



Ft : M.RAJI / G.INFO

mohamedraji1997@gmail.com

## Exercices Algorithmique

### Exercice1 :

Ecrire un algorithme permettant de saisir et d'afficher N éléments d'un tableau.

### Exercice2 :

Ecrire un algorithme permettant de calculer la somme, produit et moyenne des éléments d'un tableau.

### Exercice3 :

Ecrire un algorithme permettant de chercher toutes les occurrences d'un élément dans un tableau.

### Exercice4 :

Ecrire un algorithme permettant de calculer le nombre de fois pour lesquelles un élément apparait dans un tableau.

### Exercice5 :

Ecrire un algorithme permettant d'insérer un élément dans un tableau (au début milieu ou à la fin).

### Exercice6 :

Ecrire un algorithme permettant de supprimer un élément dans un tableau.

### Exercice7 :

Ecrire un algorithme permettant de trier par ordre croissant les éléments d'un tab

### Exercice8 :

Ecrire un algorithme permettant de fusionner les éléments de deux tableaux T1 et T2 dans un autre tableau T.

N.B :

N : nombre des éléments du tableau T1

M : nombre des éléments du tableau T2

### Exercice9 :

Ecrire un algorithme permettant de saisir les données d'un tableau à deux dimensions (10,4), de faire leur somme, produit et moyenne et de les afficher avec les résultats calcul à l'écran.

### Exercice10 :

Ecrire un algorithme qui calcule la somme des éléments de la diagonale d'une mat carrée  $M(n,n)$  donnée.



Ft : M.RAJI / G.INFO / CLUB FSF

mohamedraji1997@gmail.com

## Correction Exercices Algorithmique

### Correction Exercice1 :

Algorithme saisie\_affichage;

Var

T : tableau [1..100] de réels ;

N,i : réel ;

Debut

Ecrire('entrer le nombre d'éléments du tableau :') ;

Lire(N) ;

(\*lecture des éléments du tableau\*)

Pour i <-- 1 à N faire

Ecrire('entrer l'element N° ',i);

Lire(T[i]) ;

Finpour i

(\*afficher des éléments du tableau \*)

Pour i <-- 1 à N faire

Ecrire('l'element T['i,'] est : ',T[i]);

Finpour i

Fin

## Correction Exercice2 :

Algorithme somme\_produit\_moyenne;

Var

T : tableau [1..100] de réels ;

N,i : entiers ;

S,P,M : réels ;

Debut

Si N=0 alors

    Ecrire('le tableau est vide ');

Sinon

    S  $\leftarrow$  0 ;

    P  $\leftarrow$  1 ;

    Pour i  $\leftarrow$  1 à N faire

        S  $\leftarrow$  S+T[i] ;

        P  $\leftarrow$  P \* T[i] ;

    Finpour i

    M  $\leftarrow$  S/N ;

    Ecrire('la somme des éléments du tableau est : ',S);

    Ecrire('le produit des éléments du tableau est : ',P);

    Ecrire('la moyenne des éléments du tableau est : ',M);

Finsi

Fin

### Correction Exercice3 :

Algorithme recherche\_toutes\_occurences ;

Var

T : tableau [1..100] de réels ;

N,i : entiers ;

X : réel ;

Existe : booléen ;

Debut

Si N=0 alors

    Ecrire('le tableau est vide ');

Sinon

    Ecrire('entrer la valeur de l"élément à chercher :');

    Lire(X);

    Existe <-- Faux ;

    Pour i <-- 1 à N Faire

        Si T[i] = X alors

            Existe <-- Vrai ;

            Ecrire('l"élément à chercher apparait à la position : ',i

        Finsi

    Finpour i

    Si Existe = Faux alors

        Ecrire('l"élément n"apparait pas dans ce tableau ');

    Finsi

Fin

Fin

## Correction Exercice4 :

Algorithme frequence;

Var

T : tableau [1..100] de réels ;

N,i,F : entiers ;

X : réel ;

Debut

Si N=0 alors

Ecrire('le tableau est vide ');

Sinon

Ecrire('entrer la valeur de l'élément à chercher :');

Lire(X);

F  $\leftarrow$  0;

Pour i  $\leftarrow$  1 à N Faire

Si T[i] = X alors

F  $\leftarrow$  F+1;

Finsi

Finpour i

Ecrire('l'élément apparait : ', F,'fois dans ce tableau ');

Finsi

Fin

## Correction Exercice5 :

Algorithme insertion;

Var

T : tableau [1..100] de réels ;

N ,P,i: entiers ;

X : réel ;

Rep : caractère ;

Debut

Si N=0 alors

    Ecrire('le tableau est vide ');

Sinon

Ecrire('entrer la valeur de l'élément à insérer :');

Lire(X);

Ecrire(' Entrer la position d'insertion :');

Lire(P);

    Si (P<1) ou (P>N) alors

        Ecrire('Position hors limites du tableau ');

    Sinon

        Ecrire('Confirmer l'insertion (O/N) ');

        Lire(Rep);

        Si Rep='O' Alors

            N  $\leftarrow$  N+1 ;

            Pour i  $\leftarrow$  N à P+1 Faire

                T[i]  $\leftarrow$  T[i-1];

            Finpour i

            T[P]  $\leftarrow$  X ;

        Finsi

    Finsi

Finsi

Fin

## Correction Exercice6 :

Algorithme suppression;

Var

T : tableau [1..100] de réels ;

N ,P,i: entiers ;

Rep : caractère ;

Debut

Si N=0 alors

Ecrire('le tableau est vide ');

Sinon

Ecrire('entrer l'indice de l'élément à supprimer :');

Lire(P);

Si (P<1) ou (P>N) alors

Ecrire('Position hors limites du tableau ');

Sinon

Ecrire('la valeur dans cette position est :',T[P]);

Ecrire('Confirmer la suppression (O/N) ');

Lire(Rep);

Si Rep='O' Alors

N <-- N+1 ;

Pour i <-- P à N-1 Faire

T[i] <-- T[i+1];

Finpour i

N <-- N-1; (\*il y aura un élément en moins\*)

Finsi

Finsi

Finsi

Fin



## Correction Exercice7 :

```
Algorithme tri_Croissant;
Var
    T : tableau [1..100] de réels ;
    N ,i,j: entiers ;
    Aux: réel ;
Debut
    Si N=0 alors
        Ecrire('le tableau est vide ') ;
    Sinon
        Pour i <-- 1 à N-1 Faire
            Pour j <-- i+1 à N Faire
                Si T[i] > T[j] alors
                    Aux <-- T[i] ;
                    T[i] <-- T[j] ;
                    T[j] <-- Aux ;
                Finsi
            Finpour j
        Finpour i
    Finsi
Fin
```

## Correction Exercice8 :

Algorithme fusion\_deux\_tableaux;

Var

T1,T2 : tableau [1..100] de réels ;

T : tableau [1..200] de réels ;

N ,M,i: entiers ;

Debut

Si (N=0) et (M=0) alors

    Ecrire('le tableau est vide ');

Sinon

    Pour i <-- 1 à N Faire

        T[i] <-- T1[i] ;

    Finpour i

    Pour i <-- 1 à M Faire

        T[N+i] <-- T2[i] ;

    Finpour i

Finsi

Fin

## Correction Exercice9 :

```
Algorithme tableau_deux_dimension;
Var
    T : tableau [1..10,1..4] de réels ;
    I,j: entiers ;
    S,P,M : réels ;
Debut
    (*saisie des éléments du tableau *)
    Pour i <-- 1 à 10 Faire
        Pour j<-- 1 à 4 Faire
            Ecrire('entrer l'élément T[',i,',',j,'] :');
            Lire(T[i,j]) ;
        Finpour j;
    Finpour i
    (*calcul de la somme ,produit et moyenne *)
    S <-- 0 ;
    P <-- 1 ;
    Pour i <-- 1 à 10 Faire
        Pour j<-- 1 à 4 Faire
            S <-- S+T[i,j] ;
            P <-- P + T[i,j] ;
        Finpour j;
    Finpour i
    M <-- S/40 ; (*40 : nombre d'élément du tableau = 10x4 *)
    (* Affichage des éléments du tableau et des résultats *)
    Pour i <-- 1 à 10 Faire
        Pour j<-- 1 à 4 Faire
            Ecrire('l'élément T[',i,',',j,'] = ', T[i,j]);
        Finpour j;
    Finpour i
    Ecrire('la somme des éléments du tableau est :',S) ;
    Ecrire('le produit des éléments du tableau est :',P) ;
    Ecrire('la moyenne des éléments du tableau est :',M) ;
Fin
```

## Correction Exercice10 :

```
Algorithme diagonale_de_matrice;
Const
    N=8 ;
Var
    M : tableau [1..8,1..8] d'entiers;
    i,j: entiers ;
    Sdiag: entiers;
Debut
    (*saisie des éléments de la matrice*)
    Pour i <-- 1 à n Faire
        Pour j<-- 1 à n Faire
            Ecrire('entrer l'element M[',i,',',j,'] :');
            Lire(M[i,j]) ;
        Finpour j;
    Finpour i
    (*calcul de la somme des éléments de la diagonale *)
    Sdiag <-- 0 ;
    Pour i <-- 1 à n Faire
        Sdiag <-- Sdiag +M[i,j] ;
    Finpour i
    Ecrire('la somme des éléments de la diagonale est :',Sdiag) ;
Fin
```

