

TP N°2

Exercice 1 :

```

Program  somme ;
|   Const  m=10 ;
|   Var  T : array [1..m] of integer ;
|       i , s: integer;
|
| Begin
|   s := 0 ;
|   for i :=1 to m do
|   [   readln (T[i] ) ;
|
|   for i :=1 to m do
|   [   s := s+T[i];
|
|   writeln ( ‘ la somme = ’ , s) ;
|
| end.
    
```

Exercice 2 :

```

program  Table ;
|   Const m=10 ;
|   Var  T1, T2, T3 : array [1..m] of integer ;
|       i : integer ;
|
| Begin
|   writeln (‘Entrez les éléments du premier tableau : ‘) ;
|   for i := 1 to m do
|   [   readln (T1[i]) ;
|
|   writeln (‘Entrez les éléments du deuxième tableau : ‘) ;
|   for i := 1 to m do
|   [   readln (T2[i]) ;
|
|   for i := 1 to m do
|   [   T3[i] := T1[i] + T2[i] ;
|
|   for i := 1 to m do
|   [   writeln (T3[i] ) ;
|
| end.
    
```

TP N°2

Exercice 3 :

```

Program    Positif_negatif;

    Const m=10 ;

    Var      T : array [1..m] of integer ;
             i, nn ,np : integer;

Begin
    for i:=1 to m do
    [   readln(T[i] ) ;
    ]
    nn:=0 ; np:=0;

    for i:=1 to m do
    begin
    |   if ( t[i] > 0 ) then  np:=np+1;
    |   if ( t[i] <=0 ) then  nn:=nn+1;
    end;

    writeln (nn , np ) ;

End.
    
```

Exercice 4 :

```

Algorithme    somme ;

    Constantes  m=20 ; n=30 ;
    Variables   T : Tableau [1.. m,1.. n ] de Réel
                i , j, n, m, Som : Entiers

DEBUT
    Som ← 0
    POUR i = 1 à m faire
    |   POUR j = 1 à n faire
    | |   Lire (T[i, j])
    | |   finp
    |   finP

    POUR i = 1 à m faire
    |   POUR j = 1 à n faire
    | |   Som ← Som + T[i, j]
    | |   finp
    |   finp

    Ecrire ('La somme est : ' , Som)

FIN.
    
```

TP N°2

Exercice 5 :

Algorithme addition

Constantes $n=10$; $M=20$

Variables A,B,C : Tableau [$1..m,1..n$] de **Réel**
 i, j, N, M : **Entiers**

DEBUT

POUR $i = 1$ à m **faire**

POUR $j = 1$ à n **faire**

 Lire ($A[i, j]$)

FINP

FINP

POUR $i = 1$ à m **faire**

POUR $j = 1$ à n **faire**

 Lire ($B[i, j]$)

FINP

FINP

POUR $i = 1$ à m **faire**

POUR $j = 1$ à n **faire**

$C[i, j] \leftarrow A[i, j] + B[i, j]$

FINP

FINP

POUR $i = 1$ à m **faire**

POUR $j = 1$ à n **faire**

 Ecrire ($C[i, j]$)

FINP

FINP

FIN.