



- 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.
- ← Ecrire un programme qui permet de saisir une chaine de caractères et d'extraire les caractères non alphabétiques.
- Ecrire un programme qui permet d'afficher les entiers de 4 chiffres qui vérifient cette propriétés abcd a+b=c+d exemple 7263 (7+2=6+3).
- ÆEcrire un programme qui permet d'afficher la table de multiplication.

Exercice Nº2:

Ecrire un programme qui permet de remplir un tableau de n caractères et compter nombres de lettres alphabétiques (nombre de voyelles et nombre de consonnes) et nombre de chiffres et le nombre de symboles.

Exercice N°3:

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 Nº4:

Ecrire un programme Pascal qui saisit un tableau A de n chaînes de caractères, cherche et affiche la longueur de la chaîne la plus longue puis toutes les chaînes ayant cette longueur.



Exercice Nº1:

Ecrire un programme qui défile le nom et le prénom de l'utilisateur de telle façon que la chaîne formée du nom et du prénom rentre par la droite et sort par la gauche. Temporiser ce défilement en provoquant un ralentissement sensible.

Exercice Nº2:

Ecrire un programme qui permet d'afficher les formes suivantes (de taille de n)

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

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

```
1 1 121
34543 12321
4567654 1234321
567898765 123454321
67890109876 12345654321
```

Partie III



Exercice Nº1:

```
    O) Début Algo 2
    1) i ← 2
    f ← vrai
    Répéter
    Si T [i-1] > T[i] Alors f ← faux
    Sinon i ← i + 1
    FinSi
    Jusqu'a (NON (f)) ou (i = n + 1)
    2) Si f alors écrire (" les valeurs sont correctes")
    Sinon écrire (" les valeurs ne sont pas correctes")
    FinSi
    3) Fin Algo 2
```

Soit TAB un type utilisateur représentant un tableau de 20 réels et T un tableau de n éléments

- 1. Tracer le TDNT et TDO
- 2. Soit le Tableau T suivant :
 [12 | 15.5 | 20 | | 23.5 | 29 |
 Proposer une valeur à mettre dans la 4ème
 case du T pour que le programme affiche le
 message " les valeurs sont correctes"
 - 3. En déduire le rôle

Exercice N°2

```
Question:

1.Tracer le TDO

2. Exécute à la main le programme pour

X=4 et n =3

3. En déduire le rôle du programme
```



Exercice Nº1

Pour déterminer le carré d'un entier naturel N, on utilise une méthode dont le principe est le suivant : Le carré d'un entier N est la somme des n entiers n premiers entiers impairs

Exemple:
$$N = 10 \rightarrow 10^2 = 1+3+5+7+9+11+13+15+17+19 = 100$$

Ecrire un programme qui utilise la méthode décrite ci-dessus pour retourner le carré d'un entier naturel.

Exercice Nº2

Un entier est dit distinct s'il est composé de chiffres tous différents. Exemples :

- N=1273 est dit distinct
- N= 1565 est dit non distinct

Ecrire un programme Pascal qui permet d'afficher tous les nombres distincts compris dans l'intervalle [1000..2000].

Exercice N°3:

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

Si la différence entre La somme de chiffres de rang, et la somme de chiffres de rang impairs est divisible par 11 **Exemple :** Si X = 6182319

La somme de rang pair: 1+2+1 =4 La somme de rang impair: 6+8+3+9=26

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

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

Exercice Nº4:

On considère une suite V définie par:

$$\begin{cases} U_0 = 1 \\ U_{n} = U_{n-1} + 5 \end{cases}$$

- 1-Quel est l'ordre de récurrence de cette suite? Justifiez votre réponse.
- 2- Afficher le Nième terme de la suite
- 3-Afficher la somme de 10 premiers termes de la suite.

Exercice N°5:

Soit la suite U définie par

$$\begin{cases} U_1 = 1 \\ U_2 = 2 \\ U_n = U_{n-1} + (a^*U_{n-2}) \text{ pour } n > 2 \text{ et a un entier strictement positif donn\'e} \end{cases}$$

- 1. Quel est l'ordre de récurrence de cette suite?
- 2. Ecrire un algorithme d'un programme qui permet vérifier si un entier P est un terme de la suite U.

Exercice Nº6:

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 Nº7:

Ecrire un programme Pascal permettant de décomposer un entier N donné ($2 \le N \le 100$) 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. Si n=20 alors on affiche 20=2*2*5 .Si n = 17 alors on affiche 17=1*17.



Exercice Nº1

Ecrire un programme qui permet d'afficher un graphique à l'aide du contenu d'un tableau T de 10 entiers <u>aléatoires</u> entre 1 et 20

													12			12		
т	7	4	6	10	12	9	3	12	2	11]		12			12		11
•		-							_	11		10	12			12		11
	!	!			!		ı	•			•	10	12	9		12		11
												10	12	9		12		11
									7			10	12	9		12		11
									7		6	10	12	9		12		11
									7		6	10	12	9		12		11
									7	4	6	10	12	9		12		11
									7	4	6	10	12	9	3	12		11
									7	4	6	10	12	9	3	12	2	11
									7	4	6	10	12	9	3	12	2	11

Exercice Nº2

Remplir un tableau T par n (avec 5<n<15) nombres complexes sous la forme a+bi avec a et b deux entiers naturels non nuls, puis remplir un $2^{\text{ème}}$ tableau M par les modules des éléments de T ,sachant que le module d'un nombre complexe sous la forme a+bi égal de $\sqrt{a^2+b^2}$ afficher les nombres ayant le plus petit module.

Exemple T (n = 5)	2+3i	2+15i	23+3i	3+2i	15+2i
M =	3.60	15.13	23.19	3.60	15.13

Les nombres ayant le plus petit module sont : 2+3i, 3+2i

Exercice N°3

Remplir un tableau par n chaines de caractères non vides, puis remplir un 2^{ème} tableau M par les nombres de voyelles et les nombres de consonnes de chaque élément de T (le nombre de voyelles et les nombre de consonnes dans M séparent par un slache "/") puis affiches les nombres de jolis-mots (un joli mot : si le nombre voyelles >= nombre de

consonnes) . Exemple
Nombre de jolis-mots
= 2

Lycee	Inofs	eleve	math	sport
3/2	2/3	3/2	1/3	1/4

Exercice Nº4

Une adresse IP est un numéro d'identification qui est attribué à chaque machine connectée à un réseau informatique. Une adresse IP valide représentée sous la forme W.X.Y.Z avec W, X, Y et Z sont 4 entiers naturels appartenant chacun à l'intervalle [0. . 255] et séparés par le caractère ".".

Ecrire programme qui permet de :

- Remplirun tableau T par n (5<n<20) adresse IP valide
- Remplir un deuxième tableau V par la classe de chaque adresse IP de T

Il existe cinq classes d'adresses IP

Classe A: W appartient à l'intervalle [0..127]

Classe B: W appartient à l'intervalle [128..191]

Classe C: W appartient à l'intervalle [192.. 223]

Classe D: W appartient à l'intervalle [224..239]

Classe E: W appartient à l'intervalle [240..255]

- Afficher la classe dominante dans T

Exe	emple						
n =	8						
T =	:						
211.120.3.25	212.120.32.1	20.210.110.6	251.251.1.1	99.102.145.20	226.33.10.250	65.30.122.2	135.85.100.2
V	=						
С	С	A	E	A	D	A	В
Lo	Classe Domin						

Exercice N°5

On se propose d'écrire un programme qui permet de saisir un entier naturel \mathbf{N} (4< N < 20), puis remplir un tableau \mathbf{T} par \mathbf{N} équations de second degré de la forme $ax^2+bx+c=0$ avec a, b et c trois entiers naturels non nuls, en suite remplir un deuxième tableau \mathbf{M} par les solutions de chaque équation de \mathbf{T} , enfin afficher les nombres d'équations ayant deux solutions dans IR, les nombres d'équations ayant une solution double dans IR et les nombres d'équations n'ayant pas des solutions dans IR

Pour réaliser le traitement demandé on suivra les étapes suivantes :

- Remplir un tableau T par n équations de second degré.
- ◆ Remplir le tableau M par les solutions de chaque équation de T de façon à ce que M[i] soit les solutions d'équation de T[i]. { si l'équation admet deux solution, séparer les deux solutions par un slache double « // », si l'équation n'admet pas de solutions M[i] ← " Pas de solutions"
- ♦ Compter les nombres d'équations ayant deux solutions dans IR Compter les nombres d'équations ayant une solution double dans IR Compter les nombres d'é quations n'ayant pas des solutions dans IR.

Exemple: N = 5

T $2x^2 + 8x + 6 = 0$ $3x^2 + 2x + 5$ $1x^2 + 4x + 4 = 0$ $1x^2 + 1x + 1 = 0$ $4x^2 + 4x + 1 = 0$

M X1=-1.00//X2=-3.00 Pas de Solutions X1=X2=-2.00 Pas de Solutions X1=X2=-0.50

Les nombres d'équations ayant deux solutions dans IR: 1

Les nombres d'équations ayant une solution double dans IR: 2

Les nombres d'équations n'ayant pas des solutions dans IR : 2 Notation : $X^2 \rightarrow X^2$

Ecrire un programme qui de résoudre ce problème

