

# Exercice 1

- **Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?**

Var A, B : Entier

début

A ← 1

B ← A + 3

A ← 3

Fin

La valeur des variables est :

A = 1                      B = ?

A = 1                      B = 4

A = 3                      B = 4

## Exercice 2

- Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

Var A, B, C :Entier;

début

A ← 5

B ← 3

C ← A + B

A ← 2

C ← B - A

fin

La valeur des variables est :

A = 5                      B = ?                      C = ?

A = 5                      B = 3                      C = ?

A = 5                      B = 3                      C = 8

A = 2                      B = 3                      C = 8

A = 2                      B = 3                      C = 1

## Exercice 3

- **Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?**

Var A, B : Entier

début

A ← 5

B ← A + 4

A ← A + 1

B ← A - 4

fin

La valeur des variables est :

A = 5                      B = ?

A = 5                      B = 9

A = 6                      B = 9

A = 6                      B = 2

## Exercice 4

- Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

Var A, B, C : Entier;

début

A ← 3;

B ← 10;

C ← A + B;

B ← A + B;

A ← C;

fin

La valeur des variables est :

A = 3

B = ?

C = ?

A = 3

B = 10

C = ?

A = 3

B = 10

C = 13

A = 3

B = 13

C = 13

**A = 13**

**B = 13**

**C = 13**

## Exercice 5

- **Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?**

Var A, B : Entier

début

A ← 5

B ← 2

A ← B

B ← A

fin

La valeur des variables est :

A = 5

B = ?

A = 5

B = 2

A = 2

B = 2

A = 2

B = 2

## Exercice 6

- **Ecrire un algorithme permettant d'échanger les valeurs de deux variables A et B.**

algorithme permutation;

Var A, B, C : Entier;

début

Lire(A);

Lire (B);

    C ← A;

    A ← B;

    B ← C;

Ecrire(A);

Ecrire(B);

fin

## Exercice 7

- **On dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C.**

algorithme permutation;

Var A, B, C, D : Entier;

début

Lire(A); Lire (B); Lire(C);

D ← C;

C ← B;

B ← A;

A ← D;

Ecrire(A); Ecrire(B); Ecrire(C);

fin

## Exercice 8

- **Que produit l'algorithme suivant ?**

Var A, B, C : Caractères

début

A ← "423 " ;

B ← "12 " ;

C ← A + B;

fin

Il ne peut produire qu'une erreur d'exécution, puisqu'on ne peut pas additionner des caractères.



## Exercice 9

- **Que produit l'algorithme suivant ?**

Var A, B, C : Caractères

début

A ← "423 " ;

B ← "12 " ;

C ← A & B;

fin

En revanche, on peut les concaténer. A la fin de l'algorithme, C vaudra donc "42312".

## Exercice 10

- **Quel résultat produit le programme suivant ?**

algorithme double;

Var val, Double: entier;

début

Val ← 231;

Double ← Val \* 2;

Ecrire (Val);

Ecrire (Double);

fin

**On verra apparaître à l'écran 231, puis 462 (qui vaut  $231 * 2$ )**

## Exercice 11

- **Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.**

algorithme carré;

Var nb, carr : Entier;

début

Ecrire( "Entrez un nombre : ");

Lire (nb);

carr ← nb \* nb;

Ecrire ( "Son carré est : " , carr);

fin

## Exercice 12

- **Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement. ( $TTC = NA * HT * (1 + TVA)$ )**

```
algorithme prix TTC d'un article;  
var nb, pht, ttva, pttc :réel;  
  début  
    Ecrire ("Entrez le prix hors taxes : ");  
    Lire (pht);  
    Ecrire( "Entrez le nombre d'articles : " );  
    Lire (nb);  
    Ecrire ("Entrez le taux de TVA :");  
    Lire (ttva);  
    pttc ← nb * pht * (1 + ttva);  
    Ecrire ("Le prix toutes taxes est : ", pttc);  
  fin
```

## Exercice 13

- **Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).**

algorithme test positif ou négatif;

Var n : Entier;

début

Ecrire ("Entrez un nombre : " );

Lire (n);

Si  $n > 0$  Alors

Ecrire ("Ce nombre est positif");

Sinon

Ecrire ("Ce nombre est négatif ");

Finsi

fin

## Exercice 14

- Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas calculer le produit des deux nombres.**

algorithme test produit positif ou négatif;

Var m, n : Entier;

début

Ecrire ("Entrez deux nombres : ");

Lire (m); Lire(n);

Si  $(m > 0 \text{ ET } n > 0) \text{ OU } (m < 0 \text{ ET } n < 0)$  Alors

Ecrire ("Leur produit est positif ");

Sinon

Ecrire ("Leur produit est négatif ");

Finsi

fin

## Exercice 15

- **Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.**

algorithme ordre alphabétique;

Var a, b, c : Caractère

début

Ecrire ("Entrez successivement trois noms : ");

Lire(a); Lire(b); Lire(c);

Si (a < b ET b < c) Alors

Ecrire ("Ces noms sont classés alphabétiquement");

Sinon

Ecrire ("Ces noms ne sont pas classés ");

Finsi

fin

## Exercice 16

- **Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).**

algorithme test positif ou négatif;

Var n : Entier;

début

Ecrire ("Entrez un nombre : " );

Lire (n);

Si  $n < 0$  Alors

Ecrire ("Ce nombre est négatif " );

Sinon

Si  $n = 0$  Alors

Ecrire ("Ce nombre est nul " );

Sinon

Ecrire ("Ce nombre est positif " );

Finsi

Finsi

fin



## Exercice 17

- **Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul).**

Var m, n : Entier;

début

Ecrire ("Entrez deux nombres : ");

Lire (m); Lire(n);

Si (m = 0 OU n = 0) Alors

Ecrire ("Le produit est nul ");

Sinon

Si (m < 0 ET n < 0) OU (m > 0 ET n > 0) Alors

Ecrire ("Le produit est positif ");

Sinon

Ecrire "Le produit est négatif«

Finsi

Finsi

fin

## Exercice 18

- Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :
  - "Poussin" de 6 à 7 ans
  - "Pupille" de 8 à 9 ans
  - "Minime" de 10 à 11 ans
  - "Cadet" après 12 ans

Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

## Exercice 18

**Var** age : Entier;

**début**

Ecrire ("Entrez l'âge de l'enfant : " );

Lire (age);

Si (age  $\geq$  12) Alors

    Ecrire ("Catégorie Cadet " );

Sinon

    Si (age  $\geq$  10) Alors

        Ecrire ("Catégorie Minimale " );

    Sinon

    Si (age  $\geq$  8) Alors

        Ecrire ("Catégorie Poupille " );

    Sinon

        Si (age  $\geq$  6) Alors

            Ecrire ("Catégorie Poussin " );

        Finsi

    Finsi

    Finsi

    Finsi

**fin**

## Exercice 19

- Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

**algorithme** affichage des nombres;

**Var** N, i : Entier

**début**

Ecrire ("Entrez un nombre :");

Lire (N);

Ecrire ("Les 10 nombres suivants sont : ");

**Pour i de N + 1 à N + 10 faire**

Ecrire (i);

**finpour**

**fin**

## Exercice 20

- Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :  $1 + 2 + 3 + 4 + 5 = 15$

**algorithme** Somme;

**Var** N, i, Som : Entier;

**début**

Ecrire ("Entrez un nombre :" );

Lire (N);

Som  $\leftarrow$  0;

**Pour** i de 1 à N faire

Som  $\leftarrow$  Som + i;

**finpour**

Ecrire ("La somme est : ", Som );

**fin**

## Exercice 21

- Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

**algorithme** Factorielle;

**Var** N, i, F: Entier;

**début**

Ecrire ("Entrez un nombre :" );

Lire (N);

F ← 1;

**Pour i de 2 à N faire**

F ← F \* i;

**finpour**

Ecrire ("La factorielle est : ", F);

**fin**

## Exercice 22-A

- Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dit ensuite quel était le plus grand parmi ces 20 nombres.

**algorithme** PG\_NBR parmi 20;

**Var** N, i, PG : Entier;

**début**

PG ← 0;

**Pour i de 1 à 20 faire**

Ecrire ("Entrez un nombre :");

Lire (N);

**Si i = 1 ou N > PG Alors**

PG ← N;

**FinSi**

Ecrire ("Le nombre le plus grand était : ", PG);

**finpour**

**fin**

## Exercice 22-B

- Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre.

**algorithme** PG\_NBR parmi 20;

**Var** N, i, PG, IPG : Entier;

**début**

PG ← 0;

**Pour** i de 1 à 20 **faire**

Ecrire ("Entrez un nombre :");

Lire (N);

**Si**  $i = 1$  ou  $N > PG$  **Alors**

PG ← N;      Ecrire ("Le nombre le plus grand était : ", PG);

IPG ← i;      Ecrire (« Il a été saisi en position : ", IPG);

**FinSi**      **fin**

**finpour**



## Exercice 23

- Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

**algorithme** PG\_NBR;

**Var** N, i, PG, IPG : Entier;

**début**

N ← 1; i ← 0; PG ← 0;

**tant que** N ≠ 0 **faire**

Ecrire ("Entrez un nombre : " );

Lire (N);

i ← i + 1;

**Si** i = 1 ou N > PG **Alors**

PG ← N;

IPG ← i;

**FinSi**

**FinTantQue**

Ecrire ("Le nombre le PG était : ", PG);

**fin**

## Exercice 24

- Les habitants de Paris paient l'impôt selon les règles suivantes :
  - les hommes de plus de 20 ans paient l'impôt
  - les femmes paient l'impôt si elles ont entre 18 et 35 ans
  - les autres ne paient pas d'impôt

Le programme demandera donc l'âge et le sexe du Parisien, et se prononcera donc ensuite sur le fait que l'habitant est imposable.

## Exercice 24

**algorithme** Impôt;

**Var** sex : Caractère;

**Var** age : entier;

**Var** C1, C2 : booléen;

**début**

Ecrire ("Entrez le sexe (M/F) : "); **fin**

Lire (sex);

Ecrire ("Entrez l'âge: ");

Lire (age);

C1 ← sex = "M" ET age > 20;

C2 ← sex = "F" ET (age > 18 ET age < 35);

Si C1 ou C2 Alors

Ecrire ("Imposable");

Sinon

Ecrire ("Non Imposable ");

finSi

## Exercice 25

- Écrire un algorithme saisissant 2 variables entières qui calcule et affiche leur moyenne.

**algorithme** calculDeMoyenne;

**Var** a, b : entier;

**Var** moy : réel;

**début**

Ecrire ("Donnez la 1ière valeur");

Lire (a);

Ecrire ("Donnez la 2ième valeur");

Lire (b);

moy  $\leftarrow$  (a+b)/2;

Ecrire ("La moyenne est:", moy);

**fin**

## Exercice 26

- Saisir 3 entiers  $a$ ,  $b$ ,  $c$  et déterminer dans  $\mathbb{R}$  les racines de l'équation  $ax^2 + bx + c = 0$ .

**algorithme** RésolutionEquation;

**Var**  $a$ ,  $b$ ,  $c$ , delta : réel;

**début**

Ecrire ("Donnez la valeur de a");

Lire (a);

Ecrire ("Donnez la valeur de b");

Lire(b);

Ecrire ("Donnez la valeur de c");

Lire (c);

# Exercice 26

**si**  $a=0$  **Alors**

Ecrire ('Equation de 1er ordre');

**si**  $b=0$  **Alors**

**si**  $C=0$  **alors**

Ecrire (' Tous réel est solution');

**sinon**

Ecrire ('Pas de solution');

**finsi**

**sinon**

Ecrire ('Solution unique  $x=$ ',  $-C/B$ );

**finsi**

**sinon**

$\text{delta} \leftarrow B*B-4*A*C;$

**si**  $\text{delta} < 0$  **alors**

Ecrire ('Pas de solution');

**sinon**

**si**  $\text{delta} = 0$  **alors**

Ecrire (' Solution double :  $x_1=x_2=$ ',  $-B/2*A$ );

**sinon**

$\text{delta} \leftarrow \text{racine}(\text{delta})$

Ecrire ('Deux solutions  $x_1=$ ',  $(-B-(\text{delta}))/2*A$ , ' ;  $x_2=$ ',  $(-B+(\text{delta}))/2*A$ );

**finsi**

**finsi**

**finsi**

**fin**

## Exercice27

- Ecrire un algorithme qui permet donner le résultat d'un étudiant à un module sachant que ce module est sanctionné par une note d'oral de coefficient 1 et une note d'écrit de coefficient 2. La moyenne obtenue doit être supérieure ou égale à 10 pour valider le module.

## Exercice27

Prof.Fatima IBRAHIMI

**algorithme** resultatEtudiant;

**Var** N1, N2, Moy : réel;

**début**

Ecrire (' Donnez la note d'oral');

Lire (N1);

Ecrire (' Donnez la note d'écrit');

Lire (N2);

Moy  $\leftarrow (N1+2*N2)/3$

si moy <10 Alors

Ecrire(" Module non validé ");

sinon

Ecrire(" Module validé ");

finsi

**fin**



## Exercice28

- Ecrire l'algorithme permettant d'imprimer le triangle suivant, le nombre de lignes étant donné par l'utilisateur :

1

12

123

1234

12345

123456

1234567

## Exercice28

**algorithme** Triangle;

**Var** j, n, i : entiers;

**début**

Ecrire ('Donnez le nombre de ligne du triangle');

Lire (n);

**pour** i= 1 **à** n **faire**

**pour** j=1 **à** i **faire**

Ecrire(j)

**fin pour**

**fin pour**

**fin**

## Exercice29

- Ecrire un algorithme qui demande un nombre , calcule et affiche la somme  $\sum_{i=1}^n i^3$

**algorithme** SommeCubique;

**Var** i, n, som : entier;

**début**

Ecrire (' Donnez n');

Lire (n);

som ← 0;

**pour** i=1 à n **faire**

som ← som+i\*i\*i

**finpour**

Ecrire (' La somme cubiques des n entiers est :',som);

**fin**

## Exercice30

- Donnez le résultat de la multiplication de deux nombres en ne faisant que des additions.

**algorithme** multiplication;

**Var** a, b, i, res : Entier;

**début**

Ecrire (" donnez les valeurs a et b" );

Lire(a); Lire(b);

res ← 0;

**pour** i de 1 à b **faire**

res ← res + a;

**finpour**

Ecrire (" la multiplication de a\*b : " , res);

**fin**

## Exercice31

- Ecrire un algorithme qui déclare et remplit un tableau contenant les six voyelles de l'alphabet latin.

**algorithme** Affichage des voyelles;

**Tableau Tab(5) : Caractère**

**début**

Tab (0) ← "a " ;

Tab (1) ← "e " ;

Tab (2) ← "i " ;

Tab (3) ← "o " ;

Tab (4) ← "u " ;

Tab (5) ← "y " ;

**fin**

## Exercice32

- Ecrire un algorithme qui déclare un tableau de 9 notes, dont on fait ensuite saisir les valeurs par l'utilisateur.

**algorithme** Saisi des notes;

**Tableau** Note(8) : entier;

**Var** i : entier;

**début**

**Pour i de 0 à 8 faire**

    Ecrire ("Entrez la note numéro ", i + 1);

    Lire (Notes(i));

**finpour**

**fin**

## Exercice33

- Ecrivez un algorithme permettant à l'utilisateur de saisir un nombre quelconque de valeurs, qui devront être stockées dans un tableau. L'utilisateur doit donc commencer par entrer le nombre de valeurs qu'il compte saisir. Il effectuera ensuite cette saisie. Enfin, une fois la saisie terminée, le programme affichera le nombre de valeurs négatives et le nombre de valeurs positives.

## Exercice33

**algorithme** Valeur\_neg\_pos;

**Var** n, nbpos, nbneg : **entier**;

**Tableau** T( ) : **entier** ;

**début**

Ecrire ( "Entrez le nombre de valeurs : " );

Lire (n);

**Redim** T(n-1);

nbpos ← 0;

nbneg ← 0;

**Pour** i de 0 à n – 1 faire

Ecrire ("Entrez le nombre n<sup>o</sup> ", i);

Lire T(i);

**Si** T(i) > 0 **alors**

nbpos ← nbpos + 1;

**Sinon**

nbneg ← nbneg + 1;

**Finsi**

**finpour**

**Ecrire** ("Nombre de valeurs positives : ",  
nbpos);

**Ecrire** ("Nombre de valeurs négatives : ",  
nbneg);

**fin**



## Exercice 34

- Ecrivez un algorithme calculant la somme des valeurs d'un tableau (on suppose que le tableau a été préalablement saisi).

**algorithme** somme;

**Var** i, som, N : entier;

**Tableau** T() : entier;

**début**

Ecrire ( "Entrez le nombre de valeurs : " );

Lire (n);

**Redim** T(n-1);

som ← 0;

**Pour** i de 0 à n – 1 **faire**

som ← som + T(i);

**finpour**

Ecrire ("Somme des éléments du tableau : ", som);

**fin**

## Exercice 35

- Ecrivez un algorithme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ.

Tableau 1 :

4	8	7	9	1	5	4	6
---	---	---	---	---	---	---	---

Tableau 2 :

7	6	5	2	1	3	7	4
---	---	---	---	---	---	---	---

Tableau à constituer :

11	14	12	11	2	8	11	10
----	----	----	----	---	---	----	----

## Exercice35

**algorithme** somme;

**Var** i, n : entier;

**Tableaux** T1(), T2(), T3() : entier;

**début**

Ecrire ( "Entrez le nombre de valeurs : " );

Lire (n);

**Redim** T1(n-1);

**Redim** T2(n-1);

**Redim** T3(n-1);

{Lecture des valeur des deux tableau }

**Pour** i de 0 à n – 1 **faire**

T3(i) ← T1(i) + T2(i);

**finpour**

**fin**

## Exercice 36

- Toujours à partir de deux tableaux précédemment saisis, écrivez un algorithme qui calcule le schtroumpf des deux tableaux. Pour calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout. Par exemple si l'on a :

- Tableau 1 :

2	5	8	4
---	---	---	---

- Tableau 2 :

6	7
---	---

- Le Schtroumpf sera :

$$6*2 + 6*5 + 6*8 + 6*4 + 7*2 + 7*5 + 7*8 + 7*4 = 247$$

# Exercice36

**algorithme** somme;

**Var** i, j, N1, N2, S : entier;

**Tableaux** T1(), T2() : entier;

**début**

Ecrire ( "Entrez la taille de T1 et T2: " );

Lire (N1); Lire (N2);

**Redim** T1(N1-1); **Redim** T2(N2-1);

S ← 0;

**Pour** i de 0 à N1 – 1 **faire**

**Pour** j de 0 à N2 – 1 **faire**

    S ← S + T1(i) \* T2(j);

**finpour**

**finpour**

Ecrire( "Le schtroumpf est : ", S);

**fin**

## Exercice 37

- Ecrivez un algorithme qui permette la saisie d'un nombre quelconque de valeurs. Toutes les valeurs doivent être ensuite augmentées de 1, et le nouveau tableau sera affiché à l'écran.

**algorithme** incrémentation du tableau;

**Var** n, i : entier;

**Tableau** T() : entier;

**début**

Ecrire ("Entrez le nombre de valeurs : ");

Lire (n);

**Redim** T(n-1);

**Pour** i de 0 à n – 1 **faire**

Ecrire ("Entrez le nombre n° ", i + 1);

Lire T(i);

**finpour**

**Ecrire** ("Nouveau tableau :");

**Pour** i de 0 à n – 1 **faire**

T(i) ← T(i) + 1;

**Ecrire** T(i);

**finpour**

**fin**

## Exercice38

- Ecrivez un algorithme permettant, toujours sur le même principe, à l'utilisateur de saisir un nombre déterminé de valeurs. Le programme, une fois la saisie terminée, renvoie la plus grande valeur en précisant quelle position elle occupe dans le tableau. On prendra soin d'effectuer la saisie dans un premier temps, et la recherche de la plus grande valeur du tableau dans un second temps.

## Exercice38

**algorithme** Afficher Max;

**Var** n, i, PosMax: entier;

**Tableau** T() : entier;

**Début**

Ecrire ("Entrez le nbr de valeurs :");

Lire (n);

**Redim** T(n-1);

**Pour** i de 0 à n – 1 **faire**

Ecrire ("Entrez le nbr n<sup>o</sup> ", i + 1);

Lire (T(i));

**finpour**

PosMax ← 0;

**Pour** i de 1 à n – 1 **faire**

**Si** T(i) > T(PosMax) **alors**

**PosMax** ← i;

**finsi**

**finpour**

Ecrire ( "Elément le plus grand : ",

T(PosMax));

**Ecrire** ("Position de cet élément : ",

PosMax);

**Fin**



## Exercice39

- Toujours et encore sur le même principe, écrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe. Le programme, une fois la saisie terminée, renvoie le nombre de ces notes supérieures à la moyenne **de la classe**.

## Exercice39

**algorithme** Afficher note supérieur à la moyenne;

**Var** n, i, Som, Moy, nsup : entier;

**Tableau** T() : entier;

**début**

Ecrire ("Entrez le nombre de notes à saisir :");

Lire (n);

Redim T(n-1);

**Pour i de 0 à n – 1 faire**

Ecrire ("Entrez nbr n° ", i + 1);

Lire (T(i));

**finpour**

Som ← 0;

**Pour i de 0 à n – 1 faire**

Som ← Som + T(i);

**finpour**

Moy ← Som / n;

nsup ← 0;

**Pour i de 0 à n – 1 faire**

**Si T(i) > Moy Alors**

nsup ← nsup + 1;

**finsi**

**finpour**

**Ecrire** (nsup, " élèves dépassent la moyenne de la classe ");

**fin**

## Exercice 40

- Soit un tableau  $T$  à deux dimensions  $(12, 8)$  préalablement rempli de valeurs numériques. Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau.

## Exercice 40

**algorithme** Afficher Max;

**Var** i, j, imax, jmax: entier;

**Tableau** T(12, 8) : entier;

**début**

imax  $\leftarrow$  0;    jmax  $\leftarrow$  0;

**Pour** i de 0 à 12 **faire**

**Pour** j de 0 à 8 **faire**

**Si** T(i,j) > T(imax,jmax) **Alors**

            imax  $\leftarrow$  i;

            jmax  $\leftarrow$  j;

**finSi**

**finpour**

**finpour**

**Ecrire** ("Le plus grand élément est ", T(iMax, jMax));

**Ecrire** ("Il se trouve aux indices ", iMax, "; ", jMax);

**fin**

## Exercice 41

- Ecrire l'algorithme effectuant le décalage des éléments d'un tableau.
- Tableau initial 

D	E	C	A	L	A	G	E
---	---	---	---	---	---	---	---
- Tableau modifié (décalage à gauche) 

E	C	A	L	A	G	E	D
---	---	---	---	---	---	---	---

## Exercice 41

**algorithme** décalage;

**Var** tmp: caractère;

Var i, n : entier;

**Tableau** T() : entier;

**Début**

Ecrire (" donnez la valeur n");

Lire(n);

Redim (T(n- 1));

    tmp ← T(0);

    Pour i de 0 à n-2 Faire

        T(i) ← T(i+1);

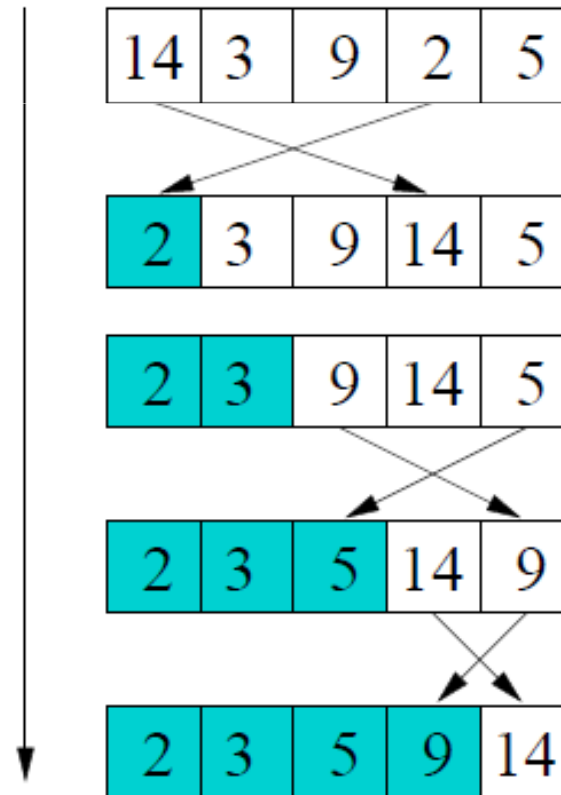
    finpour

T(n - 1) ← tmp;

**fin**

## Exercice 42

- Écrire un algorithme triant un tableau par sélection.
  - on cherche l'élément de plus petite valeur dans le tableau
  - le placer en tête du tableau
  - recommencer avec le tableau moins la première case



## Exercice 42

Prof. Fatima IBRAHIMI

**algorithme** tri sélection;

**Var** i, n, tmp : entier;

**Tableau** T() : entier;

**début**

**pour** i de 0 à n - 2 **faire**

        mi ← i;

**pour** j de i+1 à n - 1 **faire**

**Si** T(j) < T(mi) **alors**

                tmp ← T(mi);

                T(mi) ← T(j);

                T(j) ← tmp;

**finsi**

**finpour**

**finpour**

**fin**



## Exercice 43

- Écrire un algorithme qui permet d'inverser un tableau.

**algorithme** inversion tableau;

**Var** i, n, tmp : entier;

**Tableau** T() : entier;

**début**

Ecrire (" donner la valeur de n" );

Lire (n);

Redim (T(n-1));

**pour** i de 0 à n/2 **faire**

    tmp ← T(i);

    T(i) ← T((n-1) - i);

    T((n-1) - i) ← tmp;

**finpour**

**fin**

## Exercice 44

- Ecrire l'algorithme permettant d'effectuer la multiplication de deux vecteurs de taille N connue à l'avance et d'afficher le résultat.
- Rappel : Soient  $V_1$  et  $V_2$  deux tableaux.

$V_1$				
a	b	c	d	e

$V_2$				
f	g	h	i	j

$$V_1 * V_2 = a*f + b*g + c*h + d*i + e*j$$

## Exercice 44

**algorithme** produit vecteur;

**Var** i, n, res : entier;

**Tableau** V1(), V2() : entier;

**début**

Ecrire ( "donner la valeur n " );

Lire (n);

Redim (V1(n)); Redim (V2(n));

Res ← 0;

**Pour** i de 0 à n-1 **Faire**

Res ← Res + V1(i) \* V2(i);

**Fin Pour**

Ecrire ( " Le résultat est ", Res);

**fin**

## Exercice 45

- Ecrire un algorithme qui insère un élément donné  $X$  à la position  $K$  d'un tableau.

**algorithme** insertion\_elt\_X;

**Var** i, n, elt, pos, tmp: entiers;

**Tableau** T (): entier;

**début**

{lecture du tableau}

$n \leftarrow n + 1$ ;

Ecrire ( " entrer l'élément à ajouter " );

Lire (elt);

Ecrire ( " entrer sa position " );

Lire (pos);

**Pour i de pos à n – 1 faire**

tmp  $\leftarrow$  T(i);

T(i)  $\leftarrow$  elt;

elt  $\leftarrow$  tmp;

**finpour**

**fin**

## Exercice 46

- Soit  $T$  un tableau rangés dans l'ordre croissant. Ecrire un algorithme qui insère un élément donné  $X$  dans le tableau  $T$  en respectant l'ordre croissant?

# Exercice 46

**algorithme** insérer X;

**Var** i, j, n, elt, tmp : entiers;

**Tableau** T() : entier;

**début**

Ecrire("entrez la taille du tab:");

Lire(n);

Redim (T(n - 1));

**Pour** i de 0 à n - 1 **faire**

Ecrire("entrez l'élément:", i+1);

Lire(T(0));

**finpour**

$n \leftarrow n + 1;$

Ecrire("entrez l'élément X:");

Lire(elt);

$T(n - 1) \leftarrow \text{elt};$

**Pour** i décroissant de n - 1 à 1

**faire**

**pour** j de 1 à i **faire**

**si**  $T(j - 1) > T(j)$  **alors**

$\text{tmp} \leftarrow T(j - 1);$

$T(j - 1) \leftarrow T(j);$

$T(i) \leftarrow \text{tmp};$

**finsi**

**finpour**

**finpour**

**Pour** i de 0 à n - 1 **faire**

Ecrire (T(i));

**finpour**

**fin**

## Exercice 47

- Donner un algorithme pour supprimer un élément donné d'un tableau d'entiers.

**algorithme** Suppression d'un elt;

**Var** i , n, elt, Temp :Entiers ;

**Tableau** T() :Entier;

**début**

{Lecture et remplissage du tableau}

Ecrire("Entrez l'élément à supprimer :");

Lire(elt);

**pour** i de 0 à n – 1 **faire**

**si** elt = T(i) **alors**

        Pos ← i;

**finsi**

**finpour**

**pour** i de pos à n – 1 **faire**

    T(i) ← T(i + 1);

**finpour**

**pour** i de 0 à n – 2 **faire**

    Ecrire (T(i) );

**finpour**

**fin**

## Exercice 48

- Soit T un tableau de N entiers. Ecrire l'algorithme qui détermine le plus grand élément de ce tableau.

**algorithme** Grand élément;

**Var** i , max, n : entier;

**Tableau** T () : entier ;

**début**

Ecrire("Entrez la taille du tab");

Lire (n);

Redim (T(n - 1));

**pour** i de 0 à n - 1 faire

Ecrire ("Entrez l'élément:" ,i);

Lire (T(i) );

**finpour**

max ← T (0);

i ← 0;

**Répéter**

i ← i + 1;

**si** T (i) > max **alors**

Max ← T (i);

**finsi**

**Jusqu'à** i = (n - 1)

Ecrire (" Le plus grand élément est : ", max);

**fin**



# Exercice 49

Prof. Fatima IBRAHIMI

- Soit

## Exercice 50

- Proposez un algorithme permettant de calculer à la fois le minimum et le maximum d'un tableau.

```

Algorithme CalculMaxMin;
Var i, j, n, min, max : Entier;
Tableau T() : réel;
début
    max ← T(0);
    min ← T(0);
    
```

```

        pour i de 1 à n – 1 faire
            si T(i) < min alors
                min ← T(i);
            sinon
                si T(i) > max alors
                    max ← T(i);
                finsi
            finsi
        finpour
        Ecrire (" le max est : ", max, "le
min est : " , min);
    fin
    
```

## Exercice 51

Prof. Fatima IBRAHIMI

- Soit T un tableau de N réels. Ecrire un algorithme qui permet de calculer le nombre des occurrences d'un nombre X (c'est-à-dire combien de fois ce nombre X figure dans le tableau T).

**algorithme NBR occurrences;**

**Var i, n, X, cpt: entier;**

**Tableau T (): entier;**

**début**

Ecrire ( "donner le nombre X " );

Lire (X);

    cpt ← 0;

**Pour** i de 0 à n-1 **Faire**

**si** T (i) =X **alors**

            cpt ← cpt + 1;

**finsi**

**finpour**

Ecrire ( " Le nombre d'occurrences est ", cpt);

**fin**

## Exercice 52

- Ecrire un algorithme qui calcule le nombre d'entiers pairs et le nombre d'entiers impairs d'un tableau d'entiers.

**algorithme compter nbr pairs et nbr impairs;**

**Var i, n, nbP, nbImp: entier;**

**Tableau T (): entier;**

**début**

nbP ← 0;

nbImp ← 0;

**Pour** i de 0 à n-1 **Faire**

**si** T(i) MOD 2 = 0 **alors**

nbP ← nbP + 1;

**Sinon**

nbImp ← nbImp + 1;

**finsi**

**finpour**

Ecrire ( " Le nombre d'entiers pairs est ", nbP);

Ecrire ( " Le nombre d'entiers impairs est ", nbImp);

**fin**

# Exercice 53

[MCours.com](http://MCours.com)

- Soit un tableau  $T(n)$ , écrire un algorithme qui calcule VMEP (valeur moyenne des éléments positifs), VMEN (valeur moyenne des éléments négatifs) et NEM (nombre d'éléments nuls).

## Exercice 53

Prof. Fatima IBRAHIMI

**Algorithmme** Calcul VM;

**Var** i, n, nPos, nNeg, VmoyP,

VmoyN, nNull : entier;

**Var** somP, somN : entiers;

**Tableau** T ( ) : entier;

**début**

nPos ← 0; nNeg ← 0;

nNull ← 0;

somP ← 0; somN ← 0;

**pour** i de 0 à n – 1 faire

si t(i) = 0 Alors

nNull ← nNull + 1;

sinon

Ecrire( "Le nombre d'éléments nuls est : " , nNull );

Ecrire("La valeur moyenne des éléments positifs est: " , VmoyP);

Ecrire("La valeur moyenne des éléments négatifs est: " , VmoyN);

**fin**

si t(i) > 0 Alors

nPos ← nPos + 1;

somP ← somP + T(i);

sinon

nNeg ← nNeg + 1;

somN ← somN + T(i);

**finsi**

**finsi**

**finpour**

VmoyP ← somP/nPos;

VmoyN ← somN/nNeg;

## Exercice 54

- Donner un algorithme qui prend en argument un tableau d'entiers de taille  $n$  et qui le modifie de tels sorts que tous les entiers pairs se retrouvent avant les entiers impairs.

```

Var i , n, j: Entiers
Tableau T (), Tres():Entier
début
  j ← 0
  pour i de 0 à n – 1 faire
    si T(i) MOD 2 = 0 alors
      Tres ( j )←T(i);
      j ← j + 1;
    finsi
  finpour

```

```

  pour i de 0 à n – 1 faire
    si T(i) MOD 2 ≠ 0 alors
      Tres(j) ← T(i);
      j ← j + 1;
    finsi
  finpour
  pour i de 0 à n – 1 faire
    Ecrire ( Tres(i));
  finpour
fin

```

## Exercice 54

- On dispose des notes de 25 élèves ; chaque élève peut avoir une ou plusieurs notes mais toujours au moins une. Ecrire un programme permettant d'obtenir la moyenne de chaque élève lorsqu'on lui fournit les notes. On veut que les données et les résultats se présentent ainsi :

-Les parties italiques correspondent aux données tapées par l'utilisateur.

-La valeur -1 sert de critère de fin de notes pour chaque élève.

```
Notes de l'élève numéro 1
12
12
-1
Notes de l'élève numéro 2
.....
Notes de l'élève numéro 25
15
-1
Moyennes
Elève numéro 1 : 11
.....
Elève numéro 25 : 15
Moyenne de la classe : 12.3
```