

M13 Algorithmique et structure de données**Correction Travaux Dirigés 3**

(Structures répétitives)

Exercice 1:

Ecrire un algorithme qui lit un entier et calcul et affiche la somme des chiffres qui le constitue.

Application:

Donner un entier:2591

Somme(2591) = 17

```

Algorithme      sommeChiffresEntier
Variables N, NN, S, r : Entier
Début
    Ecrire("Donner un entier:")
    Lire(N)
    NN ← N      #Ici on copie N dans NN pour le garder à l'affichage
    S ← 0
    TANT QUE (N!=0) FAIRE
        r ← N%10
        S ← S + r
        N ← N//10
    FINTQ
    Ecrire("Somme(", NN, ")=", S)
FIN
  
```

Exercice 2:

Ecrire un algorithme qui permet de calculer et d'afficher le nombre d'occurrences d'un chiffre (0 ≤ chiffre < 10) dans un nombre entier.

Application:

Donner un entier:12117

Donner le chiffre:1

Occurrences(1) = 3

```

Algorithme      occurChiffresEntier
Variables N, chif, oc : Entier
Début
    Ecrire("Donner un entier:")
    Lire(N)
    Ecrire("Donner le chiffre:")
    Lire(ch)
    oc ← 0
    TANT QUE (N!=0) FAIRE
        SI(ch=N%10) ALORS
            oc ← oc + 1
        FINSI
        N ← N//10
    FINTQ
    Ecrire("Occurrences(", chif, ")=", oc)
FIN
  
```

M13 Algorithmique et structure de données**Exercice 3:**

Ecrire un algorithme qui lit un entier et calcul et affiche son miroir.

Application:

Donner un entier:873352

Miroir(873352) = 253378

```

Algorithme      miroirEntier
Variables N, mir, r, NN : Entier
Début
    Ecrire("Donner un entier:")
    Lire(N)
    NN ← N
    mir ← 0
    TANT QUE (N!=0) FAIRE
        r ← N%10
        mir ← mir*10 + r
        N ← N//10
    FINTQ
    Ecrire("miroir(",NN,")=",mir)
FIN
  
```

Exercice 4:

Ecrire un algorithme qui permet de convertir un nombre écrit en base binaire vers la base décimale.

Application:

Donner un nombre binaire:10101

Decimal(10101)=21

```

Algorithme      Binairedecimal
Variables      N, NN, dec, r, i : Entier
Début
    Ecrire("Donner un nombre binaire:")
    Lire(N)
    NN ← N
    dec ← 0
    i ← 0
    TANT QUE (N!=0) FAIRE
        r ← n%10
        dec ← dec+r*2**i
        N ← N//10
        i ← i+1
    FinTQ
    Ecrire("Decimal(",NN,")=",dec)
Fin
  
```

M13 Algorithmique et structure de données**Exercice 5:**

Ecrire un algorithme qui permet de convertir un nombre écrit en base décimale vers la base binaire.

Application:

Donner un nombre decimal:958

Binaire(958) = 1110111110

```

Algorithme      Decimalbinaire
Variables      N, NN, bin, r, i : Entier
Début
    Ecrire("Donner un nombre décimal:")
    Lire(N)
    NN←N
    bin←0
    i←0
    TANT QUE (N!=0) FAIRE
        r←n%2
        bin←bin+r*10**i
        N ←N//2
        i←i+1
    FinTQ
    Ecrire("Decimal(",NN,")=",bin)
Fin
  
```

Exercice 6:

Ecrire un algorithme qui calcul le PPCM (plus petit multiple commun) de deux nombres entiers.

Application:

Donner deux entiers:12 9

PPCM(12,9)=36

```

Algorithme      PPCM
Variables      a, b, ma, mb : Entier
#ma est le multipliant de a et mb celui b
Début
    Ecrire("Donner deux entiers:")
    Lire(a,b)
    ma ← a    #On garde a et b pour l'affichage
    mb ← b
    TANT QUE (ma!=mb) FAIRE
        si (ma<mb) ALORS
            ma← ma+a
        SINON
            mb← mb+b
        FINSI
    FinTantQue
    Ecrire("PPCM(",a," ",b,")=",ma)    #ou mb
Fin
  
```

M13 Algorithmique et structure de données**Exercice 7:**

Ecrire un algorithme qui réalise les tâches suivantes :

- Lire successivement une suite de nombres entiers.
- La saisie de 0 entraîne la fin de la lecture.
- A la fin on affiche le plus grand élément, le plus petit élément et la somme des éléments de la suite.

NB! +INF est une constante algorithmique qui désigne $+\infty$ des entiers.
(voir $-\text{INF} \equiv -\infty$)

Saisir les nombres :

2

-5

8

4

0

Max : 8**Min : -5****Somme : 9**

```
Algorithme      minmaxEntiers
Variables      n, max, min, S : Entier
Début
    min ← +INF
    max ← -INF
    Ecrire("Saisir les nombres:")
    Lire(n)
    S ← 0
    TANT QUE (n != 0) FAIRE
        S ← 0
        si (n < min) ALORS
            min ← n
        FINSI
        si (max < n) ALORS
            max ← n
        FINSI
        Lire(n)
    FinTQ
    Ecrire("Max:", max)
    Ecrire("Min:", min)
    Ecrire("Somme:", S)
Fin
```