n Do
        q[i] := inverse (p[i]);
    affiche_tab (n, q);
End.

```

<pre> Else Writeln ('Perdu le mot à deviner est ', se); End; (***** P. P *****) Begin Writeln ('Le jeu du pendu'); Writeln ; Repeat Repeat Write ('Rentrez un mot secret en majuscule : '); Readln (se); Until controle_saisie (se); masquer (se, ma); partie_jeu (se, ma, let); Write ('Voulez-vous rejouer (o/n) ? '); Readln (rep); Until Upcase (rep) = 'N'; End. </pre>	
Exercice 17 <pre> Program conversion_base1_base2; uses wincrt, ecran; var b1, b2:integer; nch:string; (***** Procedure saisie_base (var b1,b2:integer); begin repeat write('Base b1 = ');readln(b1); write('Base b2 = ');readln(b2); until (b1 in [2..16]) and (b2 in [2..16]) and (b1<>b2); end; (***** Procedure saisie_nombre (var nch:string; b1:integer); Const chb='0123456789ABCDEF'; Var i:integer; test:boolean; begin repeat writeln ('Donner un nombre en base ', b1); readln(nch); test:=true; for i:=1 to length(nch) do if (pos(nch[i],chb)>b1) or (pos(nch[i],chb)=0) then test:=false; until test=true; end; end; </pre>	<pre> (**** Conversion de la base b1 vers base 10 *****) Function Conv_b1_10 (nch:string; b1:integer) : longint; var err,i,n:integer; dec,puiss:longint; begin dec:=0;puiss:=1; for i:=length(nch) downto 1 do begin if nch[i] in ['0'..'9'] then Val(nch[i], n, err) else n:=ord(nch[i])-55; dec:=dec+n*puiss; puiss:=puiss*b1; end; conv_b1_10:=dec; end; (**** Conversion de la base 10 vers la base b2 *****) Function Conv_10_b2 (nd:longint; b2:integer) : string; var ch1, chb2:string; r:integer; begin chb2:=''; repeat r:= nd mod b2; if r in [0..9] then str(r,ch1) else ch1:=chr(55+r); chb2:=ch1+chb2; (*insert (ch1,chb2,1)*) nd:= nd div b2 ; until (nd = 0); conv_10_b2:=chb2; end; (***** P.P *****) Begin saisie_base(b1,b2); saisie_nombre(nch,b1); writeln('(',nch,')',b1,' = (', conv_10_b2 (conv_b1_10 (nch, b1), b2), ')', b2); End. </pre>
Exercice 19 <pre> Program sablier; uses wincrt; var esp,ch:string; (***** Procedure affiche_bas(var esp:string;ch:string); var ch2:string; n,i:integer; begin esp:='';n:=0;i:=1; writeln(ch); repeat esp:=' '+esp; n:=n+2;i:=i+1; ch2:=esp+copy(ch, i, length(ch)-n); writeln(ch2); until length(copy(ch, i, length(ch)-n))=1; end; (***** Procedure affiche_haut(esp,ch:string); var i,n:integer; ch2:string; </pre>	Exercice 18 <pre> Program suite_mystere; uses wincrt, ecran; var ligne, lignsuiv, c : string; n, l, nb, j, i : integer; begin write ('N = '); readln (n); ligne:='1'; for i:=1 to n do begin writeln (ligne); l:=length (ligne); nb:=1; lignsuiv:=''; for j:=1 to l do if ligne[j] = ligne[j+1] then nb:=nb+1 else begin str (nb, c); lignsuiv:= lignsuiv + c + ligne[j]; nb:=1; end; </pre>

```

begin
  i:=-1;n:=1;
  repeat
    n:=n+2;i:=i+1;
    delete(esp,1,1);
    ch2:=esp+copy(ch, length(ch) div 2-i, n);
    writeln(ch2);
  until ch2=ch;
end;
(*****pp*****)
begin
  repeat
    write('CH = ');
    readln(ch);
  until (ch<>'') and (odd(length(ch)));
  affiche_bas(esp,ch);
  affiche_haut(esp,ch);
end.

```

```

      ligne:= lignsuiv;
    end;
  end.

```

LA RECURSIVITE

Exercice 1

```

Procedure afficher ;
Var n:integer;
begin
  for n:=10 downto 1 do writeln(n);
end;
===== Solution récursive =====
Procedure afficher (n:integer);
begin
  if n >= 1
  then begin
    writeln (n);
    afficher (n-1);
  end;
end;
L'appel : afficher (10) ;

```

Exercice 2

```

Procedure affiche (a, b: integer);
begin
  if (a<4)
  then if (b<10)
        then begin
              writeln (a * 10 + b);
              affiche (a, b + 1);
            end
        else affiche (a + 1, 0);
  end;
L'appel : affiche (0,0) ;

```

Exercice 3

```

Procedure saisir (var n : byte);
begin
  writeln ('Taper un entier');
  readln (n);
  if (n<1) or (n>31) or (n mod 2 <> 0)
  then saisir (n) ;
end;

```

Exercice 4

```

Procedure remplir (n : integer ; var T:Tab);
begin
  if n>=1 then begin
    readln (T[n]);
    if T[n]<0
    then remplir (n, T)
    else remplir (n-1, T) ;
  end ;
end;
===== solution 2 =====
Procedure remplir (var T:tab ; i, n:integer);
begin
  if i<= n
  then begin
    readln (T[i]);
    if T[i]<0
    then remplir (T, i, n)
    else remplir (T, i+1, n)
  end;
L'appel : remplir (t, 1, n);

```

Exercice 4.2

```

Procedure affiche (T : tab ; n : integer);
begin
  if n>=1 then begin
    write (T[n]:4);
    affiche (T, n-1) ;
  end ;
end;
===== solution 2 =====
procedure affiche (T:tab ; i, n:integer);
begin
  if i<= n
  then begin
    write (T[i]:4);
    affiche (T,i+1,n);
  end;
L'appel : affiche (t, 1, n);

```

Exercice 5

```

Function fact (n : byte) : real ;
begin
  if n=0 then fact := 1
  else fact := n * fact(n - 1) ;
end;

```

Exercice 6

```

Function palindrome (ch: string): boolean;
begin
  if (length(ch)<2) then palindrome := true
  else if (ch[1] = ch[length(ch)]) then palindrome := palindrome (copy(ch, 2, length(ch) - 2))
  else palindrome := false;
end;
===== { solution 2 } =====
Function palindrome (ch: string): boolean;
begin
  if length (ch) < 2
  then palindrome := true
  else palindrome := (ch[1] = ch[length(ch)]) and (palindrome (copy(ch, 2, length(ch) - 2)));
end;

```

Exercise 7 <pre> Function pair (n:integer) : boolean; begin if (n=0) then pair := true else if (n=1) then pair := false else pair := pair (n-2) ; end ; </pre>	===== Solution 2 : récursivité mutuelle ===== <pre> Function pair (n:integer) : boolean; Function impair (n:integer) : boolean; begin if (n=0) then impair := false else impair := pair (n-1) ; end ; begin if (n=0) then pair := true else pair := impair (n-1) ; end ; </pre>
Exercise 8 <pre> Function prosom (p, q : integer) : integer; begin if (p=0) or (q=0) then prosom := 0 else if q=1 then prosom := p else prosom := p + prosom(p, q-1) ; end ; </pre>	Exercise 9 <pre> Function somme (n : integer) : longint; begin if (n=0) then somme := 0 else somme := n + somme (n-1) ; end ; </pre>
Exercise 10 <pre> Function puissance (x :real ; n : integer) : real; begin if (n=0) then puissance := 1 else if (n<0) then puissance := 1/puissance(x,-n) else puissance := x*puissance (x, n-1) ; end ; </pre>	Exercise 11 PGCD EUCLIDE <pre> Function Pgcd (a, b: Integer) : Integer; begin if (a=b) or (b=0) then Pgcd := a else Pgcd := Pgcd (b, a mod b) ; end; </pre>
Exercise 11 PGCD Différence <pre> Function Pgcd (a, b : integer) : integer ; begin if (a=b) then pgcd := a else if (a>b) then pgcd := pgcd (a-b, b) else pgcd := pgcd (a, b-a) ; end; </pre>	Exercise 12 <pre> Function ppcm (a, b, p : integer) : integer; begin if (p mod b)=0 then ppcm := p else ppcm := ppcm (a, b, p+a) ; end; L'appel sera : Writeln ('PPCM = ', ppcm (a, b, a)); </pre>
Exercise 12 <pre> Function ppcm (a, b, c : integer) : integer; begin if (a*c) mod b = 0 then ppcm := a*c else ppcm := ppcm (a, b, c+1) ; end; L'appel sera : Writeln ('PPCM = ', ppcm(a, b, 1)); </pre>	Exercise 13 <pre> Function fibonacci (n : integer) : integer; begin if (n <= 2) then fibonacci := 1 else fibonacci := fibonacci (n - 1)+fibonacci (n - 2); end; </pre>
Exercise 14 <pre> Procedure fact_prem (n, i : integer); begin if (n > 1) then if (n mod i)=0 then begin Write(i, ' '); fact_prem(n div i, i); end else fact_prem(n, i + 1); end; l'appel sera : fact_prem(n, 2); </pre>	Exercise 15 <pre> Function sommediv (i, n:integer):integer; begin if i>n div 2 then sommediv := 0 else if n mod i=0 then sommediv := sommediv (i+1, n)+i else sommediv := sommediv (i+1, n); end; (***** P.P *****) begin write ('N = ');readln (n); if n = sommediv (1, n) then writeln (n,' est parfait') else writeln (n, ' n"est pas parfait'); end. </pre>

Exercise 16

```

Function nb_div (i, n:integer):integer;
begin
  if (i >= n div 2)
    then nb_div := 2
    else if (n mod i = 0)
      then nb_div := nb_div (i+1, n)+1
      else nb_div := nb_div (i+1, n);
end;
(***** P.P *****)
begin
  write ('N = '); readln (n);
  if nb_div (2, n) = 2
    then write (n, ' est premier')
    else write (n, ' est non premier');
end.

```

Exercise 16

```

Function exist_div (i, n:integer):boolean;
begin
  if (i > n div 2)
    then exist_div := false
    else if (n mod i = 0)
      then exist_div := true
      else exist_div := exist_div (i+1, n);
end;
(***** P.P *****)
begin
  write ('N = '); readln (n);
  if exist_div (2, n)
    then write (n, ' est non premier')
    else write (n, ' est premier');
end.

```

Exercise 17

```

Function expn (x : real; n : integer) : real;
(*****)
function fact (n : integer) : integer;
var i, f : integer;
begin
  f := 1;
  for i := 2 to n do f := f * i;
  fact := f;
end;
function puiss (x : real; n : integer) : real;
var i : integer; p : real;
begin
  p := 1;
  for i := 1 to n do p := p * x;
  puiss := p;
end;
(*****)
begin
  if n = 0 then expn := 1
  else expn := expn(x,n-1)+ puiss(x,n) / fact(n);
end;

```

Exercise 20

```

Function max (t:tab ; n:integer):integer;
var m:integer;
begin
  if n=1
    then max := t[1]
    else if max (t, n-1) > t[n]
      then max := max (t, n-1)
      else max := t[n] ;
end;
===== Solution 2 =====
Function max (t:tab ; m,i,n:integer):integer;
begin
  if (i <= n)
    then if (t[i] > m)
      then m := max(t, t[i], i+1, n)
      else m := max(t, m, i+1, n);
  max := m;
end;
L'appel : write ( max(t, t[1], 2, n)) ;

```

Exercise 18

```

Function som (n : integer):real ;
begin
  if n=1
    then som := 1
    else som := 1/n + som (n-1);
end;

```

Exercise 19

```

Function som (n : integer):real ;
Var terme:real; signe:integer;
begin
  if n=0
    then som:=1
    else begin
      if n mod 2 = 0
        then signe := 1
        else signe := -1;
      terme := signe * 1/(2*n+1) ;
      som := terme + som(n-1);
    end;
end;

```

Exercise 21 <pre> Function cherche (t:tab;n,m:integer):integer; begin if n=1 then cherche:=diff else if abs(t[n]-m)<diff then begin diff:=abs(t[n]-m); cherche:=cherche (t,n-1,m); end else cherche:=cherche (t, n-1, m); end; (*****pp*****) BEGIN Saisir (n, m, t); diff:=abs (t[1]-m); diff:=cherche (t, n, m); writeln('Liste des valeurs les plus proches de ',m,' est :'); for i:=1 to n do if abs (m-t[i]) = diff then write(t[i],' '); end. </pre>	Exercise 22 <pre> Function cherche (elm, n : integer ; t:tab) : boolean ; begin if n=0 then cherche:=false else if elm = t[n] then cherche := true else cherche := cherche (elm, n-1, t); end; </pre> solution 2 : <pre> Function cherche (i, n, elm :integer ; t : tab):integer; begin if elm=T[i] then cherche:=i else if i= n then cherche:=0 else cherche:=cherche (i+1, n, elm, t); end; </pre>
Exercise 23 Function Dichotomique (deb, fin, val : integer ; t:tab) : boolean; <pre> var milieu:integer; begin milieu:=(deb+fin) div 2; if val = T[milieu] then dichotomique := true else if (val < T[milieu]) and (deb < milieu) then dichotomique := dichotomique (deb, milieu-1,val,T) else if (val > T[milieu]) and (fin > milieu) then dichotomique := dichotomique (milieu+1, fin,val,T) else dichotomique := false; end; </pre>	Exercise 24 Procedure Decaler (Var t : Tab; p, n : integer); <pre> Begin If p <= n Then Begin Decaler (t, p+1, n); t[p+1] := t[p]; End; End; </pre>
Exercise 25 Procedure Inverser (Var t : Tab; p, n : Integer); <pre> Var aux : Real; Begin If p < n Then Begin aux := t[p]; t[p] := t[n]; t[n] := aux; Inverser (t, p + 1, n - 1); End; End; </pre>	Exercise 26 Procedure permuter (var t : tab ; d, f : integer) ; <pre> var aux:integer; begin if d < f then begin aux:=t[d]; t[d]:=t[d+1]; t[d+1]:=aux; permuter(t,d+2,f); end; end; Appel : permuter (t, 1, n) ; </pre>
Exercise 27 program maximum; <pre> uses wincrt; var a,b,acc:integer; Function max (a,b,acc:integer):integer; begin if a=0 then max:=b+acc else if b=0 then max:=a+acc else max:=max(a-1, b-1, acc+1); end; begin write('a = ');readln(a); write('b = ');readln(b); writeln('La valeur max = ',max(a,b,0)); end. </pre>	Exercise 28 Function inverse (st: string): string; <pre> begin if st = "" then inverse := "" else inverse := st[length(st)] + inverse(copy(st, 1, length(st) - 1)); end; ===== Solution 2 ===== Procedure inverse (var st: string); Var c: char; begin if st <> "" then begin c := st[length(st)]; delete(st, length(st), 1); inverse(st); st := c + st; end; end; </pre>

Exercise 29

```

Procedure tri_selection (n:integer; var T:tab);
var p:integer;
(*****)
function posmax (n:integer; t:tab):integer;
var max,j:integer;
begin
    max:=1;
    for j:=2 to n do
        if t[j]>t[max] then max:=j;
    posmax := max;
end;
(*****)
procedure permuter (x,y:integer; var t:tab);
(*****)
begin
    If n>1 then
        begin
            p := posmax (n, t);
            if p<>n then permuter (n, p, t);
            tri_selection (n-1, t);
        end;
end;
Appel : tri_selection (n, T);

```

```

===== solution 2 =====
Procedure tri_selection (deb,n:integer ; var T:tab);
var pos:integer;
(*****)
function posmax(deb,n:integer;T:tab):integer;
var max,j:integer;
begin
    max:=deb;
    for j:=deb+1 to n do
        if T[j]>T[max] then max:=j;
    posmax:=max;
end;
(*****)
procedure permuter (var x,y:integer);
(*****)
begin
    if deb<>n then
        begin
            pos:=posmax (deb,n,T);
            if pos<>deb then permuter (T[deb], T[pos]);
            tri_selection (deb+1, n, T);
        end;
end;
Appel : tri_selection (1, n, T);

```

Exercise 30

```

Procedure tri_insertion (n:integer ; var t:tab);
(*****)
procedure inserer (n:integer ; var t:tab);
var tmp,j:integer;
begin
    if t[n]<t[n-1] then
        begin
            tmp:=t[n]; j:=n;
            while (j>1) and (t[j-1]>tmp) do
                begin
                    t[j]:=t[j-1];
                    j:=j-1;
                end ;
            t[j]:=tmp;
        end;
end;
(*****)
begin
    if n>=2 then begin
        tri_insertion (n-1, t);
        inserer (n, t);
    end;
end;

```

```

===== solution 2 =====
Procedure tri_insertion (i, n:integer ; var t:tab);
var j, tmp : integer ;
begin
    if (i<=n)
        then begin
            if t[i]<t[i-1]
                then begin
                    tmp:=t[i];
                    j:=i ;
                    while (t[j-1]>tmp) and (j>1) do
                        begin
                            t[j]:=t[j-1];
                            j:=j-1 ;
                        end;
                    t[j]:=tmp;
                end;
            tri_insertion(i+1, n, t);
        end;
end;
Appel : tri_insertion (2, n, t);

```

Exercise 31

```

Procedure tri_bulles
    (var t:tab ; n:integer ; permut:boolean);
var i, aux : integer;
begin
    if (n>1)
        then begin
            for i:=1 to n-1 do
                if t[i]>t[i+1] then begin
                    aux:=t[i];
                    t[i]:=t[i+1];
                    t[i+1]:=aux;
                    permut:=true;
                end;
            if (permut=true) then tri_bulles (t, n-1, false);
        end;
end;
Appel : tri_bulles (t, n, false);

```

```

===== solution 2 =====
Procedure tri_bulles (var t : tab ; n, i, j : integer);
(*****)
procedure permut (var x,y:integer);
(*****)
begin
    if (i<=n-1)
        then begin
            if (j<=n-i)
                then begin
                    if (t[j]>t[j+1])
                        then permut (t[j], t[j+1]);
                    tri_bulles (t, n, i, j+1);
                end ;
            tri_bulles (t, n, i+1, 1);
        end;
end;
Appel : tri_bulles (t, n, 1, 1);

```

Exercise 32

```

Procedure Tri_Shell_Rec (Var t:TAB; n, pas :integer);
Var aux,i : integer;
begin
  If pas > 0 Then
    Begin
      If n > pas Then
        begin
          Tri_Shell_Rec (t, n - pas, pas);
          If t[n] < t[n - pas] Then
            Begin
              aux:= t[n];
              i := n;
              Repeat
                t[i] := t[i - pas];
                i := i - pas;
              Until (i = pas) or (aux > t[i - pas]);
              t[i] := aux;
            End;
          End;
          Tri_Shell_Rec (t, n, pas Div 3);
        End;
      End;
    End;
  End;
end;

```

Exercise 33

```

Procedure tri_fusion (d, f :integer ; var T:tab);
var m : integer;
(*****)
procedure fusionner(var t:tab;d,m,f:integer);
var temp:tab;
    i,j,k:integer;
begin
  i:= d;
  j:= m+1;
  for k:= d to f do
    if (j>f) or((i<=m) and (T[i]<T[j]))
      then begin
        temp[k]:= T[i];
        i:= i+1;
      end
      else begin
        temp[k]:= T[j];
        j:= j+1;
      end ;
    for k:= d to f do T[k]:= temp[k];
  end;
(*****)
Begin
  if f>d then
    begin
      m:= (d+f) div 2;
      tri_fusion (d,m,t); (* trier partie gauche *)
      tri_fusion (m+1,f,t); (* trier partie droite *)
      fusionner (t,d,m,f); (* fusionner *)
    end ;
  End;
end;

```

Exercise 34

```

Procedure supprim(c:char ; n:integer ; var ch:string);
begin
  if n<>0 then
    if ch[n]=c
      then begin
        delete(ch,n,1);
        supprim (c, n-1, ch);
      end
      else supprim (c, n-1, ch);
  end;
end;

```

Exercise 35

```

Function nb_occ (n:byte ; ch:string ; c:char):byte;
begin
  if n=0 then nb_occ:=0
  else if ch[n]=c
    then nb_occ:=nb_occ (n-1, ch, c)+1
    else nb_occ:=nb_occ (n-1, ch, c)
  end;

Appel : writeln(nb_occ (length(ch), ch, c)) ;

```

Exercise 36

```

Function sc (n:integer):integer;
begin
  if n<10
    then sc := n
    else sc := (n mod 10) + sc (n div 10);
  end;
end;

```

Exercise 37

```

Function nbc (n:integer):integer;
begin
  if n <10
    then nbc := 1
    else nbc := 1+ nbc (n div 10);
  end;
end;

```

Exercise 38

```

Function anagram (mot1,mot2:string):boolean;
Var p:integer;
Begin
  if (mot1='') and (mot2='')
    then anagram:=true
  else begin
    p:=pos(mot1[1],mot2);
    if (mot1<>'')and(p>0)
      then begin
        delete(mot1,1,1);
        delete(mot2,p,1);
        anagram:= anagram(mot1,mot2);
      end
      else anagram:=false;
    end;
  end;
end;
end;

```

Exercise 39

```

Function teste (i:integer ; c:char; ch: string):boolean;
begin
  if c = ch[i]
    then teste:= true
    else if i < length(ch)
      then teste := teste (i+1, c, ch)
      else teste := false;
  end;

===== solution 2 =====
Function teste (c:char ; ch:string):Boolean;
begin
  if (ch='') or ((length(ch)=1)and(ch[1]<>c))
    then teste:=false
  else if ch[1]=c
    then teste:=true
    else teste:= teste (c, copy(ch,2,length(ch)-1));
  end;
end;

```

Exercise 40

```

Function suppr_car (c:char;ch : string):string;
begin
  case length(ch) of
    0 : suppr_car:= "";
    1 : if ch[1]=c then suppr_car:= ""
        else suppr_car:= ch;
    else if ch[1]=c
        then suppr_car:= copy(ch,2,length(ch)-1)
        else suppr_car := copy(ch,1,1) +
            suppr_car(c,copy(ch,2,length(ch)-1));
  end;
end;

```

Exercise 42

```

Function val_triangle (c, l: integer):integer;
begin
  if (c=1) or (l=c)
  then val_triangle := 1
  else val_triangle := val_triangle(c,l-1)+
                        val_triangle (c-1,l-1);
end;

```

Exercise 44

```

Function Ack (m, n : integer) : integer;
begin
  if m = 0
  then Ack := n+1
  else if n = 0
  then Ack := Ack (m-1,1)
  else Ack := Ack (m-1, Ack (m, n-1))
end;

===== Implémentation itérative =====
Function ack (m, n :integer):integer;
begin
  while m <> 0 do
    begin
      if n = 0 then n := 1 else n := ack (m, n - 1) ;
      m := m - 1;
    end;
    ack:= n + 1;
  end;
end;

```

Exercise 46

```

Program mini_cal;
uses wincrt;
var e:integer; st:string ;
Function eval0(st: string): integer;
var a, i: integer; signe: char;
begin
  i := length(st);
  repeat
    i := i-1
  until (i = 0) or (st[i] in ['*', '/']);
  val (copy(st, i + 1, length(st) - i), a, e);
  delete(st, i, length(st) - 1 + i);
  if st[i] = '*'
  then eval0 := eval0 (st) * a
  else if st[i] = '/'
  then eval0 := eval0 (st) div a
  else eval0 := a;
end;

```

Exercise 41

```

Function Constructible (mot,lettres : string):boolean;
function suppr_car(c:char;ch : string):string;
function teste (c:char ; ch:string): Boolean;
begin
  if mot=""
  then constructible:=true
  else if teste(mot[1],lettres)
  then if length(mot)=1
        then constructible:=true
        else constructible :=
            constructible(copy(mot,2,length(mot)-1),
                          suppr_car(mot[1],lettres))
        else constructible:=false;
end;

```

Exercise 43

```

Function McCarthy (n: integer): integer;
begin
  if n > 100
  then McCarthy := n - 10
  else McCarthy := McCarthy (McCarthy (n+ 11));
end;

```

Exercise 45

```

Function eval (st: string): integer;
var a, i, e: integer;
begin
  if length(st) = 0
  then eval := 0
  else begin
    i:=1;
    repeat
      i:=i+1;
    until (st[i] in ['+', '-']) or (i>length(st));
    val (copy(st, 1, i - 1),a,e);
    delete(st, 1, i-1 );
    eval := a + eval (st);
  end;
end;

```

Function eval (st: string): integer;

```

var a, i, e: integer;
begin
  if length(st) = 0 then eval := 0
  else begin
    i:=1;
    repeat
      i:=i+1;
    until (st[i] in ['+', '-']) or (i>length(st));
    a:= eval0(copy(st, 1, i - 1));
    delete(st, 1, i-1 );
    eval := a + eval(st)
  end;
end;
begin
  write ('Entrer une chaîne formée d'une somme de
        nombre :'); readln (st);
  write( 'La somme = ', eval (st));
end.

```

Exercise 48

```

Function  valeur(c: char): integer;
begin
  case c of
    'M': valeur := 1000;
    'D': valeur := 500;
    'C': valeur := 100;
    'L': valeur := 50;
    'X': valeur := 10;
    'V': valeur := 5;
    'I': valeur := 1;
  end;
end;
function eval(s: string): integer;
var n1: integer;
begin
  if length(s) = 1
  then eval := valeur(s[1])
  else begin
    n1 := valeur (s[1]);
    if n1 < valeur (s[2]) then n1 := -n1;
    eval := n1 + eval (copy(s, 2, length(s) - 1));
  end;
end;

```

Exercise 47

```

Procedure combinaison(ch, tete: string);
var i: integer;
begin
  if length(ch) = 1
  then begin
    ch:= tete+ ch;
    writeln(ch)
  end
  else
    for i:= 1 to length(ch) do
      begin
        combinaison(copy(ch, 2, length(ch)-1), tete+ ch[1]);
        ch := copy(ch, 2, length(ch)-1) + ch[1] ;
      end ;
    end;
Appel : combinaison (ch, '');

```

Exercise 49

```

Program Chaîne;
Uses Wincrt ;
Var ch: String ;
(*****)
Procedure affiche (ch:string);
begin
  if ch <> ""
  then begin
    Writeln (ch);
    Affiche (copy(ch, 1, length(ch)-1));
  end;
end;
(*****)
Begin
  Readln (ch);
  Affiche (ch);
End.

```

