

FENNI Salah

Exercices en Turbo Pascal



Chebba

Le présent manuel est conforme au programme d'informatique de la 4^{ème} année de l'enseignement secondaire relatif aux sections : Sciences Informatiques, Mathématiques, Sciences Expérimentales et Techniques.

Les énoncés d'exercices sont regroupés par catégories. La difficulté en va croissant depuis les instructions simples jusqu'à l'utilisation des sous programmes (utilisation de procédures et fonctions) et les tris.

Ces exercices sont assez variés et tiennent compte des différents niveaux des élèves.

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.

- | | | |
|--|---|--|
| <input type="checkbox"/> Readln (A) ; | <input type="checkbox"/> Readln (45) ; | <input type="checkbox"/> Readln (A+B) ; |
| <input type="checkbox"/> Readln ('A') ; | <input type="checkbox"/> Readln ('A =', A) ; | <input type="checkbox"/> Writeln (X+2*Y) ; |
| <input type="checkbox"/> Writeln ('A =', A) ; | <input type="checkbox"/> Writeln (A, ' ', B) ; | <input type="checkbox"/> Writeln (A:6:2) ; |
| <input type="checkbox"/> Writeln (5 mod 7 div 2) ; | <input type="checkbox"/> Writeln ('Saisir un réel') ; | <input type="checkbox"/> Writeln (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 **Trapeze**, 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 $ax^2+bx+c=0$ pour tout triplet de réels (a, b, c) donné.

Exercice 14

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

Exercice 15

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

Exercice 16

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

Exercice 17

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

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

Exercice 18

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

Exercice 19

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

Exemple : 10/05/2007 donne 10 Mai 2007.

Exercice 20

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

Exercice 21

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

Exercice 22

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

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

Exercice 23

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

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

Exercice 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** ($X*Y=X+X+X+\dots$).

Choisir la somme qui fait intervenir le minimum de termes.

Exercice 20

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

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

Exercice 21

On remarque que :
 $12 \times 42 = 21 \times 24$
 $12 \times 63 = 21 \times 36$
 $12 \times 84 = 21 \times 48$

Il y a 14 produits qui vérifient la propriété : $(10a + b)(10c + d) = (10b + a)(10d + c)$, où a est différent de b et c est différent de d.

Ecrire un programme Pascal qui fait sortir tous ces entiers.

Exercice 22

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

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

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** :

Sachant que $\text{PGCD}(a, b) = \text{PGCD}(b, r)$, avec $r = a \bmod b$.

Tant que le reste r est non nul, on remplace a par b et b par r. Le dernier reste r non nul est alors le PGCD des deux nombres

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

Exercice 24

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

Tant que $(a \neq b)$ on répète la recherche : $\text{PGCD}(a, b) = \text{PGCD}(a-b, b)$ si $a > b$,

Sinon $\text{PGCD}(a, b) = \text{PGCD}(a, b-a)$

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

Exercice 25

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

Exercice 26

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 à 10^{-4} . La dernière somme calculée est une valeur approchée de sin(x).

Exercice 32

Ecrire un programme Pascal qui donne une approximation de :

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

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

Exercice 33

Soit la somme S_n suivante :

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

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

Exercice 34

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

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

Exercice 35

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

Exemple :

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

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

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

Exercice 36

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

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

Exercice 37

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

Exemple d'exécution :

Entrer un entier : 45

Les sommes consécutives sont :

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

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

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

$$45 = 14+15+16$$

$$45 = 22+23$$

Exercice 38

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

Exemple :

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

```

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

```

Exercice 39

PGCD et PPCM de deux entiers :

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

Exercice 40

Soient A et B deux entiers naturels quelconques non nuls.

On a : $\text{PGCD}(A,B) * \text{PPCM}(A,B) = A*B$

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

Exercice 41

Soient a et b deux réels quelconques. Pour tout entier impair n, supérieur ou égal à 3, on a :

$$a^n + b^n = (a+b) (a^{n-1} - a^{n-2}b + \dots - ab^{n-2} + b^{n-1})$$

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

Exercice 42

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

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

Exemple : Si X = 6182319

$$\text{La somme : } 9-1+3-2+8-1+6 = 22$$

Or 22 est divisible par 11, donc 6182319 l'est aussi.

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

Exercice 43

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

Ecrire un programme Pascal qui permet de calculer la somme :

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

Exercice 44

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

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

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

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

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

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

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

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

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

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

Exercice 45

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

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

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

Ecrire un programme Pascal et l'enregistrer sous le nom **kaprekar.pas** qui permet d'afficher tous les nombres de KAPREKAR inférieurs ou égaux à 1000 en utilisant la méthode décrite ci-dessus.

Exercice 46

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

Exemples :

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

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

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 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 19

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 20

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

Exercice 21

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

Exercice 22

On se propose d'écrire un programme Pascal permettant :

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

Exercice 23

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

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

Exercice 24

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

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

Exemple :

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

Sont réarrangés comme suit :

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

Exercice 25

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

Exemple :

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

Exercice 26

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

Exercice 27

Ecrire, en s'inspirant du tri par sélection, une procédure qui permet de construire à partir d'un tableau T de n entiers un tableau Rang tel que Rang[i] soit l'indice dans T 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 28

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

Exemple :

Si on introduit (a a b c c e d e e e e f f g a a a).
Le programme doit afficher (e, 5).

Exercice 29

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

Exercice 30

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

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

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

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

Exercice 31

Ecrire un programme Pascal intitulé **EI_Frequent** permettant de remplir d'une manière automatique et aléatoire, un tableau T par N entiers (de 0 à 9), avec ($5 \leq N \leq 20$), et d'afficher l'élément qui apparaît le plus dans le tableau T, ainsi que son nombre d'occurrences.

Si plusieurs éléments différents répondent à la condition, le programme doit en fournir le premier.

Exercice 32

Ecrire un programme Pascal intitulé **Sequence** permettant de :

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

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

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

Le programme affichera :

La plus longue séquence est : 6 -2 -2 -1 -1

Exercice 33

Ecrire un programme Pascal permettant de vérifier l'existence d'une chaîne donnée **Ch** dans un tableau **T** de n chaînes de caractères ($2 \leq n \leq 10$).

Les éléments du tableau T ainsi que la chaîne recherchée sont formés de n caractères.

On admet que la chaîne **Ch** est dans le tableau **T** si elle est :

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

Exemple :

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

Si **Ch** = "HOUE"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

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

Si **Ch** = "DOUE"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

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

Si **Ch** = "ROUT"

T	VRAI	EUOD	HOUE	ICTB
	1	2	3	4

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

Exercice 34

Soit **T** un tableau de **N** chaînes de caractères non vides et dont la taille maximale est 5 caractères.

On se propose d'écrire un programme permettant de remplir le tableau **T** par **N** chaînes de caractères ($2 \leq N \leq 30$), puis de calculer et d'afficher la somme des nombres extraits des chaînes de chaque élément du tableau **T**.

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

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

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

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

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

Le programme affichera la valeur de **S**.

Exercice 35

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

Exemple :

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

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

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

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

Exercice 36 Intervalle de plus grande somme

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

Exemple :

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

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

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

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

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 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 20

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 21

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

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

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

Exercice 22

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

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

Exercice 23

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

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

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

Exemple 1 :

Si CH = "aFABzKOikvMx"

Alors le programme affichera la chaîne "FKMAObzkvxai"

Exemple 2 :

Si CH = "bonjour"

Alors le programme affichera la chaîne "bnjroou"

Exercice 24 Plus long palindrome

Étant donné une longue séquence de lettres, votre objectif est de trouver la longueur du plus long palindrome que l'on peut trouver dans cette séquence.

Par exemple, le mot "mollakayakokomassa" contient comme plus grand palindrome le mot "kayak", qui est de longueur 5.

Exercice 25

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

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

Exemple :

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

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

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

Exercice 26

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

Exemple : pour **p=4** et **n=4592**

Les facteurs premiers de N sont : **2, 2, 2, 2, 7, 41**

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

Exercice 27

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

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

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

Exemples :

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

Exercice 28

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

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

Exemple : bonjour → boonnjjjjjooouuuuuurrrrrrr

Correction

**Exercices
En**

**TURBO
PASCAL**

Prof : FENNI Salah

Lycée Ibn Rochd – La Chebba
©® 2000

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)

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

Exercice 15

- 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

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

Exercice 19

- 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

FENNI SALAH

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 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 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 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 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 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.
```

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');
```

```
  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 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 8

```
Program Invite ;
Uses Wincrt ;
Var titre, foulon, term1, term2 : String ;
Begin
  Write ('Titre = '); Readln (titre) ;
  Write ('Votre nom = '); Readln (foulon) ;
  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, ' ', foulon, ' ', soyez l', term1,
    ' bienvenu', term2) ;
End.
```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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 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 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

```

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 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.

```

```

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.

```

Exercice 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.

```

Exercice 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;
  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 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 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 ;
  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 ;
  Repeat
    som := som + u ;
    u := u + 3 ;
    i := i+1 ;
  Until (i>100) ;
  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 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 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 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 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.
  
```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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 ;

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

```

IF y<0 Then Begin
  y:= -y ;
  x:= -x ;
End ;
s := 0 ;
FOR i:=1 To y Do s := s+x ;
Writeln (s) ;
End.

```

Exercice 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.

```

Exercice 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.

```

Exercice 25

```

Program PPCM;
Uses Wincrt ;
Var pcm, m, n, aux : Integer;
Begin
  Repeat
    Writeln ('Saisir deux entiers > 0'); Readln (m, n) ;
  Until (m>0) and (n>0);
  IF m < n Then Begin
    aux:= m;
    m := n;
    n := aux;
  End;
  pcm := m;
  While (pcm mod n <> 0) Do pcm := pcm + m;
  Writeln ('PPCM = ', pcm);
End.

```

Exercice 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.

```

Exercice 28

```

Program Parfait ;
uses wincrt;
var nb, d, som, a, b : integer;
begin
  repeat
    Readln(a,b);
  until (1<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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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:=term2 * -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 (- Pi <= x) and (x <= Pi);
  write ('sin(', x:1:2,') = ', sinus(x):10:10);
end.

```

Exercice 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.

```

Exercice 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;
(*****)
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) ;

```

Exercice 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.

```

Exercice 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.

```

```

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.

```

Exercice 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;
  End.

```

Exercice 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.

```

Exercice 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.

```

Exercice 39

```

Program ppcm_pgcd;
uses wincrt;
var a, b : integer;
(*****)
Procedure saisie (var a,b:integer);
begin
  repeat
    writeln('Donner deux entiers >0');
    readln (a, b);
  until (a>0) and (b>0);
end;
(*****)
Procedure affiche(a,b:integer);
var k:integer;
begin
  k := 0;
  repeat
    k := k+1
  until (a*k) mod b = 0;
  writeln ('ppcm de ', a, ' et ', b, ' = ', a*k);
  writeln ('pgcd de ', a, ' et ', b, ' = ', b div k);
end;
(***** p.p *****)
BEGIN
  saisie(a,b);
  affiche(a,b);
END.

```

Exercice 40

```

Program produit_ab;
uses wincrt;
var a, b,ppcm,pgcd : integer;
(*****)
Procedure pgcd_ppcm(a,b:integer; var pgcd,ppcm:integer);
var k:integer;
begin
  k := 0;
  repeat
    k := k+1
  until (a*k) mod b = 0;
  ppcm:=a*k;
  pgcd:=b div k;
end;
(***** p.p *****)
begin
  writeln('Donner a et b : ');
  readln (a, b);
  pgcd_ppcm(a,b,pgcd,ppcm);
  writeln(a, ' * ',b, ' = ',pgcd*ppcm);
end.

```

Exercice 41

```

Program Exercice_41;
uses wincrt;
var a,b,s:real;
    n,i,signe:integer;
(*****)
Procedure saisie(var a,b:real;var n:integer);
begin
  writeln('donner a et b'); readln(a,b);
  repeat
    writeln('donner n ');
    readln(n);
  until (n>=3) and odd(n);
end;
(*****)
Function power ( x:real;n:integer):real;
var k:integer;
    p:real;
begin
  p:=1;
  for k:= 1 to n do p:=p*x;
  power:=p;
end;
(***** p,p *****)
begin
  saisie(a,b,n);
  s:=0;
  signe:=1;
  for i:=0 to n-1 do
    begin
      s:=s+signe*power(b,i)*power(a,n-i-1);
      signe:=-signe;
    end;
  writeln((a+b)*s:2:2);
end.

```

Exercice 43

```

Program somme ;
Uses WinCrt ;
Var n,p,signe: Integer;
    som:real;
(*****)
Function Comb(p,n:integer):real;
(*****)
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
  comb:=fact(n)/(fact(p)*fact(n-p));
end;
(***** p,p *****)
Begin
  Writeln ('Donner n : ');
  ReadLn (n);
  som:=1;
  signe:=-1;
  for p:=1 to 2*n do
    begin
      som:=som+signe*sqr(comb(p,2*n));
      signe:=-signe;
    end;

  Writeln ('somme = ',som:2:2) ;
End.

```

Exercice 42

```

Program divisible_11;
uses wincrt;
var x:integer;
(*****)
Procedure saisie (var x:integer);
begin
  repeat
    write('Donner un entier ');
    readln(x);
  until x>0;
end;
(*****)
Function divs_11(x:integer):boolean;
var signe,som:integer;
begin
  signe:=1;
  som:=0;
  repeat
    som:=som+signe*(x mod 10);
    x:=x div 10;
    signe:=-signe;
  until x=0;
  divs_11:= som mod 11 = 0;
end;
(***** p,p *****)
begin
  saisie(x);
  if divs_11(x)
  then writeln('divisible par 11')
  else writeln('non divisible par 11');
end.

```

Exercice 44

```

Program divis_7_13;
uses wincrt;
var n,nb:integer;
(*****)
Function div_7 (n : integer) : boolean ;
begin
  while (n>99) do n := (n div 10) - 2 * (n mod 10) ;
  div_7:=(n mod 7 = 0);
end ;
(*****)
Function div_13 (n : integer) : boolean ;
begin
  while (n>99) do n := (n div 10) + 4 * (n mod 10) ;
  div_13:=(n mod 13 =0);
end ;
(***** p,p *****)
BEGIN
  writeln('les nombres divisibles par 7 :');
  nb:=0;n:=0;
  repeat
    if div_7(n) then begin
      write(n:5);
      nb:=nb+1;
      end;

      n:=n+1;
  until nb=100;
  writeln;
  writeln('les nombres divisibles par 13 :');
  nb:=0; n:=0;
  repeat
    if div_13(n) then begin
      write(n:5);
      nb:=nb+1;
      end;

      n:=n+1;
  until nb=100;
END.

```

Exercice 45

```

Program Exercice_45;
uses wincrt;
var k: integer;
(*****
Function kaprekar(m : longint): boolean;
var l,n1,n2,err : integer;
    ch,ch1,ch2 : string;
begin
    str(sqr(m),ch);
    l := length(ch);
    ch1 := copy(ch, 1, l div 2);
    ch2 := copy(ch, (l div 2)+1, l);
    val(ch1,n1,err);
    val(ch2,n2,err);

    kaprekar := (m=n1+n2);
end;
(***** P,P *****)
Begin
    For k:=1 to 1000 do
        if kaprekar(k) then writeln(k);
End.

```

Exercice 46

```

Program Exercice_46;
uses wincrt;
var p,q,n: integer;
(*****
Procedure saisie(var p,q:integer);
begin
    repeat
        write('p='); readln(p);
        write('q='); readln(q);
    until (10<p)and(p<q)and(q<20000);
end;
(*****
Function circulaire(n:integer):boolean;
Var err,i: integer;
    ok: boolean;
    ch: string;

function premier(n:integer):boolean;
var i,d: integer;
begin
    d:=2;
    for i:=2 to (n div 2) do
        if (n mod i)=0 then d:=d+1;
    premier := (d=2);
end;

begin
    ok:=premier(n);
    if ok
        then begin
            str(n,ch); i:=0;
            repeat
                i:=i+1;
                ch:=ch[length(ch)]+copy(ch,1,length(ch)-1);
                val(ch,n,err);
                ok:=premier(n);
            until (i=length(ch)-1) or (not ok);
            end;
            circulaire:=ok;
end;
(***** P,P *****)
BEGIN
    saisie(p,q);
    for n:=p to q do
        if circulaire(n) then write(n, ' ');
END.

```

LES TABLEAUX

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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 ;
    IF i <> k
    then begin
      tmp := T[i] ;
      FOR j:=i Downto k+1 Do T[j] := T[j-1] ;
      T[k] := tmp ;
    end;
  End ;
Writeln ; Writeln ;
FOR i:=1 To n Do Write (T[i]:4) ;
End.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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);

```

Exercice 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;
(******)
Begin
  Repeat
    Writeln ('Donner la taille du triangle : '); Readln (n);
  Until n in [2..15];
  Init (n, T);
  Triangle (n, T);
End.

```

Exercice 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.

```

```

  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.

```

Exercice 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;
(******)
Begin
  Repeat
    Writeln ('Donner la taille du triangle : '); Readln (n);
  Until n in [2..15];
  Triangle (n, T);
  Afficher (n, T);
End.

```

Exercice 18

```

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;

```

```

Procedure Transpose (Var M:Mat; n:integer);
var 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.

```

Exercice 19

```

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.

```

Exercice 20

```

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;
    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);
    End;
  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.

```

Exercise 21

```

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;

```

Exercise 22

```

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.

```

```

(*****)
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;
(*****)
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 24

```

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.

```

Exercice 23

Program tri_bulles_bidirectionnel;

```
uses wincrt;
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,j,aux:integer;
    permut:boolean;
(*****)
begin
    j:=0;
    repeat
        permut:=false;
        j:=j+1;
        for i:=j 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
            begin
                permut:=false;
                for i:=n-1 downto j+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;
            end;
        n:=n-1;
    until (permut=false) or (j>=n);
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.
```

Exercice 25

Program fusion_2_tab;

```
uses wincrt;
type tab1=array [1..20] of integer;
    tab2=array [1..40] of integer;
var v1,v2:tab1;
    v3:tab2;
    n,m,c:integer;
(*****)
procedure lecture (var taille:integer);
begin
    repeat
        readln(taille);
    until taille in [2..20];
end;
(*****)
procedure remplir (var t:tab1; taille:integer);
var i:integer;
begin
    for i:= 1 to taille do readln(t[i]);
end;
(*****)
procedure trier (taille:integer; var t:tab1);
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:tab1; var v3:tab2; 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:tab2; 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 du tableau V1 : '); lecture (n);
write ('Taille du tableau V2 : '); lecture (m);
writeln ('Remplir le tableau V1 :'); remplir (v1,n);
```

Exercice 26

```
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;
```

Exercice 27

```
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;
```

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

```
program long_suite;
uses wincrt, écran;
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.
```

Exercice 29

```

Program symetrique;
uses wincrt;
type  tab=array [1..200] of integer;
var   t:tab;
      n,i:integer;
(*****
  Procedure saisie(var n:integer;var t:tab);
  var i:integer;
  begin
    repeat
      write('N = ');
      readln(n);
    until n in [5..200];
    Randomize;
    for i:=1 to n do
      T[i]:=100+Random(900);
  end;
(*****
  Function verif (x:integer):boolean;
  var ch:string;
  begin
    str(x,ch);
    verif:= ch[1] = ch[3]
  end;
(***** p,p *****)
BEGIN
  saisie(n,t);
  writeln('les nombres symétriques de T sont: ');
  for i:= 1 to n do
    if verif(t[i]) then write(t[i]:4);
END.

```

Exercice 31

```

Program El_frequent;
Uses Wincrt ;
Type  tab1=Array [1..20] of 0..9 ;
      tab2=Array [0..9] of 0..20 ;
Var   T:tab1;
      F:tab2;
      n:integer;
(*****
  Procedure Saisir (var n:integer);
  begin
    Repeat
      writeln('Saisir un entier N, (5<=n<=20)');
      readln(n);
    Until n in [5..20];
  end;
(*****
  Procedure remplir (n:integer;var t:tab1);
  var i:integer;
  begin
    randomize;
    for i:=1 to n do
      begin
        t[i]:=random(10);
        write(t[i]:3);
      end;
    writeln;
  end;
(*****
  Procedure affiche(n:integer;t:tab1;var f:tab2);
  var i,max:integer;
  Begin
    For i:=0 To 9 Do F[i] := 0 ;

```

Exercice 30

```

Program element_manquant ;
uses wincrt;
type  tab=array[1..20] of integer;
var   t:tab;
      n:integer;
(*****
  Procedure saisie(varn:integer;var t:tab);
  var i:integer;
  begin
    repeat
      writeln('Donner le nombre d'éléments N, 2<=n<=20');
      readln(n);
    until n in [2..20];
    repeat
      write('T[1] : '); readln(T[1]);
    until T[1]>=0;
    for i:=2 to n do
      repeat
        write('T['i,'] : ');
        readln(T[i]);
      until T[i]>=T[i-1];
    end;
(*****
  Procedure manque (n:integer;t:tab);
  var x,i,j:integer;
  begin
    write('Les entiers manquants sont : ');
    x:=0;
    for i:=2 to n do
      if (T[i]<>T[i-1]+1)
        then for j:=(T[i-1]+1) to (T[i]-1) do
          begin
            write(j,' ');
            x:=x+1;
          end;
    write(' leur nombre est : ',x);
  end;
(***** p,p *****)
  begin
    saisie(n,t);
    manque(n,t);
  end.

```

Exercice 32

```

Program Sequence;
uses wincrt;
type  tab = array[1..24] of integer;
var   T:tab;
      n,p1,p2:integer;
(*****
  Procedure Saisie (var n:integer ; var T:tab);
  var i:integer;
  begin
    repeat
      write('N = ');
      readln(n);
    until n in [2..24] ;
    for i:=1 to n do
      Repeat
        write('T['i,'] = ');
        readln(T[i]);
      Until (T[i]<>0);
    end;
(*****
  Procedure Recherche (n:integer;t:tab;var p1,p2:integer ) ;
  var s,i,j,max:integer;
  begin
    max:=1;
    for i:=1 to n-1 do
      begin
        s:=0;
        for j:=i to n do
          begin
            s:=s+T[j];
            if (s=0) and (j-i+1>max)
              then begin

```

```

For i:=1 To n Do
    F[T[i]] := F[T[i]] + 1 ;
max:=1;
For i := 2 To 9 Do
    if F[i]>F[max] then max:=i;

writeln(max,', son nombre d'occurrence est ', F[max]);
End;
(***** p,p *****)
BEGIN
    saisir(n);
    remplir(n,t);
    affiche(n,t,f);
END.

```

```

p1:=i;
p2:=j;
max:=j-i+1;
end;

end;

end;
(*****
Procedure Affiche (p1,p2:integer ; t:tab);
var i:integer;
begin
    writeln('La plus longue séquence est :');
    for i:=p1 to p2 do
        write(T[i], ' ');
    end;
(*****p,p*****)
BEGIN
    Saisie(n,t);
    Recherche(n,t,p1,p2);
    Affiche(p1,p2,t);
END.

```

Exercice 33

```

Program Recherche_ch_tab;
uses wincrt;
type tab = array[1..10] of string;
var T : tab;
    ch, message : string;
    n : integer;
(*****
Procedure saisies (var chn:string ; var m:integer ; var A:tab);
var i : integer;
begin
    repeat
        write('Donner un entier : ');
        readln(m);
    until m in [2..10];
    writeln ('Donner les éléments du tableau :');
    for i:=1 to m do
        repeat
            readln(A[i]);
        until length(A[i]) = m;
    repeat
        write('Donner la chaîne à chercher : ');
        readln(chn);
    until length(chn) = m;
end;
(*****
Function recherche (chn:string ; m:integer ; A:tab) : boolean;
var i : integer;
    trouve : boolean;
    invchn : string;
    { ===== }
function inverse (chn:string):string;
var k : integer;
    chinvs : string;
begin
    chinvs := '';
    for k:=1 to length(chn) do chinvs := chn[k] + chinvs;
    inverse := chinvs;
end;
    { ===== }
begin
    invchn := inverse(chn);
    i := 0;
    Repeat
        i:= i+1;
        Trouve := (chn=A[i]) or (invchn=A[i]);
    Until Trouve or (i=m);
    recherche := trouve;
end;
(***** p,p *****)
BEGIN
    saisies(ch, n, T);
    if recherche (ch, n, T)
    then message := 'La chaîne ' + ch + ' existe dans le tableau T'

```

Exercice 34

```

Program Exercice34 ;
uses wincrt;
type tab=array[1..30] of string[5];
var t:tab;
    n:byte;
    s:longint;
(*****
Procedure saisie(var n:byte;var t:tab);
Var i:byte;
begin
    repeat
        write ('n = ');
        readln (n);
    until n in [2..30] ;
    writeln('Entrer ', n,' chaînes de 5 caractères au maximum');
    for i:=1 to n do
        repeat
            write ('T['i,']= ');
            readln(t[i]);
        until length(t[i]) in [1..5];
    end;
(*****
Function Somme(n:byte; t:tab):longint;
var i,j:byte;
    p,s:longint;
begin
    s:=0;
    for i:=1 to n do
        begin
            p:=0;
            for j:=1 to length(t[i]) do
                if t[i][j] in ['0'..'9']
                then p:=p*10+(ord(t[i,j])-ord('0'));
            s:=s+p;
        end;
    somme:=s;
end;
(*****p,p*****)
BEGIN
    saisie(n,t);
    writeln('La somme est : ',somme(n,t));
END.

```

```

else message := 'La chaîne '+ ch + ' n'existe pas dans le
tableau T';
writeln (message);
END.

```

Exercice 35

```

Program recherche_major;
uses wincrt;
type tab=array [1..25] of integer;
var t:tab;
    p, n:integer;
(*****
Procedure saisie(var n:integer; var t:tab);
var i:integer;
begin
    repeat
        write('n = ');
        readln(n);
    until n in [5..25];

    for i:=1 to n do Readln(t[i])
end;
(*****
Function major_existe(n:integer; t:tab; var p:integer): boolean;
var i,j,occ:integer;
begin
    major_existe:=false;
    for i:=1 to n do
        begin
            occ:=0;
            for j:=1 to n do if t[i]=t[j] then occ:=occ+1;
            if occ > (n div 2)
            then begin
                major_existe:=true;
                p:=i;
            end;
        end;
end;
(*****p*****
BEGIN
saisie(n,t);
if major_existe(n,t,p)
then writeln (t[p], ' est majoritaire')
else writeln ('pas d'élément majoritaire');
END.

```

Exercice 37

```

PROGRAM Segmentation ;
uses wincrt ;
type tab=array[1..20] of integer ;
var T: tab; n,i : integer;
(*****
procedure saisie (var n:integer; var T:tab);
begin
    repeat
        write ('n = '); readln (n);
    until n in [5..20] ;
    for i:=1 to n do
        BEGIN
            write ('T[' ,i ,'] = ');
            readln (t[i]);
        END;
end;
(*****
procedure segmenter (n:integer ; var t : tab);

```

Exercice 36

```

Program grande_somme;
uses wincrt;
type tab=array[1..50] of integer;
var n,d,f:integer;
    t:tab;
(*****
Procedure saisies(var n:integer; var t:tab);
var i:integer;
begin
    repeat
        write('n = '); readln(n);
    until n in [5..50];
    for i:=1 to n do
        begin
            write('T[' ,i ,'] = ');
            readln(t[i]);
        end;
end;
(*****
procedure interval(n:integer; t:tab; var d,f:integer);
var max,i,j,s:integer;
begin
    d:=1; f:=1; max:= T[1];
    for i:=1 to n do
        begin
            s:=0;
            for j:=i to n do
                begin
                    s:= s + T[j];
                    if s > max
                    then begin
                        d:=i;
                        f:=j;
                        max:= s;
                    end;
                end;
        end;
end;
(*****
Procedure affiche(n,d,f:integer; t:tab);
var i:integer;
begin
    writeln('La plus grande somme est défini par les valeurs :');
    for i:=d to f do write(t[i], ' ');
end;
(*****p*****
begin
saisies (n,t);
interval (n,t,d,f);
affiche (n,d,f,t);
end.
var p,aux,i,j:integer;
begin
    p:=1;
    for i:=2 to n do
        begin
            if t[i] <= t[p]
            then begin
                aux:=t[i];
                for j:=1 downto p+1 do t[j]:=t[j-1];
                t[p]:=aux;
                p:=j;
            end;
        end;
end;
(*****p*****
begin
saisie (n,t);
segmenter (n,t);
for i:=1 to n do write(t[i], ' ');
end.

```

LES CHAINES DE CARACTERES

Exercice 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.

```

Exercice 4

```

Program Chaîne_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.

```

Exercice 5

```

Program Chaîne1 ;
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.

```

Exercice 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.

```

Exercice 3

```

Program chaîne_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;

```

Exercice 5

```

Program Chaîne2;
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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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.

```

Exercice 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 devoiler (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);
    devoiler (se, let, ma);
    Writeln (ma);
  Until (nb=Length (se)) Or (ma=se);

```

Exercice 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.

```

Exercice 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;
(*****)
Function inverse (ch : String) : String;
Var i : Integer; chinvs : String;
Begin
  chinvs:="";
  FOR i:=Length (ch) Downto 1 Do
    chinvs:=chinvs+ch[i];
  inverse:=chinvs;
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.

```

```

IF ma=se
Then Writeln ('Bravo, trouvé en ', nb, ' propositions')
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 Ucase (rep) = 'N';
End.

```

Exercice 17

```

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;

```

```

(**** 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.

```

Exercice 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.

```

Exercice 20

```
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;
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.
end.
```

Exercice 21

```
Program Totalogram;
uses wincrt;
var ch:string;
(*****)
Function lettre_maj_esp (ch:string):boolean;
var i:integer;
    test:boolean;
begin
  i:=1; test:=true;
  while (i<=length(ch)) and (test) do
    if ch[i] in ['A','Z',' ']
      then i:=i+1
      else test:=false;
  lettre_maj_esp:=test;
end;
(*****)
Function totalogramme (ch:string):boolean;
var p:integer; test:boolean;
begin
  ch:=ch+' ';
  repeat
    p:=pos(' ',ch);
    test:=ch[1]=ch[p-1];
    delete(ch,1,p);
  until (test=false) or (ch="");
  totalogramme:=test;
end;
(***** p,p *****)
begin
  repeat
    writeln('Saisir une chaîne en majuscule :');
    readln(ch);
  until lettre_maj_esp(ch);
  if totalogramme(ch)
    then writeln('totalogramme')
    else writeln('non totalogramme');
end.
```

Exercice 19

```
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;
        ligne = lignsuiv;
      end;
    end.
end.
```

Exercice 22

```
Program ch_distincte;
uses wincrt;
var ch:string;
(*****)
procedure saisie (var ch:string);
begin
  repeat
    writeln('Saisir une chaîne non vide :');
    readln(ch);
  until ch<>'';
end;
(*****)
Function distincte (ch:string):boolean;
var i:integer;
    test:boolean;
begin
  i:=0;
  test:=true;
  repeat
    i:=i+1;
    if pos(ch[i],ch)<>i then test:=false;
  until (test=false) or (i=length(ch));
  distincte:=test;
end;
(***** p,p *****)
begin
  saisie(ch);
  if distincte(ch)
    then writeln('cette chaîne est distincte')
    else writeln('cette chaîne est non distincte');
end.
```

Exercice 23

```

Program Exercice_23;
uses wincrt;
var ch:string;
(*****)
Procedure saisie(var ch:string);
var i:integer;
begin
  repeat
    writeln('saisir une chaine alphabétique :');
    readln(ch);
    i:=1;
    while upcase(ch[i]) in ['A'..'Z'] do i:=i+1;
  until (i>length(ch)) and (length(ch) in [1..50]);
end;
(*****)
Function construire (ch:string):string;
var i, p1, p2, p3 : integer;
    res:string;
begin
  res:="";
  p1:=1; p2:=1; p3:=1;
  for i:=1 to length(ch) do
    case ch[i] of
      'A'..'Z' : if not (ch[i] in ['A','E','I','O','U','Y'])
        then begin
            insert(ch[i],res,p1);
            p1:=p1+1;
            p2:=p2+1;
            p3:=p3+1;
          end
        else begin
            insert(ch[i],res,p2);
            p2:=p2+1;
            p3:=p3+1;
          end;
      'a'..'z' : if not (ch[i] in ['a','e','i','o','u','y'])
        then begin
            insert(ch[i],res,p3);
            p3:=p3+1;
          end
        else res:=res+ch[i];
    end; {fin selon}
  construire:=res;
end;
(*****p,p*****)
BEGIN
  saisie(ch);
  writeln('La chaine devient : ',construire(ch));
END.

```

Exercice 25

```

Program Cryptage ;
uses wincrt;
var ch,ch2:string;
(*****)
procedure saisie(var ch:string);
function verif(ch:string):boolean;
var i:integer;
    ok:boolean;
begin
  i:=0;
  repeat
    i:=i+1;
    ok:= upcase(ch[i]) in ['A'..'Z'],' '];
  until (not ok) or (i=length(ch));
  verif:=ok;
end;
begin
  repeat
    write('Phrase initiale = ');
    readln(ch);
  until (pos(' ',ch)= 0) AND (verif(ch)=true);
end;
(*****)

```

Exercice 24

```

Program long_palindrome;
uses wincrt;
var ch,ch1,max:string;
    i,j:integer;
(*****)
Function palindrome(ch:string):boolean;
var i:integer;
    verif:boolean;
begin
  i:=0;
  repeat
    i:=i+1;
    verif := (ch[i] = ch[length(ch)-i+1]);
  until (verif=false) or (i=length(ch) div 2);
  palindrome:=verif;
end;
(*****p,p*****)
begin
  write('Donner ch = ');readln(ch);
  max:=ch[1];
  repeat
    for j:=length(ch) downto 3 do
      begin
        ch1:=copy(ch,1,j);
        if palindrome(ch1) and (length(ch1)>length(max))
          then max:=ch1;
      end;
    delete(ch,1,1);
  until length(ch)=2;
  writeln ('La plus longue chaine palindrome est : ', max);
end.

```

Exercice 26

```

Program facteurs_premiers;
uses wincrt;
var p:integer;
    n:longint;
(*****)
Procedure saisie(var p:integer;var n:longint);
var ch:string;
begin
  repeat
    write('p = ');
    readln(p);
  until (2<p) and (p<6);
  repeat
    write('Donner un entier de ',p,' chiffres : ');
    readln(n);
    str(n,ch);
  until length(ch)=p ;
end;
(*****)
Function Facteurs(n:longint):string;
var ch,chc,cbd:string;
    d,c:integer;
begin
  d:=2;ch:="";
  repeat

```

```

function crypter(ch:string):string;
var i,p:integer;
begin
  if ch[1]=' '
    then p:=0
    else p:=1;

  for i:=1 to length(ch) do
    if ch[i]<>' '
      then if (ord(uppercase(ch[i]))+p) <= ord('Z')
        then ch[i]:=chr(ord(ch[i])+p)
        else ch[i]:=chr(ord(ch[i])+p-26)
      else p:=p+1;
  crypter:=ch
end;
(*****p,p*****)
BEGIN
  saisie(ch);
  ch2:=crypter(ch);
  writeln('Phrase cryptée = ',ch2);
END.

```

Exercice 27

Program romain_decimal;

```

uses WinCrt;
var ch : string;
(*****)
procedure saisie(var ch : string);

```

```

function valide(ch : string):boolean;
var i : integer;
    ok : boolean;
begin
  i:=0;
  repeat
    i:=i+1;
    ok:= ch[i] in ['M','D','C','L','X','V','I']
  until (not ok) or (i=length(ch));
  valide := ok;
end;

```

```

begin
  repeat
    Write('Entrer un nombre en chiffres romains : ');
    Readln(ch);
  until valide(ch);
end;

```

Exercice 28

```

program duplicate_chaine;
uses wincrt;
var ch:string[20];
    res:string;
    i,j:integer;
begin
  write('ch = ');readln(ch);
  res:="";
  for i:=1 to length(ch) do
    for j:=1 to i do
      res:=res+ch[i];
  writeln(res);
end.

```

```

  if (n mod d) = 0
  then begin
    c:=0;
    repeat
      c:=c+1;
      n:=n div d;
    until (n mod d)<>0;
    str(c, chc);
    str(d, chd);
    ch:=ch+chc+chd;
    end
  else d:=d+1;
  until (n=1);
  facteurs:=ch;
end;
(*****p,p*****)
BEGIN
  saisie(p,n);
  Writeln(facteurs(n));
END.

```

```

(*****)

```

function convert(ch : string):integer;

```

var i, s, v, v2 : integer;

```

function decimal(c : char):integer;

```

begin
  case c of
    'I' : decimal := 1;
    'V' : decimal := 5;
    'X' : decimal := 10;
    'L' : decimal := 50;
    'C' : decimal := 100;
    'D' : decimal := 500;
    'M' : decimal := 1000;
  end;

```

```

end;
begin
  s := 0;
  for i:=1 to Length(ch) do
    begin
      v := decimal(ch[i]);
      if (i < Length(ch))
        then begin
          v2 := decimal(ch[i+1]);
          if (v < v2) then v:= -v;
        end;
      s := s + v;
    end;
  convert := s;

```

```

end;
(*****p,p*****)

```

```

begin
  Saisie(ch);
  Writeln(ch , ' = ', convert(ch));
end.

```

STRUCTURE GENERALE D'UN PROGRAMME PASCAL

```
PROGRAM Nom_programme ; {En-tête du programme}*
Uses ... ; {Utilisation des unités / bibliothèques}*
Const ... ; {Déclaration des constantes}*
Type ... ; {Déclaration des types}*
Var ... ; {Déclaration des variables}*
{===== Définition des procédures =====}*
Procedure Nom_procedure (pf1 : type1 ; Var pf2 : type2 ; ... ; pfn : typen) ;
  {Déclarations locales : Const, Type, Var, Function, Procedure, ...}*
Begin
  Instructions de la procédure ;
End ;
{===== Définition des fonctions =====}*
Function Nom_fonction (pf1 : type1 ; pf2 : type2 ; ... ; pfn : typen) : Type_résultat ;
  {Déclarations locales : Const, Type, Var, Function, Procedure, ...}*
Begin
  Instructions de la fonction ;
  Nom_fonction := résultat ;
End ;
{===== P. P. =====}
BEGIN {Début du programme principal}
  Instructions ;
  ..... ;
  {Bloc principal du programme avec appel des procédures et des fonctions}
END. {Fin du programme}
```

* : facultatif