

## Corrigé série 1

### **Solution 1 :**

#### **a) Algorithme A ;**

Var a, b, c : entier ; *// déclaration des variables*

Début

$a \leftarrow 5$  ; *// affectation de la valeur 5 à la variable a*

$b \leftarrow a+1$  ; *// affectation du résultat de a+1 à la variable b*

$c \leftarrow a+b$  ; *// affectation de la somme de a et b à la variable c*

Fin

**b)** Les commentaires dans un algorithme (ou programme) sont ajoutés à titre indicatif pour le rendre plus lisible et compréhensible mais ils ne sont pas obligatoires.

**c)** Le résultat du déroulement de cet algorithme est comme suit :

$a=5$

$b=5+1=6$

$c=6+5=11$

### **Solution 2 :**

**a)** L'algorithme calcule la somme et le produit de deux entiers

**b)**  $C=5+6=11$

**c)**  $D=5 \times 6=30$

### **Solution 3 :**

#### **a) Algorithme C ;**

Var x,y :entier ;

Const z =10 ;

Début

Ecrire ("donner x, y") ;

Lire (x,y) ;

$x \leftarrow y+1$  ;

~~$z \leftarrow z+x$~~  ; (z est une constante)

Ecrire ('x,z') ;

Fin

**b)** Déroulement de l'algorithme pour  $x=9$  et  $y=3$

$x=3+1=4$

Affichage :  $x=4$ ,  $z=10$

**Solution 4 :**

**a)**

```
Algorithme D ;  
Var a, b, c : entier ;  
Début  
    Ecrire ("donner a et b") ;  
    Lire (a, b) ;  
    c ← a ;  
    a ← b ;  
    b ← c ;  
    Ecrire (a, b) ;  
Fin
```

**b)** Déroulement de l'algorithme pour a=5 et b=7

```
c=5  
a=7  
b=5  
Affichage : a=7 et b=5
```

Donc, l'algorithme réalise la permutation de a et b.

**Solution 5 :**

s : variable déclarée sans être utilisée

m : doit être de type réel

Affichage : 22.5 ; 35

**Solution 6 :**

**a)** val=1000 ; double=2000 ; triple=3000

**b)** Simplification :

```
Var val : entier ;  
Début  
    val ← 1000 ;  
    Ecrire (val, val*2, val*3) ;  
Fin
```

**Solution 7 :**

**a)** Algorithme Un\_Bonjour ;

Début

Ecrire ("Bonjour à tous") ;

Fin

Cet algorithme n'a qu'une sortie, le message « Bonjour à tous ».

**b)** Pour manipuler les entrées et les sorties d'un algorithme, on utilise respectivement les deux instructions : Lire() et Ecrire().

**c)** Algorithme Un\_ReBonjour ;

Var mge : chaîne ;

Début

Ecrire ("Veuillez saisir un message :)") ;

Lire (mge) ;

Ecrire (mge) ;

Fin

## Corrigé série 2

### **Solution 1 :**

a) Algorithme carré ;

Var nb, carr : réel ;

Début

Lire (nb) ;

carr  $\leftarrow$  nb \* nb ; // ou carr  $\leftarrow$  nb<sup>2</sup>

Ecrire (carr) ;

Fin

b) Algorithme cube ;

Var nb, cub : réel ;

Début

Lire (nb) ;

cub  $\leftarrow$  cub ^ 3

Ecrire (cub) ;

Fin

Algorithme inverse ;

Var nb, inv : réel ;

Début

Lire (nb) ;

inv  $\leftarrow$  1/ nb ; // tel que nb  $\neq$  0

Ecrire (inv) ;

Fin

### **Solution 2 :**

a. Algorithme moyenneA ;

Var n1, n2, n3, moy : réel ;

Début

Ecrire ("Donner trois notes :") ; // ce message est optionnel

Lire (n1, n2, n3) ;

moy  $\leftarrow$  (n1+n2+n3) / 3 ;

Ecrire (moy) ;

Fin

**b. Algorithme moyenneB ;**

```
Var n1, n2, n3, moy : réel ; c1, c2, c3 : entier ;
Début
  Lire (n1, n2, n3) ;
  Lire (c1, c2, c3) ;
  moy ← (n1*c1+n2*c2+n3*c3) / (c1+c2+c3) ;
  Ecrire (moy) ;
Fin
```

**Solution 3 :**

Algorithme permutation ;

Var a,b,c : réel ;

Début

Ecrire ("Entrer la valeur de a :") ;

Lire (a) ;

Ecrire ("Entrer la valeur de b :") ;

Lire (b) ;

Ecrire ("a=",a) ;

Ecrire ("b=",b) ;

c ← a ;

a ← b ;

b ← c ;

*La permutation est réalisée à travers ces trois affectations.  
La variable « c » est une variable temporaire qui doit être  
de même type que a et b.*

Ecrire ("a=",a) ;

Ecrire ("b=",b) ;

Fin

**Solution 4 :**

Algorithme permutation2 ;

Var x, y : réel ;

Début

Lire (x,y) ;

x ← x + y ;

y ← x - y ;

x ← x - y ;

Ecrire (x,y) ;

Fin

### **Corrigé série 3**

**Solution 1 :** (L'écriture des entêtes d'algorithmes est omise dans quelques solutions)

```
Var n : réel ;  
Début  
  Ecrire ("Entrez un nombre : ") ;  
  Lire (n) ;  
  Si (n < 0) Alors  n ← -n ;  
  FinSi  
  Ecrire ("la valeur absolue est", n) ;  
Fin
```

**Solution 2 :**

```
Var n : entier ;  
Début  
  Ecrire ("Entrez un nombre : ") ;  
  Lire (n) ;  
  Si (n mod 2 = 0) Alors  Ecrire ("Ce nombre est pair") ;  
  Sinon  Ecrire ("Ce nombre est impair") ;  
  FinSi  
Fin
```

**Solution 3 :**

```
Var a, b, c : caractère ;  
Début  
  Ecrire ("Entrez successivement trois lettres : ") ;  
  Lire (a, b, c) ;  
  Si (a < b ET b < c) Alors  Ecrire ("Les lettres sont classées alphabétiquement") ;  
  Sinon  Ecrire ("Les lettres ne sont pas classées") ;  
  FinSi  
Fin
```

**Solution 4 :**

**a)**     Var a, b, c, max ;  
Début  
 Ecrire ("Entrez trois nombres réels : ") ;  
 Lire (a, b, c) ;

## **Corrigé série 4**

### **Solution 1 :**

#### **Algorithme 1 :**

```
Bonjour
2
Bonjour
3
Bonjour
4
Bonjour
5
Bonjour
6
Bonjour
7
Bonjour
8
Fin
```

#### **Algorithme 2 :**

```
Bonjour
Fin
```

#### **Algorithme 3 :**

```
Fin
```

### **Solution 2 :**

```
Var p, s : réel ;
Début
    s ← 0 ;
    Répéter
        Ecrire ("Entrer le prix de l'article (0 si fin):");
        Lire (p) ;
        s ← s + p ;
    Jusqu'à (p = 0)
    Ecrire (" La somme des prix des articles est ", s) ;
Fin
```

Puisque le nombre d'articles n'est pas connu à l'avance ainsi qu'il faut faire au moins une saisie pour terminer, la boucle « Répéter » est appliquée dans ce cas.

**Solution 3 :**

a) Var N, i : Entier ;

Début

Ecrire ("Entrer un entier : ") ; Lire (N) ;

Ecrire ("Les 10 nombres suivants sont : ")

Pour i de (N + 1) à (N + 10)

Ecrire (i) ; /\* Si (i mod 2 = 0) Ecrire i ; \*/

FinPour

Fin

b) Pour le deuxième cas, il suffit de remplacer l'instruction : « Ecrire (i) » par

« Si (i mod 2 = 0) Alors Ecrire i » dans la 6<sup>ème</sup> ligne.

**Solution 4 :** (les instructions correspondantes à la question b sont mises en commentaires)

Var N, i, som : Entier ;

Début

Ecrire ("Donner un entier : ") ;

Lire (N) ; som  $\leftarrow$  0 ; // *prod*  $\leftarrow$  1 ;

Pour i de 1 à N

som  $\leftarrow$  som + i ; // *prod*  $\leftarrow$  *prod* \* i ;

FinPour

Ecrire ("La somme = ", som) ; // *Ecrire ("Le produit = ", prod) ;*

Fin



**Solution 5 :**

```
Var N, i, F : entier ;
Début
  Ecrire ("Entrer un entier positif : ") ; Lire (N) ;
  Si (N<0) Alors Ecrire ("Erreur !") ;
  Sinon
    F ← 1 ;
    Si (N>1) Alors
      Pour i de 2 à N
        F ← F * i ;
      FinPour
    Fsi
    Ecrire ("La factorielle est ", F) ;
  Fsi
Fin
```

```
#include<stdio.h>
int main()
{
  int N,i,F;
  printf("Entrer un entier positif : \n");
  scanf("%d",&N);
  if(N<0)
  {
    printf("Erreur !") ;
  }
  else
  {
    F=1;
    if(N>1)
    {
      for(i=2;i<=N;i++)
      {
        F=F*i;
      }
    }
    printf("La factorielle est %d:\t", F) ;
  }
}
```

```

    Pour i de 2 à N
        F ← F * i ;
    FinPour
Fsi
    Ecrire ("La factorielle est ", F) ;
Fsi
Fin

```

### Solution 6 :

#### a) et b)

```

Var N, i, min, pmin : Entier ;
Début
    min ← 0 ; // juste pour initialiser la var
    Pour i de 1 à 10
        Ecrire ("Entrer le nombre numéro ", i) ; Lire (N) ;
        Si (i = 1 ou N < min) Alors min ← N ; pmin ← i ;
    FinSi
    FinPour
    Ecrire ("Le minimum est : ", min) ;
    Ecrire ("Il a été saisi en position :", pmin) ;
Fin

```

#### c/

```

Var N, i, min, pmin : Entier ;
Début
    i ← 1 ; min ← 0 ;
    Répéter
        Ecrire ("Entrer le nombre numéro ", i) ; Lire (N) ;
        Si (i = 1 ou N < min) Alors
            min ← N ; pmin ← i ;
        FinSi
        i ← i + 1 ;
    Jusqu'à (N=0)
    Ecrire ("Le minimum est ", min),
    Ecrire ("Il a été saisi en position numéro ", pmin),
Fin

```

### Solution 7 :

Algorithme nombre\_premier

Var i, N : entier ;

    x : booleen;

Début

    Ecrire ("entrer N"); Lire(N);

    x ← faux;

    i ← 2;

    Tant que (i < N et x = faux) Faire

        Si (N mod i = 0) alors

            Ecrire ("le nombre n'est pas premier") ;

            x ← vrai ;

            i ← i+1;

        Finsi

    Fin Tq

    Si (x=faux) alors Ecrire ("le nombre est premier") ;

Fin

### Solution 8 :

a) et b)

Algorithme codePin ;

Var codePin, monCode, tentative: Entier ;

Début

    codePin ← 5454 ;

    tentative ← 3 ;

    Répéter

        Ecrire ("Entrez le code PIN :)"); Lire (monCode) ;

        tentative ← tentative – 1 ;

        Si (monCode ≠ codePin) Alors Ecrire ("Code incorrect") ; FinSi

    Jusqu'à (monCode = codePin ou tentative=0)

    Si (tentative=0) Alors

        Ecrire ("Vous ne pouvez plus saisir de code") ;

    Sinon

        Ecrire ("Bienvenue") ;

    FinSi

Fin

**Corrigé série 5****Solution 1 :****Algorithme algo ;**

Var i : entier ;

Tableau T[10] : entier ;

Début

Pour i de 0 à 9

Ecrire ("Donner un entier : ") ; Lire T[i] ;

FinPour

Pour i de 0 à 9

Si (T[i] mod 2  $\neq$  0) Alors Ecrire T[i] ;

FinPour

Fin

**Algorithme b ;**

Var i, n: entier ;

Tableau T[n] : entier ;

Début

Ecrire ("Donner le nombre de valeurs à saisir ") ; Lire (n) ;

Pour i de 0 à n-1

Ecrire ("Donner un entier : ") ; Lire T[i] ;

FinPour

Pour i de 0 à n-1

Si T[i] mod 2  $\neq$  0 Alors Ecrire T[i] ;

FinPour

Fin

**Algorithme\_c ;**

Var i, n, j, k: entier ;

Tableau T[n], Ti[n], Tp[n] : entier ; // on suppose par défaut que les trois tableaux ont la  
même taille

Début

Ecrire ("Donner le nombre de valeurs à saisir ") ; Lire (n) ;

Pour i de 0 à n-1

Ecrire ("Donner un entier : ") ; Lire T[i] ;

FinPour

```
j ← 0 ; k ← 0 ;
Pour i de 0 à n-1
    Si (T[i] mod 2 <> 0) Alors Ti[j] ← T[i] ; j ← j+1 ;
    Sinon Tp[k] ← T[i] ; k ← k+1 ;
FinPour
Pour i de 0 à j-1
    Ecrire Ti[i] ;
FinPour
Pour i de 0 à k-1
    Ecrire Tp[i] ;
FinPour
Fin
```

**Solution 2:****a)**

```
Algorithme Max_tableau1D
Var i, pos: entier ; max : réel ;
Tableau A [10]: réel ;
Début
    Pour i de 0 à 9
        Ecrire ("Entrer un nombre:"); Lire (A[i]) ;
        Si (i = 0 OU max < A[i]) Alors
            max ← A[i]; pos ← i ;
        Finsi
    Finpour
    Ecrire ("Le maximum du tableau est :", max);
    Ecrire ("Il se trouve à la position", pos+1);
Fin
```

**b)**

```
Algorithme Max_tableau2D
Var i, j, pos: entier ; max : réel ;
Tableau A[5] [10]: réel ;
Début
    Pour i de 0 à 4
        Pour j de 0 à 9
            Ecrire ("Entrer un nombre:");
```

```
        Lire (A[i][j]) ;
        Si (i = 0 ET j = 0 OU max < A[i][j]) Alors
            max ← A[i][j] ; lin ← i ; col ← j ;
        Finsi
    Finpour
Finpour

Ecrire ("Le maximum du tableau est :", max);
Ecrire ("Il est positionné à la ligne", lin+1, "et la colonne", col+1);

Fin
```

**Solution 3 :**

```
Algorithme Recherche_tableau2D
Var i, j, n, m: entier ; v : réel ; drap : booléen ;
Tableau A[n][m]: réel ;
Début
    Ecrire ("Donner le nombre de lignes:"); Lire(n) ;
    Ecrire ("Donner le nombre de colonnes:"); Lire(m) ;
    Pour i de 0 à n-1
        Pour j de 0 à m-1
            Ecrire ("Entrer un nombre:"); Lire (A[i][j]) ;
        Finpour
    Finpour
    Ecrire ("Donner la valeur recherchée:"); Lire(v) ;
    drap ← Faux ; i ← 0 ; j ← 0 ;
    Tant que (i < n ET drap = Faux) faire
        Tant que (j < m ET drap = Faux) faire
            Si (A[i][j] = v) Alors drap ← Vrai ;
        Finsi
    FinTantque
    FinTantque
    Si (drap = Vrai) Alors
        Ecrire ("La valeur recherchée existe dans le tableau");
    Sinon
        Ecrire ("La valeur recherchée n'existe pas dans le tableau");
    Finsi
Fin
```

**Solution 4 :**

```
Algorithme Occurence_tableau2D
Var i, j, n, m, compt: entier ; v : réel ;
Tableau A[n][m]: réel ;
Début
    Ecrire ("Donner le nombre de lignes:"); Lire(n) ;
    Ecrire ("Donner le nombre de colonnes:"); Lire(m) ;
    Pour i de 0 à n-1
        Pour j de 0 à m-1
            Ecrire ("Entrer un nombre:");
            Lire (A[i][j]) ;
        Finpour
    Finpour
    Ecrire ("Donner une valeur:"); Lire(v) ;
    compt ← 0 ; i ← 0 ; j ← 0 ;
    Pour i de 0 à n-1
        Pour j de 0 à m-1
            Si (A[i][j]=v) Alors compt ← compt+1 ;
        Finsi
    Finpour
    Finpour
    Ecrire ("Le nombre d'occurrences de la valeur donnée est :", compt);
Fin
```

**Solution 5 :**

```
Algorithme Calcul_rationnel ;
Structure Rationnel // Définition de la structure d'un nombre rationnel
    numérateur, dénominateur: entier;
FinStructure ;
Var p, q, r : Structure Rationnel ;
Début
    /* Saisie */
    Ecrire ("Entrez le numérateur et le dénominateur du premier nombre : ");
    Lire (p.numérateur, p.dénominateur);

    Ecrire ("Entrez le numérateur et le dénominateur du deuxième nombre : ");
    Lire (q.numérateur, q.dénominateur);
```

```
/* Affichage */
Ecrire (p.numerateur, p.denominateur);
Ecrire (q.numerateur, q.denominateur);
/* Multiplication*/
r.numerateur ← p.numerateur * q.numerateur ;
r.denominateur ← p.denominateur * q.denominateur ;

Ecrire (r.numerateur, r.denominateur);

/* Addition*/
r.numerateur ← p.numerateur * q.denominateur + q.numerateur * p.denominateur;
r.denominateur ← p.denominateur * q.denominateur;
Ecrire (r.numerateur, r.denominateur);

Fin
```

**Solution 6 :**

```
Algorithme Tableau_comptes ;
Structure Date
    jour, mois, annee : entier ;
FinStructure ;
Structure Compte
    num: entier;
    nom, pnom: chaine ;
    dtOuvert : Date ;
FinStructure ;
Var i , n : entier ;
Tableau C[n] : Structure Compte ;
Début
    Ecrire ("Nombre des comptes : ") ;
    Lire (n) ;
    Pour i de 0 à n-1
        Ecrire ("No du compte: ") ;
        Lire(C[i].num) ;
        Ecrire ("Nom et prénom du client : ") ;
        Lire(C[i].nom, C[i].pnom) ;
        Ecrire ("Date de naissance du client : ") ;
        Lire(C[i].dtOuvert.jour, C[i].dtOuvert.mois, C[i].dtOuvert.annee) ;
    Finpour
```



Pour i de 0 à n-1

Ecrire ("N°:" C[i].num, "Nom et prénom:", C[i].nom,  
"Date de naissance:", C[i].dtOuvert.jour, C[i].dtOuvert.mois,  
C[i].dtOuvert.annee) ;

Finpour

Fin