

# Série d'exercice

## Corrigé

Préparé par : Zouari Lazhar

Professeur d'enseignement secondaire

Lycée El-Omrane Supérieur

2006 / 2007

**Exercice N° 01**

Ecrire un programme en Pascal qui permet de convertir un temps donné en secondes en heures, minutes et secondes.

**Exemple :** Si temps = 3674 alors le programme affichera : 3674 s = 1 h : 1 min : 14 s

**Exercice N° 02**

Soit l'algorithme suivant :

- 0) Début inconnu
- 1) Ecrire ("Tapez un caractère : "), lire (c1)
- 2) Si ( (ORD (c1) ≥ 97) et (ORD (c1) ≤ 122) )

Alors c2 ← CHR (ORD (c1) - 32)

Sinon c2 ← c1

Fin Si

- 3) Ecrire (c2)
- 4) Fin inconnu

**Travail demandé**

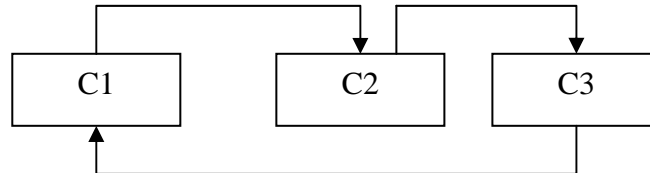
1. Lancer Turbo Pascal
2. Traduire cet algorithme en Pascal
3. Que fait cet algorithme ? (Ecrire la réponse comme commentaire à la fin du programme).
4. Enregistrer votre Travail dans D : qui a comme nom Devoir

**Exercice N° 03**

Ecrire un programme en Pascal permet de déterminer et d'afficher le successeur et le prédécesseur d'un caractère c donnée.

**Exercice N° 04**

Ecrire un programme en Pascal permet d'afficher le résultat d'une permutation circulaire de droite à gauche de trois caractères donnés.

**Exercice N° 05**

Ecrire un programme en Pascal qui permet d'extraire puis afficher les chiffres de centaine, dizaine et unité d'un entier composé de trois chiffres.

**Exercices N° 06**

Ecrire un programme en Pascal qui permet de calculer et d'afficher la distance entre deux points dont les coordonnées sont données.

Soit les points M (a, b) et N (c, d) ; la distance entre eux est donnée par la formule suivante :

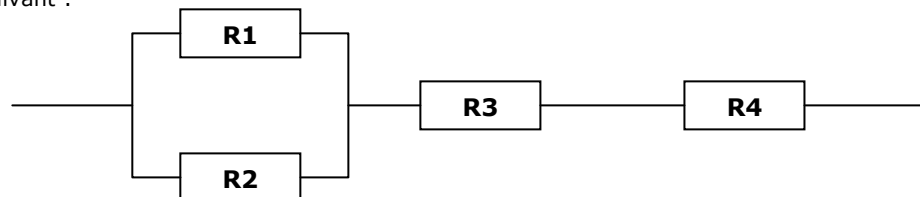
$$d(M, N) = \sqrt{(a - c)^2 + (b - d)^2}$$

**Exercice N° 07**

Ecrire un programme en Pascal intitulé CAPACITE, qui convertit en Octets, Kilo octets, Mega octets et Giga octets un nombre donnée en bits.

**Exercice N° 08**

Soit le schéma électrique suivant :



Ecrire un programme nommé RESISTANCE, qui saisi les valeurs des résistances R1, R2, R3 et R4 puis affiche la résistance équivalente RE du circuit.

On rappelle que :

- La résistance équivalente de 2 ou plusieurs résistances montées en série est leur somme.
- La résistance équivalente de deux ou plusieurs résistances montées en parallèle a un inverse égal à la somme des inverses.

**Exercice N° 09**

Soit c et ch deux variables données tel que c est un caractère et ch est une chaîne de caractère. On vous demande d'afficher la deuxième position de c dans ch.

**Exemple :**

Pour c = "m" et ch = "programmation", le programme affichera : 8

Pour c = "a" et ch = "programmation", le programme affichera : 9

Pour c = "g" et ch = "programmation", le programme affichera : 4

Pour c = "k" et ch = "Zouari", le programme affichera : 0

**Exercice N° 10**

Soit chif et n deux variables données tel que chif est un chiffre décimal et n un entier strictement positif. On veut demande d'afficher la deuxième position de chif dans n.

**Remarque :** Chif et n deux variables de type entier

**Exemple :**

Pour chif = 0 et n = 2006, le programme affichera : 3

Pour chif = 0 et n = 3764, le programme affichera : 0

Pour chif = 0 et n = 30764, le programme affichera : 2

**Exercice N° 11**

Ecrire un programme qui permet de saisir le sexe (M/F), la taille (cm), et le poids (kg) d'une personne et d'afficher :

1. PI, le poids idéal d'une personne, sachant que ce poids théorique est donné par la formule de Lorenz comme suit :
  - Pour un homme :  $PI = (taille - 100) - (taille - 150) / 4$
  - Pour une femme :  $PI = (taille - 100) - (taille - 120) / 4$
2. BMI, l'indicateur d'obésité (Body Mass Index) où  $BMI = poids / taille^2$  avec taille en mètre
3. Si une personne est considérée comme : Normale ( $BMI \leq 27$ ), ou obèse ( $BMI > 27$ ) ou Malade ( $BMI \geq 32$ )

**Exercice N° 12**

Ecrire un programme qui, à partir du numéro d'un mois, permet d'afficher la saison.

**Exemple :**

Si mois = 6 Alors le programme affiche Eté

Si mois = 3 Alors le programme affiche Printemps

Si mois = 9 Alors le programme affiche Automne

Si mois = 1 Alors le programme affiche Hiver

**Exercice N° 13**

Ecrire un programme qui, à partir du numéro d'un mois, permet d'afficher le nombre de jours qui lui correspond.

**Exemple :**

Si mois = 1 Alors le programme affiche : Le nombre de jours du mois 1 est de 31 jours

Si mois = 4 Alors le programme affiche : Le nombre de jours du mois 4 est de 30 jours

Si mois = 2 et Année = 2004 Alors le programme affiche : Le nombre de jours du mois 2 est de 29 jours

Si mois = 2 et Année = 2006 Alors le programme affiche : Le nombre de jours du mois 2 est de 28 jours

**Exercice N° 14**

Ecrire un programme qui, à partir de la saisie de deux réels et un opérateur affiche le résultat après exécution de l'opération choisie.

**Exemple :**

Si les entiers sont 14 et 2 et l'opérateur est "+" alors le résultat affiché est  $14.00 + 2.00 = 16.00$

Pour les mêmes entiers et si l'opérateur est "/" alors le résultat affiché est  $14.00 / 2.00 = 7.00$

**Exercice N° 15**

Ecrire un programme permettant de saisir un tableau T de n réels ( $4 < n < 100$ ) et de trouver et d'afficher le nombre d'occurrences d'un réel R dans T

**Exemple :**

Si n = 10

T

2	4	2.5	0	4	5	7	8.8	10	4
---	---	-----	---	---	---	---	-----	----	---

Et si R = 4 alors le résultat affiché est : 4.00 existe 3 fois dans T

Mais, si R = 12.2 alors le résultat affiché est : 12.20 existe 0 fois dans T

**Exercice N° 16**

Ecrire l'analyse, l'algorithme et la traduction en pascal d'un programme intitulé FACTORIELLE, qui permet de lire un entier nb positif puis de calculer et afficher son factoriel.

**Exemple :**  $5! = 1 * 2 * 3 * 4 * 5 = 120$  alors le programme doit afficher  $5! = 120$ .

**Exercice N° 17**

Écrire l'analyse, l'algorithme et la traduction en Pascal d'un programme intitulé PARFAIT qui permet d'afficher les 4 premiers nombres parfaits.

Un nombre parfait est un nombre présentant la particularité d'être égal à la somme de tous ses diviseurs, excepté lui-même. Le premier nombre parfait est 6, il est bien égal à  $1 + 2 + 3$ , qui sont des diviseurs de 6.

**Exercice N° 18**

Écrire l'analyse, l'algorithme et la traduction en Pascal d'un programme intitulé RECH\_CUBIQUE qui permet de chercher et afficher tous les entiers cubiques de 3 chiffres.

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

**Exercice N° 19**

Ecrivez un programme qui saisit un texte à mettre dans une variable chaîne ch puis chercher toutes les occurrences des lettres de l'alphabet et les met dans un tableau OCC. Les indices des éléments de OCC vont de "A" à "Z". On suppose que le texte n'utilise pas de caractères accentués. Le programme affiche ensuite chaque lettre suivie de ses occurrences dans le texte.

On rappelle que les occurrences d'une lettre signifient le nombre d'apparitions.

**Exercice N° 20**

Ecrire un programme Pascal permettant de saisir n entiers pairs puis d'afficher leur moyenne. (n étant un entier de l'intervalle [15, 30]).

**Exercice N° 21**

Ecrire un programme Pascal permettant de chercher puis d'afficher la plus grande valeur d'un tableau T contenant n entiers ( $5 \leq n \leq 20$ ) ainsi que son indice. Dans le cas d'ex aequo, on affiche l'indice de la première occurrence.

**Exercice N° 22**

On désire programmer la commande Rechercher et remplacer du menu Edition d'un logiciel de traitement de texte, qui, en fournissant le texte, un mot1 et un mot2, permet de remplacer toute occurrence de mot1 par mot2 dans texte.

**Exemple :**

Soit le texte suivant : "Les structures conditionnelles sont simples à comprendre. L'emploi des structures conditionnelles rend la programmation plus intéressante."

Soit mot1 = "conditionnelles"

Soit mot2 = "itératives"

Le programme doit afficher : "Les structures itératives sont simples à comprendre. L'emploi des structures itératives rend la programmation plus intéressante."

**Exercice N° 23**

Ecrire un programme qui permet de calculer et d'afficher le PGCD de deux entiers non nuls

**Exercice N° 24**

Ecrire un programme qui permet de saisir une chaîne de chiffres CH, de chercher la combinaison maximale CMAX et la combinaison minimale CMIN qu'on peut obtenir à partir des chiffres de CH, et enfin de les afficher.

Pour déduire CMAX à partir de CH, on vous propose les étapes suivantes :

Chercher le plus grand chiffre dans CH

Le placer dans la chaîne CMAX et CMIN

Remplacer le chiffre qui était le plus grand par le caractère "\*"

**Exercice N° 25**

Un CODEC est un logiciel compresseur décompresseur de fichiers. En effet, les suite de bits composant un fichier comportent des similitudes comme 10001111. Plutôt que de stocker la totalité de cet octet, on gagne de la place en écrivant 14031 (qui se lit un quatre zéros trois un). Cet octet (huit bits) retrouva ensuite son format original à la décompression.

Il s'agit alors de saisir une chaîne de huit chiffres formée uniquement 0 et 1 pour désigner un octet puis la compresser suivant le principe de compression du CODEC et enfin l'afficher.

**Exemple :** Si octet = "10010111" Alors l'octet compressé est : "1201031"

**Exercice N° 26**

On veut écrire un programme Pascal permettant de coder un message selon le procédé suivant : permuter chaque caractère d'indice pair avec le caractère qui le précède.

**Exemple :** Le codage de la chaîne de caractère : "Baccalauréat" donne "aBcclauaérta"

**Exercice N° 27**

Ecrire un programme qui permet de remplir un tableau T par n caractère ( avec  $6 \leq n \leq 30$ ). Et de répartir ces n caractères sur trois tableaux et les afficher :

**TL :** un tableau de lettres

**TC :** un tableau de chiffres

**TS :** un tableau de symboles

**Exemple :**

Soit n = 10

T 

--	--	--	--	--	--	--	--	--	--

On doit obtenir les tableaux suivants :

TL 

H	K	R	d
---	---	---	---

TC 

4	2	6
---	---	---

TS 

!	}	\$
---	---	----

**Exercice N° 28**

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  $60 = 2 * 2 * 3 * 5$

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 29**

On veut écrire un programme Pascal permettant de supprimer les espaces superflus dans une chaîne de caractère, contient au moins un espace.

**Exemple :**

Si la chaîne est "  Travaux    pratique  "

Alors l'exécution du programme donnera la chaîne = "Travaux \_ pratique".

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 30**

Ecrire un programme Pascal qui saisit une phrase et l'affiche renversée. La phrase commence, obligatoirement, par une lettre et ses mots sont séparés par un seul espace et ne se termine pas par un espace.

**Exemple :**

Votre phrase :

"RESOLUTION DE PROBLEMES"

Résultat :

"PROBLEMES DE RESOLUTION"

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 31**

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
----	---	---	---	---	---	---	---	---	---

Pour regrouper 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 et détermine et affiche le nombre d'éléments différents de T.

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 32**

Soit un tableau T de 20 entiers positifs. Ecrire un programme Pascal qui permet d'afficher les éléments de T compris entre deux positions P1 et P2, leur moyenne arithmétique, la valeur maximale et la valeur minimale contenues dans cet intervalle. On donne  $1 \leq P1 < P2 \leq 20$ .

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 33**

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.

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

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 34**

Soit T un tableau de N éléments ( $2 < N < 200$ ) de type caractère. On désire écrire un programme Pascal permettant de vérifier l'existence dans le tableau T d'un certain nombre de mots saisis dans un tableau Tm de P éléments ( $2 < P < 20$ ).

**Exemple :**

Tm

BAC	Canne	Sujet
-----	-------	-------

T

L	B	S	u	j	e	t	a	B	A	C	a	n	n	e	d
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Remarques :**

1. Les caractères de la chaîne recherchée doivent être adjacents dans le tableau T et non dispersés.
2. on remarque que les mots BAC, Canne et Sujet figurent dans le tableau T.

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 35**

Soit un tableau T1 contenant n lettres majuscules (de A à Z). n étant un entier compris entre 5 et 20.  
On désire trier en ordre croissant les éléments de T1 et les ranger dans un tableau T2 en utilisant le principe suivant :

1. Chercher la lettre qui a le plus petit code ASCII dans T1
2.       a) Ranger cette lettre dans T2  
       b) Remplacer cette lettre par "\*" dans T1
3. Répéter n fois les étapes 1 et 2.

Ecrire un programme Pascal qui permet de :

- Saisir les éléments de T1.
- Trier les éléments de T1 et les ranger dans T2.
- Afficher les éléments de T2.

**N.B :** La solution doit comporter au moins une fonction et deux procédures.

**Exercice N° 36**

Ecrire un programme Pascal qui permet de trier par ordre décroissant les éléments d'un tableau A de n entiers positifs dans un nouveau tableau B de même dimension.

N étant un entier vérifiant  $5 < n < 25$ .

On utilisera la démarche suivante :

1. chercher le maximum de A
2. placer ce maximum dans B
3. remplacer le maximum par -1 dans A
4. refaire les étapes 1, 2 et 3 jusqu'à ce que le tableau A soit entièrement composé de -1.

**N.B :**

- 1) Prévoir l'affichage des éléments du tableau B
- 2) La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 37**

Deux entiers naturels strictement positifs m et n sont dits nombre 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.

**Exemple :**

220 et 284 sont deux nombres amis. En effet :

$$D_{284} = \{1, 2, 4, 71, 142, 284\}$$

$$D_{220} = \{1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, 220\}$$

D284 et D220 sont respectivement les ensembles de tous les diviseurs de 284 et de 220.

$$284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110$$

$$220 = 1 + 2 + 4 + 71 + 142$$

Écrire un programme Pascal qui permet de déterminer puis d'afficher si deux entiers naturels donnés m et n sont amis ou non.

**N.B :** La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 38**

Ecrire un programme Pascal qui réalise le traitement suivant :

- choisir un entier n de l'intervalle [100, 500] et un entier m de l'intervalle [10,99]
- afficher tous les entiers de l'intervalle [1, m] en remplaçant par le caractère "\*" tous les diviseurs de n ainsi que tous les entiers comportant dans leurs écritures le chiffre des unités de n.

**Exemple d'exécution :**

Si n = 100 et m = 20 alors la liste suivante sera affichée :

\* \* 3 \* \* 6 7 8 9 \* 11 12 13 14 15 16 17 18 19 \*

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 39**

On veut écrire un programme permettant de remplir deux tableaux **P** et **Q** de taille maximales **n** ( $n \leq 25$ ) par des réels. Puis de ranger respectivement les éléments positifs de **P** puis ceux de **Q** dans un tableau **TPOS**. Ensuite calculer puis afficher la moyenne arithmétique des éléments de **TPOS**.

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 40**

On appelle moyenne olympique d'un ensemble de nombres la moyenne arithmétique de tous les nombres de cet ensemble sauf le plus petit et le plus grand.

Ecrire un programme Pascal permettant de saisir un tableau de **N** réels ( $5 \leq N \leq 20$ ) distincts et d'afficher leur moyenne olympique.

**N.B :** La solution doit comporter au moins trois modules.

**Exercice N° 41**

Soit l'algorithme suivant :

- 0) **Début Exercice**
- 1) [Lire (**n**)] **Pour** k de 1 à n **répéter**  
     Lire (T[k])  
     **FinPour**
- 2) Lire (v)
- 3) [trv ← faux, i ← 0] **répéter**  
     i ← i + 1  
     trv ← (T[i] = v)  
     **Jusqu'à** (i = n) ou (trv)
- 4) **Si** (trv) **Alors**  
     rt ← " est dans T "  
     **Sinon** rt ← " n'est pas dans T "  
     **FinSi**
- 5) Ecrire (v, rt)
- 6) **Fin Exercice**

**Questions :**

1. Traduire cet algorithme en Pascal.
2. Que fait cet algorithme ? (Ecrire la réponse comme commentaire à la fin du programme).
3. Ajouter une fonction intitulée **OCCURRENCES** qui permet de chercher le nombre d'occurrences de la valeur **v** dans **T**.
4. Modifier le programme principal pour qu'il affiche le nombre d'occurrences de la valeur **v** dans **T**, si elle existe au moins une seule fois.
5. Ajouter les messages nécessaires avant la lecture de toutes les données
6. Enregistrer votre travail dans C:\BAC2007 qu'a comme nom le numéro de votre carte d'identité national (NCIN)

**Exercice N° 42**

On se propose d'écrire un programme Pascal permettant de déterminer et d'afficher la lettre alphabétique la plus utilisée dans un texte donné. Le texte étant saisi comme une chaîne de caractères contenant **n** de caractères ( $5 \leq n \leq 20$ ).

Dans le cas d'ex aequo afficher toutes les lettres ayant la plus grande fréquence.

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N°43**

On veut écrire un programme Pascal permettant de lire un mot intitulé **CHM** composé au moins de 5 caractères et d'afficher les chaînes de caractères suivantes :

- La chaîne formée par le premier et le dernier caractère de **CHM**
- La chaîne formée par les deux premiers et les deux derniers caractères de **CHM**
- Etc.

**Exemple :** Si la chaîne **CHM** contient "**TURBO**" alors le programme affichera :

TO  
TUBO  
TURRBO  
TURBURBO  
TURBOTURBO

**N.B:** La solution doit comporter au moins deux modules.

**Exercice N° 44**

Deux joueurs lancent en même temps un dé dont les faces sont numérotées de 1 à 6. Le joueur qui obtiendra la plus grande valeur aura un point. Le jeu s'arrête quand l'un des joueurs arrive le premier à un score de 10 points.

Ecrire un programme Pascal simulant ce jeu et afficher le numéro du joueur gagnant.  
On pourra utiliser la fonction prédéfinie **RANDOM (n)** qui retourne un entier de l'intervalle **[0, n - 1]**.

**N.B :**

La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 45**

Soit T un tableau de n entiers ( $2 \leq n \leq 20$ ). On veut écrire un programme Pascal qui saisit n et T puis affiche la valeur maximale V\_max et la valeur minimale V\_min d'une séquence de T allant de i à j. (i et j sont deux entiers donnés vérifiant la condition  $i \leq j \leq n$ ).

**N.B :**

La solution doit comporter au moins deux modules.

**Exercice N° 46**

Ecrire un programme Pascal qui saisit une chaîne de caractères de longueur minimal 3 et l'affiche sous la forme d'un triangle comme indiqué ci-dessous.

**Exemple :**

Si la chaîne saisie est "**INTERNET**", on aura :

```
I
IN
INT
INTE
INTER
INTERN
INTERNE
INTERNET
```

**N.B :**

La solution doit comporter au moins deux modules.

**Exercice N° 47**

Soit T un tableau de N caractères alphabétique ( $2 < N < 20$ )  
Ecrire un programme Pascal permettant de crypter les données figurant dans le tableau T comme suit :

1. Convertir chaque caractère en sa représentation en code ASCII.
2. Permuter les chiffres des unités avec ceux des dizaines.
3. Insérer le caractère correspondant à ce nouveau code ASCII dans un tableau R.

Puis afficher le tableau R obtenu.

**Exemple :**

Si N = 3 et T =

	"B"	"A"	"C"
A	"8"	"L"	

Alors R =

"B"	"8"	"L"
-----	-----	-----

Ord ("B") = 66, si on permute le chiffre de unité avec celui de dizaine on obtient le code ASCII du caractère "B"  
Ord ("A") = 65, si on permute, on obtient (56) le code ASCII du caractère "8"  
Ord ("C") = 67, si on permute, on obtient (76) le code ASCII du caractère "L"

**N.B :**

La solution doit comporter au moins une fonction et une procédure.



**Exercice N° 48**

Ecrire un programme Pascal qui permet de saisir un entier positif N, composé de trois chiffres, de déterminer et d'afficher tous les nombres qui peuvent être formés par les chiffres de N, ainsi que le plus petit et le plus grand de ces nombres

**Exemple :**

Pour N = 427, le programme affichera :

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

Le plus petit nombre est 247

Le plus grand nombre est 742

**N.B :**

La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 49**

L'algorithme suivant est celui d'une fonction permettant de calculer la somme d'une partie d'éléments d'un tableau T de n entiers, délimité par les indices p1 et p2.

- 0) Def FN somme (T : TAB ; p1,p2 : entier) : entier ;
- 1) [s ← 0] Pour i de p1 à p2 faire  
 $s \leftarrow s + T[i]$   
 Fin Pour
- 2) somme ← s
- 3) Fin somme

En exploitant la fonction dont l'algorithme est ci-dessus, Ecrire un programme Pascal qui permet de :

- Saisir un tableau V de N entiers strictement positifs ( $5 \leq n \leq 20$ ).
- Afficher l'indice (ind) de l'élément du tableau dont l'écart entre la somme (s1) des éléments qui le précèdent et celle des éléments qui le succèdent (s2) soit minimal
- Afficher les sommes s1 et s2 correspondantes

**Exemple :**

Pour le tableau T suivant :

**V**

11	3	9	24	30	7	4	14	16	21	13	16
1	2	3	4	5	6	7	8	9	10	11	12

Le programme affiche : S1 = 84, S2 = 80 et ind = 7

**N.B :**

La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 50**

Ecrire un programme qui permet de déterminer et d'afficher tous les diviseurs suivis de tous les multiples d'un entier p donné, dans une partie d'un tableau T de n entiers donnés. Cette partie est délimitée par deux indices ind\_inf et ind\_sup. Avec ( $0 < \text{ind\_inf} < \text{ind\_sup} \leq n \leq 15$ )

**Exemple :**

**T**

25	32	43	4	32	72	80	15	24	2	48	56	10	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14

↑ ind\_inf
↑ ind\_sup

Pour n = 14, p = 8, ind\_inf = 3 et ind\_sup = 11, le programme affichera :

Les diviseurs de 8 sont : 4 2

Les multiples de 8 sont : 32 72 80 24 48

**N.B :**

La solution doit comporter au moins une fonction et une procédure.

**Exercice N° 51**

Ecrire un programme qui saisit un entier naturel suivi de  $n$  réels à placer dans un tableau A (avec  $2 \leq n \leq 25$ ). Ce programme mettra les éléments de A dans un deuxième tableau B avec une rotation de  $k$  éléments,  $k$  étant un entier donné et tel que  $0 < k < n$ .

**Exemples :**Si  $n = 5$ , A =

2.2	6	-1.5	30.12	8
-----	---	------	-------	---

Et  $k = 1$  Alors B =

6	-1.5	30.12	8	2.2
---	------	-------	---	-----

Mais si  $k = 2$  Alors B =

-1.5	30.12	8	2.2	6
------	-------	---	-----	---

Et si  $k = 4$  alors B =

8	2.2	6	-1.5	30.12
---	-----	---	------	-------

**N.B :** La solution doit comporter au moins deux modules.

**Exercice N° 01**

```

Program CONVERTIR;
Uses WinCrt;
Var
temps,h,min,s:integer;
Begin
Writeln('Temps en seconde'); Readln(temps);
h:= temps Div 3600;
min:= (temps Mod 3600) Div 60;
s:= temps Mod 60;
Writeln(temps,' s ',' = ',h,' h : ',min,' min : ',s,' s');
End.

```

**Exercice N° 02**

```

Program inconnu;
Uses WinCrt;
Var
c1,c2:char;
Begin
Writeln('Tapez un caractère :'); Readln(c1);
If (( ORD(c1) >= 97) and (ORD(c1) <= 122)) Then
    c2 := CHR(ORD(c1) - 32)
Else    c2 := c1;
Writeln(c2);
End.

```

*{Ce Programme permet d'afficher un caractère (c1) Donnée en majuscule}*

**Exercice N° 03**

```

Program SUCC_PRED;
Uses WinCrt;
Var
c,s,p:char;
Begin
Writeln('Tapez un caractère');
Readln(c);
s:= succ(c); p:= pred(c);
Writeln('Le successeur de "',c,'" est "',s,'"');
Writeln('Le prédécesseur de "',c,'" est "',p,'"');
End.

```

**Exercice N° 04**

```

Program PERMUTATION_CIRCULAIRE;
Uses WinCrt;
Var
C1,C2,C3,aux:char;
Begin
Writeln('Tapez trois caractère');
Readln(C1,C2,C3);
aux:= C1; C1:= C3; C3:= C2; C2:= aux;
Writeln(C1,C2,2,C3:2);

```

**End.****Exercice N° 05**

```

Program EXTRAIRE;
Uses WinCrt;
Var
x,c,d,u:integer;
Begin
Writeln('Donnez un entier composé de trois chiffres');
Readln(x);
c:= x Div 100;
d:= (x Mod 100) Div 10;
u:= x Mod 10;
Writeln('Centaine = ',c,' Dizaine = ',d,' Unité = ',u);
End.

```

**Exercice N° 06**

```

Program DISTANCE;
Uses WinCrt;
Var
a,b,c,d,r:real;
Begin
Writeln('Donnez les coordonnées du premier point (M)');
Readln(a,b);
Writeln('Donnez les coordonnées du deuxième point (N)');
Readln(c,d);
r:= sqrt(sqrt(a - c) + sqrt(b - d));
Writeln('d (M, N) = ',r:7:3);
End.

```

**Exercice N° 07**

```

Program CAPACITE;
Uses WinCrt;
Var
bits:integer;
o,Ko,Mo,Go:real;
Begin
Writeln('Capacité en bits'); Readln(bits);
o:= bits / 8;
Ko:= o / 1024;
Mo:= Ko / 1024;
Go:= Mo / 1024;
Writeln(bits,' bits = ',o:7:3,' o = ',Ko:7:4,' Ko = ',Mo:7:5,' Mo = ',Go:8:6,' Go');
End.

```

**Exercice N° 08**

```

Program RESISTANCE;
Uses WinCrt;
Var R1,R2,R3,R4,RE:real;
Begin
Writeln('Donnez les valeurs des résistances R1, R2, R3 et R4');
Readln(R1, R2, R3, R4);

```

```

RE:= ((R1 * R2) / (R1 + R2)) + R3 + R4;
Writeln('La résistance équivalente RE du circuit = ',RE:7:3);
End.

```

**Exercice N° 09**

```

Program POSITION2;
Uses WinCrt;
Var c:char; ch:string; p:integer;
Begin
Writeln('Tapez un caractère'); Readln(c);
Writeln('Tapez une chaîne de caractère'); Readln(ch);
p:= pos(c,ch);
p:= p + pos(c, copy(ch,p + 1, length(ch) - p));
Writeln('La deuxième position de "',c,'" dans "',ch,'" = ',p);
End.

```

**Exercice N° 10**

```

Program POSITION2;
Uses WinCrt;
Var c:string[1]; ch:string; chIf,n,p:integer;
Begin
Writeln('Tapez un chiffre'); Readln(chIf);
Writeln('Tapez un entier'); Readln(n);
str(chIf,c); str(n,ch); p:= pos(c,ch);
p:= p + pos(c, copy(ch,p + 1, length(ch) - p));
Writeln('La deuxième position de ',c,' dans ',n,' = ',p);
End.

```

**Exercice N° 11**

```

Program ETAT_PERSONNE;
Uses WinCrt;
Var
sexe:char; taille,poids,PI,BMI:real;
Begin
Writeln('Donnez le sexe de la personne (M/F)'); Readln(sexe);
Writeln('Donnez la taille de la personne en Cm');
Readln(taille);
Writeln('Donnez le poids de la personne en Kg');
Readln(poids);
If Upcase(sexe) = 'M' Then
    PI:= (taille - 100) - (taille - 150) / 4
Else
    PI:= (taille - 100) - (taille - 120) / 4;
Writeln('PI = ',PI:5:2);
BMI:= poids / sqrt(taille / 100);
Writeln('BMI = ',BMI:5:2);
If BMI <= 27 Then
    Writeln('La personne est Normale')
Else If (BMI > 27) and (BMI < 32) Then
    Writeln('La personne est Obèse')
Else
    Writeln('La personne est Malade');
End.

```

**Exercice N° 12**

```

Program SAISON;
Uses WinCrt;
Var mois:integer; r:string[9];
Begin
  Writeln('Donnez le numéro d'un mois'); Readln(mois);
Case mois of
    3..5:r:='Printemps';
    6..8:r:='Eté';
    9..11:r:='Automne';
    12,1,2:r:='Hiver';
  End;
  Writeln('Saison = ',r);
End.

```

**Exercice N° 13**

```

Program NOMBRE_JOURS;
Uses WinCrt;
Var
  mois,nbj,annee:integer;
Begin
  Writeln('Donner le mois (N°)');
  Readln(mois);
  Case mois of
    1,3,5,7,8,10,12:nbj:=31;
    4,6,9,11:nbj:=30;
    2:
      Begin
        Writeln('Donnez l'année');
        Readln(annee);
        If annee Mod 4 = 0 Then
          nbj:=29
        Else
          nbj:=28;
      End;
  End;
  Writeln('Le nombre de jours du mois ',mois,' est de ',nbj,' jours');
End.

```

**Exercice N° 14**

```

Program CALCULATRICE;
Uses WinCrt;
Var
  a,b,r:real; op:char;
Begin
  Writeln('Tapez la première valeur');
  Readln(a);
  Writeln('Donnez l'opérateur');
  Readln(op);
  Writeln('Tapez la deuxième valeur');
  Readln(b);

```

**Case op of**

```

  '+':r:=a + b;
  '-':r:=a - b;
  '*':r:=a * b;
  '/':r:=a / b;
End;
  Writeln(a:5:2,' ',op,' ',b:5:2,' = ',r:5:2);
End.
Exercice N° 15
Program EXISTANCE;
Uses WinCrt;
Type TAB = Array[1..99] of real;
Var
  T:TAB; n,i,occ:integer; r:real;
Begin
  Repeat
    Writeln('Donnez la taille du tableau T'); Readln(n);
  Until (n in [1..99]);
  For i := 1 To n Do
    Begin
      Writeln('Donnez la valeur de l'élément N° ',i);
      Readln(T[i]);
    End;
    Writeln('Tapez la valeur recherchée'); Readln(r);
    occ:=0;
  For i := 1 To n Do
    If T[i] = r Then occ := occ + 1;
  Writeln(r:5:2,' existe ',occ,' fois dans T');
End.

```

**Exercice N° 16**

```

Program FACTORIELLE;
Uses WinCrt;
Var f,i,nb:integer;
Begin
  Repeat
    Writeln('Donnez un entier positif'); Readln(nb);
  Until (nb >= 0);
  f:=1;
  For i := 2 To nb Do
    f:= f * i;
  Writeln(nb,'! = ',f);
End.

```

**Exercice N° 17**

```

Program PARFAIT;
Uses WinCrt;
Var
  nb,s,x,i:integer;
Begin
  Writeln('Les 4 premiers nombres parfaits sont:');
  nb:= 0; x:= 6;
  Repeat

```

```

  s:=1;
  For i := 2 To x Div 2 Do
    If x Mod i = 0 Then s:=s + i;
  If x = s Then Begin
    Writeln(x); nb:=nb + 1;
  End;
  x:=x + 1;
Until(nb = 4);
End.

```

**Exercice N° 18**

```

Program RECH_CUBIQUE;
Uses WinCrt;
Var
  x,s,i,c,d,u:integer;
Begin
  For x := 100 To 999 Do
    Begin
      c:= x Div 100;
      d:= (x Mod 100) Div 10;
      u:= x Mod 10;
      s:= (c*c*c) + (d*d*d) + (u*u*u);
      If x = s Then Writeln(x);
    End;
  End.

```

**Exercice N° 19**

```

Program OCCURENCES;
Uses WinCrt;
Type
  TAB = Array['A'..'Z'] of integer;
Var
  OCC:TAB;
  i:integer;ch:string; j:char;
Begin
  Writeln('Tapez votre texte'); Readln(ch);
  For i:= 1 To length(ch) Do
    OCC[uppercase(ch[i])]:=OCC[uppercase(ch[i])] + 1;
  For j:= 'A' To 'Z' Do
    If OCC[j] <> 0 Then Writeln(j,' existe ',OCC[j],' Dans ',ch);
  End.

```

**Exercice N° 20**

```

Program moyenne;
Uses WinCrt;
Type
  TAB = Array[1..30] of integer;
Var
  n,i:integer; moy:real; T:TAB;
Begin
  Repeat
    Writeln('Donnez le nombre des éléments du Tableau');
    Readln(n);

```

```

Until (n in [15..30]);
For i := 1 To n Do
  Repeat
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln(T[i]);
  Until (T[i] Mod 2 = 0);
For i := 1 To n Do
  moy := moy + T[i];
moy := moy / n;
Writeln('La moyenne du tableau = ',moy:7:3);
End.
Exercice N° 21
Program plus_grande;
Uses WinCrt;
Type
TAB = Array[1..20] of integer;
Var
T:TAB;
i,n,ind,max:integer;
Begin
Repeat
  Writeln('Donnez le nombre des éléments du Tableau');
  Readln(n);
Until (n in [5..20]);
For i := 1 To n Do
  Begin
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln(T[i]);
  End;
max := T[1];
ind := 1;
For i := 2 To n Do
  If T[i] > max Then
  Begin
    max := T[i];
    ind := i;
  End;
Writeln(max, ' C'est la plus grande valeur du tableau d'indice ',ind);
End.

```

```

Exercice N° 22
Program rechercher_remplacer;
Uses WinCrt;
Var
texte,mot1,mot2:string;
p,l1:integer;
Begin
Writeln('Tapez votre texte');
Readln(texte);
Writeln('Tapez le mot rechercher');
Readln(mot1);
Writeln('Tapez le mot à remplacer');

```

```

Readln(mot2);
p:=pos(mot1,texte);
l1:=length(mot1);
While p <> 0 Do
  Begin
    Delete(texte,p,l1);
    Insert(mot2,texte,p);
    p:=pos(mot1,texte);
  End;
Writeln('Texte après modification: ',texte);
End.

```

```

Exercice N° 23
Program PGCD;
Uses WinCrt;
Var a,b:integer;
Begin
Repeat
  Writeln('Donnez deux entiers non nuls'); 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 N° 24
Program cmax_cmin;
Uses WinCrt;
Var ch,cmax,cmin:string; ok:boolean; i,l,j,ind_max:integer;
Begin
Repeat
  Writeln('Tapez une chaîne de chiffres'); Readln(ch);
  ok := true; i:= 1; l:=length(ch);
  Repeat
    If (not (ch[i] in ['0'..'9'])) Then ok := false;
    i := i + 1;
  Until((not ok) or (i > l));
Until (ok);
cmin:=ch; cmax:=ch;
For j := 1 To l Do
  Begin
    ind_max:=1;
    For i:= 2 To l Do
      If ch[i] > ch[ind_max] Then ind_max := i;
      cmax[j] :=ch[ind_max];
      cmin[l-j+1]:=ch[ind_max]; ch[ind_max]:='*';
    End;
Writeln('La combinaison minimale ',cmin);
Writeln('La combinaison maximale ',cmax);
End.

```

```

Exercice N° 25
Program CODEC;

```

```

Uses WinCrt;
Var
octet,octet_com:string[8]; car:string[1]; i,l,occ:integer;
ok:boolean;
Begin
Repeat
  Writeln('Tapez la valeur de l'octet'); Readln(octet);
  ok := true; i:= 1; l:=length(octet);
  Repeat
    If (not (octet[i] in ['0','1'])) Then ok := false;
    i := i + 1;
  Until ((not ok) or (i > 8) and (l = 8));
Until (ok);
i:=1;
Repeat
  occ:=1; ok:=true;
  Repeat
    If octet[i] = octet[i+1] Then
      Begin
        occ := occ + 1; i := i + 1;
      End Else ok := false;
  Until ((not ok) or (i > 8));
  If occ = 1 Then
    octet_com:= octet_com + octet[i]
  Else Begin
    str(occ,car);
    octet_com:= octet_com + car + octet[i];
  End;
  i:=i+1;
Until (i > 8);
writeln('Si octet = "',octet,'" Alors l'octet compressé est: "',octet_com,'"');
End.
Exercice N° 26
Program CODER;
Uses WinCrt;
Var ch,ch_codee:string; l,i:integer;
Begin
Repeat
  Writeln('Tapez une chaîne'); Readln(ch);
  l:=length(ch);
  Until (l in [3..255]);
  ch_codee:=ch; i:=2;
  Repeat
    ch_codee[i-1]:=ch[i];
    ch_codee[i]:=ch[i-1];
    i:=i+2;
  Until (i > l);
  Writeln('Le codage de la chaîne de caractère: "',ch,'" Donne "',ch_codee,'"');
End.
Exercice N° 27

```

```

Program REPARTIR;
Uses WinCrt;
Type
TAB = Array[1..30] of char;
Var
n,i,PTL,PTC,PTS:integer;
T,TL,TC,TS:TAB;
Begin

Repeat
  Writeln('Donnez la taille du tableau');
  Readln(n);
Until (n in [6..30]);

For i:= 1 To n Do
  Begin
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln(T[i]);
  End;

PTL:=1; PTC:=1; PTS:=1;
For i := 1 To n Do
  If Ucase(T[i]) in ['A'..'Z'] Then
    Begin
      TL[PTL]:=T[i]; PTL:=PTL + 1;
    End
    Else If T[i] in ['0'..'9'] Then
      Begin
        TC[PTC]:=T[i]; PTC:=PTC + 1;
      End
    Else Begin
      TS[PTS]:=T[i]; PTS:=PTS + 1;
    End;

Writeln('Tableau des lettres');
For i := 1 To PTL - 1 Do Write(TL[i], ' ');
Writeln; Writeln('Tableau des chiffres');
For i:= 1 To PTC - 1 Do Write(TC[i], ' ');
Writeln; Writeln('Tableau des symboles');
For i:= 1 To PTS - 1 Do Write (TS[i], ' ');
End.

```

**Exercice N° 28**

```

Program FACTEURS_PREMIERS;
Uses WinCrt;
Type TAB = Array[1..6] of integer;
Var n,r,nb:integer; T:TAB;
Procedure lire (Var x:integer);
Begin
Repeat
  Writeln('Donnez un entier compris entre 2 et 100');
  Readln(x);
Until (x in [2..100]);
End;

```

```

Procedure decomposer(x:integer; Var T:TAB; Var
y:integer);
Var r,q,d:integer;
Begin
d:= 2; y:= 0;
Repeat
  r:= x Mod d; q:= x Div d;
  If r = 0 Then
    Begin
      y:= y + 1; T[y]:= d; x:= q;
    End Else d:= d + 1;
Until(q = 0);
End;
Procedure afficher (n,y:integer;T:TAB);
Var i:integer;
Begin
Write(n, ' = ');
If y <> 1 Then
  Begin
    For i := 1 To y - 1 Do Write(T[i], ' * ');
    Writeln(T[i+1]);
  End Else Writeln(T[1]);
End;
Begin
lire(n); decomposer(n,T,nb); afficher(n,nb,T);
End.

```

**Exercice N° 29**

```

Program SUPPRIMER;
Uses WinCrt;
Var chaine:string;
Procedure lire (Var ch:string);
Begin
Repeat
  Writeln('Tapez une chaîne non vide'); Readln(ch);
Until (pos(' ',ch) > 0);
End;
Procedure superflus (Var ch:string);
Var l,p,ind:integer;
Begin
p:=pos(' ',ch); l:=length(ch);
While p <> 0 Do
  Begin
    If (p=1) or (l=p) or (ch[p+1]=' ') Then
      delete(ch,p,1);
    If (p<>1) and (l<>p) and (ch[p+1] <> ' ') Then
      Begin ch[p]:='*'; ind:=p; End;
    p:= pos(' ',ch);
  End;
  ch[ind]:=' ';
End;
Begin
lire(chaine);

```

```

superflus(chaine);
Writeln('Votre chaîne Après suppression des espaces superflus
=');
Writeln(chaine);
End.

```

**Exercice N° 30**

```

Program RENVERSEE;
Uses WinCrt;
Var phrase:string;

Procedure lire (Var ch:string);
Begin
Repeat
  Writeln('Tapez votre phrase');
  Readln(ch);
Until ((upcase(ch[1]) in ['A'..'Z']) and (pos(' ',ch)= 0) and
(ch[length(ch)] <> ' '));
End;

```

**Function** renverser (ch:**string**):**string**;

```

Var
p:integer;
ch1:string;
Begin
While (pos(' ',ch) <> 0) Do
  Begin
    p:= pos(' ',ch);
    ch1:= copy(ch,1,p-1)+ ' ' + ch1;
    delete(ch,1,p);
  End;
ch1:= ch + ' ' + ch1;
renverser:=ch1;
End;

```

```

Begin
lire (phrase);
Writeln(renverser(phrase));
End.

```

**Exercice N° 31**

```

Program ELEMENTS_DIFFERENTS;
Uses WinCrt;
Type
TAB = Array[1..20] of integer;
VAR T:TAB; n:integer;

Procedure lire_entier(Var x:integer);
Begin
Repeat
  Writeln('Donnez le nombre des éléments du tableau');
  Readln(x);
Until (x in [2..20]);

```

```

End;

Procedure lire_tableau (Var vec:TAB;y:integer);
Var i:integer;
Begin
For i := 1 To y Do
Begin
  Repeat
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln(vec[i]);
  Until (vec[i] > 0);
End;
End;

Procedure garder (Var vec:TAB;y:integer);
Var i,j:integer;
Begin
For i:=1 To y-1 Do
  For j:= i+1 To y Do
    If vec[i] = vec[j] Then vec[j]:=0;
End;
Procedure regrouper (Var vec:TAB;Var y:integer);
Var vec1:TAB; ind1,ind2,i:integer;
Begin
vec1[1]:=vec[1]; ind1:=1; ind2:=y;
For i:= 2 To y Do
  If vec[i] > 0 Then
    Begin
      ind1:= ind1 + 1; vec1[ind1]:=vec[i];
    End
  Else Begin
    vec1[ind2]:=vec[i]; ind2:= ind2 - 1;
  End;
y:=ind1; vec:=vec1;
End;
Procedure afficher(vec:TAB;y:integer);
Var i:integer;
Begin
For i:= 1 To y Do
  Writeln(vec[i]);
End;
Begin
lire_entier(n);
lire_tableau(T,n);
garder(T,n);
regrouper(T,n);
Writeln('Il y a ',n,' élément(s) dIfférent(s) dans votre tableau
qui sont:');
afficher(T,n);
End.

```

**Exercice N° 32**  
**Program** POSITION;

```

Uses WinCrt;
Type TAB = Array[1..20] of integer;
Var T:TAB; P1,P2:integer;
Procedure remplir(Var vec:TAB);
Var
i:integer;
Begin
For i:=1 To 20 Do
Begin
  Repeat
    Writeln('Donnez la valeur de l'éléments N° ',i);
    Readln(vec[i]);
  Until(vec[i] > 0);
End;
End;
Procedure lire(Var x,y:integer);
Begin
Repeat
  Writeln('Tapez les deux positions');
  Readln(x,y);
Until ((x in [1..y-1]) and (y <= 20));
End;
Procedure afficher (vec:TAB;x,y:integer);
Var moy:real; min,max,i:integer;
Begin
min:=vec[x];
max:=vec[x];
moy:=0;
Writeln('Les éléments compris entre ',x,' et ',y,' sont');
For i:= x To y Do
Begin
  Writeln(vec[i]);
  moy:= moy + vec[i];
  If min > vec[i] Then min:=vec[i];
  If max < vec[i] Then max:=vec[i];
End;
moy:=(moy / (y-x));
Writeln('Leur moyenne arithmétique = ',moy:6:2);
Writeln('La valeur maximale = ',max);
Writeln('La valeur minimale = ',min);
End;
Begin
remplir(T);
lire(P1,P2);
afficher(T,P1,P2);
End.

```

**Exercice N° 33**  
**Program** ANAGRAMME;  
**Uses** WinCrt;  
**Var** MOT1,MOT2:**string**;  
**Procedure** lire(**Var** ch1,ch2:**string**);  
**Begin**

```

Repeat
  Writeln('Tapez votre premier mot'); Readln(ch1);
  Writeln('Tapez votre deuxième mot'); Readln(ch2);
Until ((ch1 <> '') and (ch2 <> '') and (ch1 <> ch2));
End;
Function verIfier (ch2,ch1 : string):boolean;
Var ok:boolean; i,l,p:integer; ch3:string;
Begin
l :=length(ch2); ok:= true;
If l <> length (ch1) then ok:= false
Else
Begin
  For i:=1 to l Do
    ch3:= ch3 + upcase(ch1[i]);
    i:= 1;
  Repeat
    p:= pos(upcase(ch2[i]),ch3);
    If p = 0 then ok:= false
    Else Begin i:= i + 1; ch3[p] := '*'; End;
  Until ((not ok) or (i > l));
End;
VerIfier := ok;
End;
Begin
lire (MOT1,MOT2);
If verIfier(MOT2,MOT1) then
  Writeln(MOT2,' est une anagramme de ',MOT1)
Else
  Writeln(MOT2,' n'est pas une anagramme de ',MOT1);
End.

```

**Exercice N° 34**  
**Program** EXISTANCE;  
**Uses** WinCrt;  
**Type**  
TAB1 = **Array**[1..199] **of** **char**;  
TAB2 = **Array**[1..19] **of** **string**;  
**Var**  
T:TAB1; Tm:TAB2;  
i,n,p:**integer**;  
**Procedure** lire\_n(**Var** x:**integer**);  
**Begin**  
**Repeat**  
 Writeln('Donnez la taille du tableau T'); Readln(x);  
**Until** (x in [3..199]);  
**End**;  
**Procedure** lire\_p(**Var** x:**integer**);  
**Begin**  
**Repeat**  
 Writeln('Donnez le nombre des mots');  
 Readln(x);  
**Until** (x in [3..19]);  
**End**;

```

Procedure remplir_T(Var vec:TAB1;y:integer);
Var i:integer;
Begin
For i := 1 To y Do
Begin
  Writeln('Tapez le caractère N° ',i);
  Readln(vec[i]);
End;
End;
Procedure remplir_Tm (Var vec:TAB2;y:integer);
Var i:integer;
Begin
For i:= 1 To y Do
Begin
  Writeln('Donnez le mot N° ',i);
  Readln(vec[i]);
End;
End;
Function verIfier (ch1,ch2:string):boolean;
Begin
If pos(ch1,ch2) = 0 Then
  verIfier:= false
Else verIfier:= true;
End;
Begin
lire_n(n);
remplir_T(T,n);
lire_p(p);
remplir_Tm(Tm,p);
For i:= 1 To p Do
Begin
  If verIfier (Tm[i],T) Then
    Writeln(Tm[i], ' existe dans ',T)
  Else
    Writeln(Tm[i], ' n'existe pas dans ',T);
End;
End.

```

```

Exercice N° 35
Program TRIER;
Uses WinCrt;
Type TAB = Array [1..20] of char;
Var
T1,T2: TAB;
n,i,p:integer;

```

```

Procedure lire (Var x:integer);
Begin
Repeat
  Writeln('Donnez la taille de votre Tableau');
  Readln(x);
Until (x in [5..20]);
End;

```

```

Procedure remplir(Var T:TAB;y:integer);
Var i:integer;
Begin
For i := 1 To y Do
Begin
  Repeat
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln(T[i]);
  Until (T[i] in ['A'..'Z']);
End;
End;
Function chercher(T:TAB;y:integer):integer;
Var ind,i:integer;
Begin
ind:= 1;
For i:= 2 To y Do
  If (T[i] <> '*') Then
    If (T[ind] = '*') or (T[ind] > T[i]) Then ind:= i;
chercher:= ind;
End;
Procedure afficher (T:TAB;y:integer);
Var i:integer;
Begin
ind:= 1;
For i:= 1 To y Do Writeln(T[i]);
End;
Begin
lire (n);
remplir(T1,n);
For i:= 1 To n Do
Begin
  p:= chercher(T1,n);
  T2[i]:= T1[p];
  T1[p]:= '*';
End;
afficher (T2,n);
End.
Exercice N° 36
Program TRIER;
Uses WinCrt;
Type TAB = Array[1..24] of integer;
Var A,B:TAB; n,i,p:integer;
Procedure lire (Var x:integer);
Begin
Repeat
  Writeln('Donnez la taille du tableau'); Readln(x);
Until (x in [6..24]);
End;
Procedure remplir(Var T:TAB;y:integer);
Var i:integer;
Begin
For i:= 1 To n Do
  Repeat

```

```

  Writeln('Donnez la valeur de l'élément N° ',i);
  Readln(T[i]);
  Until (T[i] > 0);
End;
Function maximum (T:TAB;y:integer):integer;
Var i,ind:integer;
Begin
ind:=1;
For i:= 2 To y Do
  If T[i] > T[ind] Then ind := i;
maximum := ind;
End;
Procedure afficher (T:TAB;y:integer);
Var i:integer;
Begin
For i:= 1 To y Do Writeln(T[i]);
End;
Begin
lire(n); remplir(A,n);
For i:= 1 To n Do
Begin
  p:= maximum(A,n);
  B[i]:=A[p];
  A[p]:= -1;
End;
Writeln('_____');
afficher(B,n);
End.

```

**Exercice N° 37**

```

Program AMIS;
Uses WinCrt;
Var m,n:integer;
Procedure lire(Var x:integer);
Begin
Repeat
  Writeln('Donnez un entier strictement positif'); Readln(x);
Until (x > 0);
End;
Function somme_Div (x:integer): integer;
Var i,s:integer;
Begin
s:=1;
For i:= 2 To x Do
  If (x Mod i) = 0 Then s:= s + i;
somme_Div := s;
End;
Begin
lire(m);
lire(n);
If ((somme_Div(m) - m) = n) and ((somme_Div(n) - n) = m) Then Writeln (m, ' et ',n, ' sont deux nombres amis')
Else Writeln (m, ' et ',n, ' ne sont pas deux nombres amis');

```



End.

**Exercice N° 38**

```

Program BOUM;
Uses WinCrt;
Var n,m:integer;
Procedure lire_n(Var x:integer);
Begin
Repeat
Writeln('Donnez un entier compris entre 100 et 500');
Readln(x);
Until ((x >= 100) and (x <= 500));
End;
Procedure lire_m(Var x:integer);
Begin
Repeat
Writeln('Donnez un entier compris entre 10 et 99');
Readln(x);
Until (x in [10..99]);
End;
Function comporte(n,x:integer):boolean;
Var u:integer; ch1,ch2:string;
Begin
u:= n Mod 10; str(u,ch1); str(x,ch2);
If pos(ch1,ch2) > 0 Then
comporte := true
Else
comporte := false;
End;
Procedure afficher (m,n:integer);
Var i:integer;
Begin
For i:= 1 To m Do
If (((n Mod i) = 0) or comporte(n,i)) Then Write('* ')
Else Write(i, ' ');
End;
Begin
lire_n(n);
lire_m(m);
afficher(m,n);
End.

```

**Exercice N° 39**

```

Program MOYENNE;
Uses WinCrt;
Type
TAB1 = Array[1..25] of real;
TAB2 = Array[1..50] of real;
Var P,Q:TAB1; TPOS:TAB2; n,nb:integer; moy:real;
Procedure lire (Var x:integer);
Begin
Repeat
Writeln('Donnez la taille maximale de deux tableaux');

```

```

Readln(x);
Until (x in [1..25]);
End;
Procedure remplir(Var T:TAB1;y:integer);
Var i:integer;
Begin
For i:=1 To y Do
Begin
Writeln('Donnez la valeur de l'élément N° ',i);
Readln(T[i]);
End;
End;
Procedure ranger (P,Q:TAB1;n:integer;Var TPOS:TAB2;Var
j:integer);
Var i:integer;
Begin
j:=0;
For i:=1 To n Do
If P[i] >= 0 Then
Begin j := j + 1; TPOS[j] := P[i]; End;
For i:= 1 To n Do
If Q[i] >= 0 Then
Begin j := j + 1; TPOS[j] := Q[i]; End;
End;
Function calculer (T:TAB2;nb:integer):real;
Var s:real; i:integer;
Begin
s:= T[1];
For i:=2 To nb Do s := s + T[i];
calculer := s / nb;
End;
Begin
lire(n);
remplir(P,n);
remplir(Q,n);
ranger(P,Q,n,TPOS,nb);
moy := calculer (TPOS,nb);
Writeln('La moyenne arithmétique des éléments positifs des
deux tableau = ',moy:7:3);
End.

```

**Exercice N° 40**

```

Program OLYMPIQUE;
Uses WinCrt;
Type TAB = Array [1..20] of real;
Var T:TAB; N:integer; moy:real;
Procedure lire (Var x:integer);
Begin
Repeat
Writeln('Donnez la taille du tableau');
Readln(x);
Until (x in [5..20]);
End;

```

```

Procedure remplir (Var V:TAB;y:integer);

```

```

Var
i:integer;
Begin
For i:= 1 To y Do
Begin
Writeln('Donnez la valeur de l'élément N° ',i);
Readln(V[i]);
End;
End;
Function moyenne_oly (V:TAB;y:integer):real;
Var
i:integer;
s,min,max:real;
Begin
s:= V[1];
min := s;
max := s;
For i:= 2 To y Do
Begin
s:= s + V[i];
If min > V[i] Then min := V[i]
Else If max < V[i] Then max := V[i];
End;
moyenne_oly:= (s - min - max) / (y - 2);
End;
Begin
lire(N);
remplir(T,N);
moy := moyenne_oly (T,N);
Writeln('La moyenne olympique de cet ensemble =
',moy:7:3);
End.
Exercice N° 41
Program Exercice;
Uses WinCrt;
Type TAB = Array [1..100] of integer;
Var n,k,i,v:integer; T:TAB; trv:boolean; rt:string;
Function OCCURRENCES(x:integer;V:TAB;y:integer) :
integer;
Var nb,i:integer;
Begin
nb:=0;
For i:= 1 To y Do
If V[i] = x Then nb := nb + 1;
OCCURRENCES := nb;
End;
Begin
Writeln('Donnez la taille de votre tableau');
Readln(n);
For k:= 1 To n Do
Begin
Writeln('Donnez la valeur de l'élément N° ',k);

```

```

    Readln(T[k]);
End;
Writeln("Tapez la valeur recherché");
Readln(v);
trv := false;
i := 0;
Repeat
    i := i + 1;
    trv := (T[i] = v);
Until ((i = n) or (trv));
If (trv) Then
    Begin
        str(OCCURRENCES(v,T,n),rt);
        rt := ' est dans T ' + rt + ' fois';
    End
    Else rt := ' n'est pas dans T';
Writeln(v,rt);
End.

```

{Ce Programme permet de saisir n entiers dans le tableau T, et d'afficher si un entier v existe ou pas dans ce tableau}

```

Exercice N° 42
Program FREQUENCE;
Uses WinCrt;
Type TAB = Array ['A'..'Z'] of integer;
Var n:integer; T:TAB; texte:string;
Procedure lire (Var x:integer);
Begin
Repeat
    Writeln('Donnez la taille de votre chaîne');
    Readln(x);
Until (x in [5..20]);
End;
Procedure lire_ch (Var ch:string;x:integer);
Begin
Repeat
    Writeln('Tapez votre chaîne'); Readln(ch);
Until (length(ch) = x);
End;
Procedure occurrences (ch:string;y:integer;Var V:TAB);
Var i:integer;
Begin
For i:= 1 To y Do
    If upcase(ch[i]) in ['A'..'Z'] Then
        V[upcase(ch[i])] := V[upcase(ch[i])] + 1;
End;
Function indice_max(V:TAB) : char;
Var max,i:char;
Begin
    i:='A'; max := i;

```

```

For i:= 'B' To 'Z' Do
    If V[i] > V[max] Then max := i;
    indice_max := max;
End;
Procedure afficher (V:TAB;ind:char);
Var i:char;
Begin
    Writeln('Le(s) lettre(s) alphabétique(s)le(s) plus utilisée(s):');
For i:=ind To 'Z' Do
    If V[ind] = V[i] Then Writeln(i);
End;
Begin
    lire (n);
    lire_ch (texte,n);
    occurrences (texte,n,T);
    afficher(T,indice_max(T));
End.

```

```

Exercice N° 43
Program AFFICHAGE;
Uses WinCrt;
Var CHM:string;
Procedure lire (Var ch:string);
Begin
Repeat
    Writeln ("Tapez une chaîne"); Readln (CHM);
Until (length (CHM) >= 5);
End;
Procedure afficher ( ch:string);
Var i:integer; ch1:string;
Begin
For i:= 1 To length (ch) Do
Begin
    ch1 := copy (ch,1,i) + copy (ch,length(ch) - i + 1,i);
    Writeln (ch1);
End;
End;
Begin
    lire (CHM);
    afficher (CHM);
End.

```

```

Exercice N° 44
Program JEU;
Uses WinCrt;
Var joueur1,joueur2 : integer;

Procedure jouer (Var score1,score2:integer);
Var l1,l2:integer;
Begin
    score1 := 0; score2 := 0; ranDomize;
Repeat
    l1 := ranDom (6) + 1;

```

```

    l2 := ranDom (6) + 1;
    writeln( l1, ' ',l2);
    If l1 > l2 Then score1 := score1 + 1
    Else If l1 < l2 Then score2 := score2 + 1;
Until ((score1 = 10) or (score2 = 10));
End;

Function gagnant (score1,score2:integer): integer;
Begin
If score1 > score2 Then gagnant := 1 Else gagnant := 2;
End;
Begin
    jouer (joueur1,joueur2);
    Writeln ('Le joueur gagnant c"est le joueur N° ',gagnant
(joueur1,joueur2));
End.

```

```

Exercice N° 45
Program SEQUENCE;
Uses WinCrt;
Type TAB = Array [1..20] of integer;
Var i,j,n,V_max,V_min:integer; T:TAB;

```

```

Procedure lire (Var x:integer);
Begin
Repeat
    Writeln ('Donnez la taille du tableau'); Readln (x);
Until (x in [2..20]);
End;
Procedure remplir (Var V:TAB;y:integer);
Var i:integer;
Begin
For i:= 1 To y Do
Begin
    Writeln ('Donnez la valeur de l"éléments N° ',i);
    Readln (V[i]);
End;
End;
Function lire_entier (x,y:integer):integer;
Var z:integer;
Begin
Repeat
    Writeln ('Donnez un entier compris entre ',x,' et ',y); Readln
(z);
Until (z in [x..y]);
    lire_entier := z;
End;
Function max (Var V: TAB; x,y: integer):integer;
Var i, Vmax :integer;
Begin
    Vmax := V[x];
For i:= x + 1 To y Do
    If Vmax < V[i] Then Vmax := V[i];

```

```

max := Vmax;
End;
Function min (Var V:TAB;x,y:integer):integer;
Var i,Vmin:integer;
Begin
Vmin := V[x];
For i:= x + 1 To y Do
  If Vmin > V[i] Then Vmin := V[i];
min := Vmin;
End;
Begin
lire (n);
remplir (T,n);
i:=lire_entier (1,n);
j:=lire_entier (i,n);
V_max := max (T,i,j);
V_min := min (T,i,j);
Writeln ('La valeur maximal dans cet intervalle = ',V_max);
Writeln ('La valeur minimal dans cet intervalle = ',V_min);
End.

```

**Exercice N° 46**

```

Program TRIANGLE;
Uses WinCrt;
Var ch:string;
Procedure lire (Var ch:string);
Begin
Repeat
  Writeln ('Tapez une chaîne'); Readln (ch);
Until (length (ch) >= 3);
End;
Procedure afficher (ch:string);
Var i,l:integer;
Begin
For i:= 1 To length (ch) Do Writeln (Copy (ch,1,i));
End;
Begin
lire (ch);
afficher (ch);
End.

```

**Exercice N° 47**

```

Program CRYPTER;
Uses WinCrt;
Type
TAB = Array [1..19] of char;
Var
T,R:TAB;
N,i,code1,code2:integer;

Procedure lire (Var x:integer);
Begin
Repeat
  Writeln ('Donnez la taille du tableau');

```

```

  Readln (x);
Until (x in [3..19]);
End;

Procedure remplir (Var V:TAB;x:integer);
Var i:integer;
Begin
For i:=1 To x Do
  Begin
    Writeln ('Donnez la valeur de l'élément N° ',i);
    Readln (V[i]);
  End;
End;

Function permuter (x:integer):integer;
Var
u,d:integer;
Begin
u:= x Mod 10;
d:= (x Mod 100) Div 10;
permuter:= ((x Div 100) * 100) + (u*10) + d;
End;

```

**Procedure** afficher (V:TAB;y:**integer**);

```

Var i:integer;
Begin
For i:= 1 To y Do Writeln (V[i]);
End;

```

**Begin**

```

lire (N);
remplir (T,N);
For i:= 1 To N Do
  Begin
    code1 := Ord(T[i]);
    code2 := permuter (code1);
    R[i] := Chr (code2);
  End;
afficher (R,N);
End.

```

**Exercice N° 48**

```

Program COMPOSE;
Uses WinCrt;
Type TAB = Array [1..6] of integer;
Var N:integer; T:TAB;
Procedure lire (Var x:integer);
Begin
Repeat
  Writeln ('Donnez un entier non nul composé de trois
chiffres');
  Readln(x);
Until ((x >= 100) and (x <= 999));
End;

```

**Procedure** Former (**Var** V:TAB;x:**integer**);

```

Var u,d,c:integer;
Begin
c:= x Div 100; d:= (x Mod 100) Div 10; u := x Mod 10;
V[1] := x;
V[2] := (c*100) + (u*10) + d;
V[3] := (u*100) + (d*10) + c;
V[4] := (u*100) + (c*10) + d;
V[5] := (d*100) + (c*10) + u;
V[6] := (d*100) + (u*10) + c;
End;

Procedure afficher (V:TAB);
Var i:integer;
Begin
For i:= 1 To 6 Do Write (V[i]:5);
Writeln;
End;

Function Min (V:TAB):integer;
Var i,x:integer;
Begin
x:= V[1];
For i:= 2 To 6 Do If V[i] < x Then x:= V[i];
Min := x;
End;

Function Max (V:TAB):integer;
Var i,x:integer;
Begin
x:= V[1];
For i:= 2 To 6 Do
  If V[i] > x Then x:= V[i];
Max := x;
End;

Begin
Lire (N);
Former (T,N);
Write ('Les nombres Formés par les chiffres de ',N,' sont:');
Afficher (T);
Writeln ('Le plus petit nombre est ',Min (T));
Writeln ('Le plus grand nombre est ',Max (T));
End.

```

**Exercice N° 49**

```

Program ECART_MINIMAL;
Uses WinCrt;
Type TAB = Array [1..20] of integer;
Var v:TAB; n,i,s1,s2:integer;
Procedure lire (Var x:integer);
Begin
Repeat
  Writeln ('Donnez la taille du tableau');
  Readln (x);
Until (x in [5..20]);
End;

```

```

Procedure saisir (Var t:tab;n:integer);
Var i:integer;
Begin
For i := 1 to n Do
Begin
  Repeat
    Writeln('Donnez la valeur de l'élément N° ',i);
    Readln (t[i]);
  Until (T[i] >= 0);
End;
End;

```

```

Function somme (T:TAB;p1,p2:integer):integer;
Var
s,i:integer;
Begin
s := 0;
For i := p1 to p2 Do
  s := s + t[i];
somme := s;
End;
Begin
lire (n);
saisir (v,n);
i := 1;
Repeat
  i := i + 1;
  s1 := somme (v,1,i-1);
  s2 := somme (v,i+1,n);
Until (s1 >= s2) or (i = n-1);
Writeln ('S1 = ',s1,' , S2 = ',s2,' et ind = ',i);
End.

```

```

Exercice N° 50
Program DIVISEURS_MULTIPLES;
Uses WinCrt;
Type TAB = Array [1..15] of integer;
Var ind_inf,ind_sup,n,p,i:integer;
T:TAB;
diviseurs,multiples,ch:string;

```

```

Procedure lire (Var x:integer);
Begin
Repeat
  Writeln('Donnez la taille du tableau');
  Readln(x);
Until ( x in [1..15]);
End;

```

```

Procedure remplir (Var T:TAB;n:integer);
Var i : integer;
Begin
For i := 1 to n Do

```

```

Begin
  Writeln ('Donnez la valeur de l'élément N° ', i);
  Readln(T[i]);
End;
End;

Function diviseur (x,y:integer):boolean;
Begin
diviseur := false;
If (y mod x) = 0 then diviseur := true;
End;
Begin
lire (n);
remplir (T,n);
Repeat
  Writeln('Donnez l'indice supérieur');
  Readln(ind_sup);
Until ( ind_sup in [2..n]);
Repeat
  Writeln('Donnez l'indice inférieur');
  Readln(ind_inf);
Until (ind_inf in [1..ind_sup-1]);
Writeln('Tapez l'entier qu'on cherche ces diviseurs et ces multiples');
Readln(p);
For i:= ind_inf to ind_sup Do
  If T[i] < p then
    Begin
      If diviseur(T[i],p) then
        Begin
          str(T[i],ch);
          diviseurs := diviseurs + ch + ' ';
        End;
      End;
    Else
      If diviseur (p,T[i]) then
        Begin
          str(T[i],ch);
          multiples := multiples + ch + ' ';
        End;
    End;
Writeln ('Les diviseurs de ',p,' sont : ',diviseurs);
Writeln ('Les multiples de ',p,' sont : ',multiples);
End.

```

```

Exercice N° 51
Program ROTATION;
Uses WinCrt;
Type
TAB = array [1..25] of real;
Var
A,B:TAB;
n,k:integer;

```

```

Procedure lire (Var x:integer);
Begin
Repeat
  Writeln ('Donnez la taille du tableau A');
  Readln (x);
Until (x in [2..25]);
End;

```

```

Procedure remplir (Var T:TAB;y:integer);
Var
I:integer;
Begin
For i:= 1 to y Do
Begin
  Writeln('Donnez la valeur de l'élément N° ',i);
  Readln(T[i]);
End;
End;

```

```

Procedure lire_k(Var x,y:integer);
Begin
Repeat
  Writeln('Donnez le nombre d'éléments de rotation');
  Readln(x);
Until (x in [1..y-1]);
End;

```

```

Procedure pivoter (Var A,B:TAB;k,n:integer);
Var
i:integer;
Begin
For i:= 1 to n Do
  If ((i - k) >= 1) then
    B[i - k] := A[i]
  Else
    B[n - k + i] := A[i];
End;

```

```

Procedure afficher(T:TAB;y:integer);
Var
i:integer;
Begin
For i:= 1 to y Do
  Writeln(T[i]:6:3);
End;

Begin
lire (n);
remplir(A,n);
lire_k(k,n);
pivoter(A,B,k,n);
afficher(B,n);
End.

```