

---

## Série de TD N°04

---

### **Exercice 1 :**

Ecrire un algorithme qui permet de remplir un tableau de 10 éléments entiers. Ensuite on calcule la moyenne de ces éléments.

### **Exercice 2 :**

Ecrire un algorithme permet de :- remplir un tableau de 10 éléments entiers,.

- chercher le nombre d'occurrence d'une valeur introduite par l'utilisateur et afficher le résultat.

### **Exercice 3 :**

Ecrire un algorithme permettant de calculer le max et le min d'un tableau de 10 éléments introduit par l'utilisateur.

### **Exercice 4 :**

Écrire un algorithme qui remplit un tableau T de n ( $n < 100$ ) entiers. Le programme doit copier ensuite toutes les composantes strictement positives dans un deuxième tableau TPOS et toutes les valeurs strictement négatives dans un troisième tableau TNEG. Les tableaux TPOS et TNEG doivent être affichés par le programme.

### **Exercice 5:**

Écrire un algorithme qui demande de remplir une matrice de  $5 \times 5$  réels, et met Zéro dans tous les éléments du diagonale , puis affiche la matrice résultante.

### **Exercice 6 :**

Ecrire un algorithme qui demande de remplir deux Vecteurs A et B, de 30 réels et de les additionner dans un troisième tableau C, élément par élément, d'afficher le contenu de T3, puis de calculer leur produit scalaire.

## Solution série 04

### Solution Exo 01 :

En algorithmique	En C
<p>Algo moy_tab</p> <p>Var T : tableau [0..9] de entier</p> <p style="padding-left: 20px;">i, somme : entier</p> <p style="padding-left: 20px;">moyenne : réel</p> <p>Début</p> <p style="padding-left: 20px;">somme ← 0</p> <p>Ecrire ('entrer les 10 éléments du tableau')</p> <p>Pour (i allant de 0 à 9 ) alors</p> <p style="padding-left: 20px;">Lire (T[i])</p> <p style="padding-left: 20px;">somme ← somme + T[i]</p> <p>FinPour</p> <p style="padding-left: 20px;">moyenne ← somme / 10</p> <p>Ecrire (' la moyenne du tableau est ', moyenne)</p> <p>Fin</p>	<pre>#include&lt;stdio.h&gt; main () {     int t[10];     int i, som;     float moy ;     som = 0;     printf("entrer les éléments du tableau")     for(i=0; i&lt;10; i++)     {         scanf("%d",&amp;t[i]);         som = som + t[i];     }     moy = som/10;     printf ("la moyenne du tableau est %f \n", moy); }</pre>

### Solution Exo 02 :

En algorithmique	En C
<p>Algo Trouv</p> <p>Var T : tableau [0..9] de entier</p> <p style="padding-left: 20px;">i, n, nbr : entier</p> <p>Début</p> <p>Ecrire ('entrer n ')</p> <p>Lire (n)</p> <p>Ecrire ('entrer les 10 éléments du tableau')</p> <p>Pour (i allant de 0 à 9 ) faire</p> <p style="padding-left: 20px;">Lire (T[i])</p> <p>FinPour</p> <p style="padding-left: 20px;">nbr ← 0</p> <p>Pour (i allant de 0 à 9 ) faire</p> <p style="padding-left: 20px;">Si (n == T[i]) alors</p> <p style="padding-left: 40px;">nbr ← nbr +1</p> <p>FinPour</p> <p>Ecrire (n, 'a apparu ', nbr , 'fois dans le tablau')</p> <p>Fin</p>	<pre>#include&lt;stdio.h&gt; main () {     int t[10];     int i, n,nbr ;     printf("entrer n \n");     scanf ("%d",&amp;n);     printf("entrer les valeurs du tableau \n");      for(i=0; i&lt;10; i++)     {         scanf("%d",&amp;t[i]);     }     nbr =0;     for(i=0; i&lt;10; i++)     {         if (n == t[i])             nbr++;     }     printf("%d a apparu %d fois dans le tableau", n, nbr); }</pre>

### Solution Exo 3 :

En algorithmique	En C
<pre>Algo Max_min Var T : tableau [0..9] de entier     i, max, min : entier Début Ecrire ('entrer les 10 éléments du tableau') Pour (i allant de 0 à 9 ) faire     Lire (T[i]) FinPour Max ← T[0] Min ← T[0] /* c'est pas la peine de vérifier le premier élément donc i commence de 1 */ Pour (i allant de 1 à 9 ) faire     Si (T[i] &lt; min) alors         min ← T[i]     FinSi     Si (T[i] &gt; max) alors         max ← T[i]     FinSi FinPour Ecrire ('le max du tableau est ', max) Ecrire ('le min du tableau est ', min) Fin</pre>	<pre>#include&lt;stdio.h&gt; main () {     int t[10];     int i, min,max ;     printf("entrer lesvaleurs du tableau \n");      for(i=0; i&lt;10; i++)     {         scanf("%d",&amp;t[i]);     }     max =t[0];     min =t[0];     for(i=1; i&lt;10; i++)     {         if (t[i]&gt;max)             max = t[i];         if (t[i]&lt;min)             min = t[i];     }     printf("le max du tableau est %d \n", max);     printf("le min du tableau est %d \n", min); }</pre>

## Solution Exo 04 :

En algorithmique	En C
<pre>Algo Pos_Neg Var T : tableau [0..99] de entier     TPOS : tableau [0..99] de entier     TNEG : tableau [0..99] de entier     i, pos, neg : entier Début Ecrire ('entrer les 100 éléments du tableau') Pour (i allant de 0 à 99 ) faire     Lire (T[i])     FinPour /* -1 ça veut dire que le tableau est vide : l'indice commence de 0 */ pos ← -1 neg ← -1 Pour (i allant de 0 à 99 ) faire     Si (T[i] &gt; 0) alors         pos ← pos + 1         TPOS[pos] ← T[i]     FinSi     Si (T[i] &lt; 0) alors         neg ← Neg + 1         TNEG[neg] ← T[i]     FinSi FinPour Si ( pos= -1 ) alors     Ecrire ('le tableau des nombres positifs est vide' ) Sinon     Pour (i allant de 0 à pos ) faire         Ecrire (TPOS[i])     FinPour FinSi Si ( neg= -1 ) alors     Ecrire ('le tableau des nombres négatifs est vide' ) Sinon     Pour (i allant de 0 à neg ) faire         Ecrire (TNeg[i])     FinPour FinSi Fin</pre>	<pre>#include&lt;stdio.h&gt; main () {     int t[10];     int tPos[10] ;     int tNeg[10] ;     int i ;     int pos = -1 ;     int neg = -1;     printf("entrer les valeurs du tableau \n");     for(i=0; i&lt;10; i++)     {         scanf("%d",&amp;t[i]);     }     for(i=0; i&lt;10; i++)     {         if (t[i]&gt;0)         {             pos = pos +1 ;             tPos[pos] = t[i];         }         if (t[i]&lt;0)         {             neg = neg +1 ;             tNeg[neg] = t[i];         }     }     if(pos == -1)         printf("le tableau des nombres positifs est vide \n");     else     {         printf("le tableau des nombres positifs contient: \n");         for (i=0; i&lt;=pos; i++)             printf("%d \n",tPos[i]);     }     if(neg == -1)         printf("le tableau des nombres négatifs est vide \n");     else     {         printf("le tableau des nombres négatifs contient: \n");         for (i=0; i&lt;=neg; i++)             printf("%d \n",tNeg[i]);     } }</pre>

## Solution Exo5 :

En Algorithmique	En C
<b>Algo Matrice</b>  <b>Var M</b> : tableau [0..4, 0..4] de entier  <b>i,j</b> : entier  <b>Debut</b>  <b>Ecrire</b> ('entrer les valeurs de la matrice ')  <b>Pour</b> (i de 0 à 4 ) faire <b>Pour</b> (j de 0 à 4) faire <b>Lire</b> (M[i,j]) <b>FinPour</b> <b>FinPour</b>  <b>Pour</b> (i de 0 à 4 ) faire <b>Pour</b> (j de 0 à 4) faire <b>Si</b> (i = j) <b>M</b> [i,j] = 0 <b>FinPour</b> <b>FinPour</b>  <b>Pour</b> (i de 0 à 4 ) faire <b>Pour</b> (j de 0 à 4) faire <b>ecrire</b> (M[i,j]) <b>FinPour</b> <b>FinPour</b>  <b>Fin</b>	<pre>#include&lt;stdio.h&gt; main () {     int t[5][5];      int i,j ;      printf("entrer les valeurs de la matrice \n");      for(i=0; i&lt;5; i++)     {         for (j=0; j&lt;5; j++)         {             scanf("%d",&amp;(t[i][j]));         }     }      for(i=0; i&lt;5; i++)     {         for (j=0; j&lt;5; j++)         {             if (i==j)                 t[i][j]= 0;         }     }      printf(" la matrice devient : \n");      for(i=0; i&lt;5; i++)     {          for (j=0; j&lt;5; j++)         {             printf("% d ",(t[i][j]));         }         printf(" \n"); // pour sauter la ligne     } }</pre>

## Solution Exo 6 :

En algorithmique	En C
<pre>Algo Exo6_S4 Var T1 : tableau [0..29] de réel     T2 : tableau [0..29] de réel     T3 : tableau [0..29] de réel     i, som: entier Début Ecrire ('entrer les 30 éléments du premier tableau') Pour (i allant de 0 à 29 ) faire     Lire (T1[i]) FinPour Ecrire ('entrer les 30 éléments du deuxième tableau') Pour (i allant de 0 à 29 ) faire     Lire (T2[i]) FinPour  Pour (i allant de 0 à 29 ) faire     T3[i] ← T1[i] + T2[i] Ecrire (T3[i]) FinPour  som ← 0 Pour (i allant de 0 à 29 ) faire     som ← som + (T1[i] * T2[i]) FinPour Ecrire ('le produit scalaire des deux vecteur est ', some) Fin</pre>	<pre>#include &lt;stdio.h&gt; main() {     float t1[30];     float t2[30];     float t3[30];     int i ;     float som;     printf("entrer les valeurs du premier tableau \n");     for(i = 0 ; i&lt; 30 ; i++)     {         scanf ("%f",&amp;t1[i]);     }     printf("entrer les valeurs du deuxième tableau \n");     for(i = 0 ; i&lt; 30 ; i++)     {         scanf ("%f",&amp;t2[i]);     }     printf(" les valeurs du troisième tableau \n ");     for(i = 0 ; i&lt; 30 ; i++)     {         t3[i] = t1[i] + t2[i];         printf ("%f \n",t3[i]);     }     som = 0;     for(i = 0 ; i&lt; 30 ; i++)     {         som = som + (t1[i] * t2[i])    ;     }     printf ("le produit scalaire des deux vecteur est %f \n",som); }</pre>