

TURBO PASCAL

Exercices Corrigés

Prof : FENNI Salah

Lycée Chebba

© 2006

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"/> Lire (A) | <input type="checkbox"/> Lire (45) | <input type="checkbox"/> Lire (A+B) |
| <input type="checkbox"/> Lire ("A") | <input type="checkbox"/> Lire ("A =", A) | <input type="checkbox"/> Ecrire (X+2*Y) |
| <input type="checkbox"/> Ecrire ("A =", A) | <input type="checkbox"/> Ecrire (A, " ", B) | <input type="checkbox"/> Ecrire (A:6:2) |
| <input type="checkbox"/> Ecrire (5 mod 7 div 2) | <input type="checkbox"/> Ecrire ("Saisir un réel") | <input type="checkbox"/> Ecrire (45) |

Exercice 2

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

Exercice 3

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

Exercice 4

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

Exercice 5

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

Exercice 6

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

Exercice 7

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

Exercice 8

Proposer une marche à suivre qui fait, une **permutation circulaire** à droite, des valeurs de trois variables A, B et C.

Par exemple : à partir de $(A, B, C) = (10, 25, 4)$, on passe à $(A, B, C) = (4, 10, 25)$.

Exercice 9

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

Exercice 10

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

Exercice 11

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

Exercice 12

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

Exercice 13

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

Exercice 14

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

Exercice 15

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

Exercice 16

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

Exercice 17

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

Exercice 18

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

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

Exercice 19

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

Exemple : pour N=427

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

Exercice 20

Ecrire un algorithme puis la traduction en Pascal d'un programme, qui fait entrer deux entiers A et B et fait calculer et afficher leur **inf** et leur **sup**.

Utiliser une formule mathématique donnant directement le résultat.

LES STRUCTURES CONDITIONNELLES

Exercice 1

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

Exercice 2

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

Exercice 3

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

Exercice 4

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

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

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

Exercice 5

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

Exercice 6

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

Exercice 7

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

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

Exercice 8

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

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

Exercice 9

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

Exercice 10

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

Exercice 11

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

Exercice 12

Ecrire un programme Pascal intitulé **EQUA_1D**, qui fait résoudre dans IR l'équation **ax+b=0** pour tout couple de réels (a, b) donné.

Exercice 13

Ecrire un programme Pascal intitulé **EQUA_2D**, qui fait résoudre dans IR l'équation $ax^2+bx+c=0$ 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.

LES STRUCTURES ITERATIVES

Exercice 1

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

Exercice 2

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

Exercice 3

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

Exercice 4

Considérons la suite définie par la relation : $U_{n+1} = U_n + 3$ et $U_1 = 2$

On veut calculer la somme de 100 premiers termes de cette suite. Donner trois solutions distinctes en utilisant les boucles : **POUR**, **TANT QUE** et **REPETER**.

Exercice 5

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

Exercice 6

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

Exemple : (pour nL=5)

```

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

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

Exemple : (pour nL=6)

```

          1
        232
      34543
    4567654
  567898765
67890109876

          1
         121
        12321
       1234321
      123454321
     12345654321
```

Exercice 7

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

```

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

Exercice 8

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

Exercice 9

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

Exercice 10

Ecrire un programme Pascal qui permet de saisir un entier n, 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 :

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

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

Exercice 33

Soit la somme S_n suivante :

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

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

Exercice 34

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

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

Exercice 35

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

Exemple :

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

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

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

Exercice 36

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

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

Exercice 37

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

Exemple d'exécution :

Entrer un entier : 45

Les sommes consécutives sont :

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

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

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

$$45 = 14+15+16$$

$$45 = 22+23$$

Exercice 38

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 :

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

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 :

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

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.

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 = $(note1 + 2 * note2) / 3$, 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 plus longue suite de caractères identiques et d'afficher le caractère concerné ainsi que le nombre de fois qu'il est répété.

Exemple :

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

Le programme doit afficher (e, 5).

Exercice 29

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

Exercice 30

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

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

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

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

Exercice 31

Écrire un programme Pascal intitulé **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.

LES CHAINES DE CARACTERES

Exercice 1

Question de cours :

On introduit le type suivant : **Type chain = String [63]** ;

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

Exercice 2

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

Exercice 3

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

Exercice 4

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

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

Exercice 5

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

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

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

Exercice 6

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

Exemple : si la chaîne est : ' _Travaux_ _pratique_ _pascal_ _ '

Alors l'exécution du programme donnera la chaîne = 'Travaux_pratique_pascal'

Exercice 7

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

Par exemple :

'RESOLUTION DE PROBLEMES' devient 'PROBLEMES DE RESOLUTION'

Exercice 8

On veut écrire un programme Pascal permettant de :

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

Exercice 9

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

Exercice 10

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

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

Résultat : "Cile"

Exercice 11

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

Par exemple : DEVOIR devient D_E_V_O_I_R

Exercice 12

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

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

Exercice 13

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

Exercice 14

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

Exercice 15

Le Jeu du **PENDU** consiste à :

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

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