Correction de l'examen 02

Exercice 01 : Questions de cours

1. Les identificateurs valides sont :

- Prix_achat
- calcul_du_prix_total.

2. Différence entre une procédure et une fonction

- La différence entre une procédure et une fonction, c'est que la deuxième retourne une seule valeur tandis que la première retourne plusieurs valeurs (pas une) ou aucune valeur.
- Une macro VBA pour Excel est une procédure sans paramètres. Une procédure est une série d'instructions effectuant une tache bien déterminée. Une macro sert à automatiser les taches répétitives.

3. Les instructions erronées

- 4 ← 8 car 4 n'est pas une variable
- i + 1 ← i. L'instruction qui est juste : i ← i + 1
- i ←2,5. L'instruction qui est juste est : i ← 2.5

4. Différence entre le OU logique et le OUex (ou exclusif)

Le OU logique est défini de la manière suivante : a OU b est VRAI si et seulement si a est VRAI ou b est VRAI. Si a est vrai et que b est vrai aussi, alors a OU b est vrai. Le tableau suivant présente la table de vérité de OU :

a	b	a OU b
0	0	0
0	1	1
1	0	1
1	1	1

Table de vérité de OU

La fonction OU exclusif, souvent appelée XOR (eXclusive OR), est un opérateur logique de l'algèbre de Boole. À deux opérandes, qui

Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

peuvent avoir chacun la valeur VRAI ou FAUX, il associe un résultat qui a lui-même la valeur VRAI seulement si les deux opérandes ont des valeurs distinctes. Le tableau suivant présente la table de vérité de OUex :

a	b	a OUex b
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité de OUex

Comme on peut le voir, l'opérateur logique OUex (OU exclusif) peut se définir par les deux phrases suivantes qui sont équivalentes :
Le résultat est VRAI si un et un seul des opérandes a et b est VRAI
ou
Le résultat est VRAI si les deux opérandes a et b ont des valeurs distinctes

Exercice 02 : Affichage à l'écran

Instructions	Affichage à l'écran
1	12
2	la valeur de X est :
3	La valeur de T[2] est : 8

Exercice 03 : Service photocopie

Algorithme :

Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

```
Algorithme Facture
Variables Nombre : Entier
          Prix : Réel
Début
   Ecrire ("Donner le nombre de photocopies : ")
   Lire(Nombre)
   Si Nombre <= 10 Alors
      Prix ← Nombre * 0.50
   SinonSi Nombre <= 30 Alors
      Prix ← 10 * 0.50 + (Nombre - 10) * 0.40
   Sinon
      Prix ← 10 * 0.50  + 20 * 0.40 + (Nombre - 30) * 0.30
   FinSi
   Ecrire("Le prix total est: ", Prix, " Dh")
Fin
```

Exercice 01 : Fonction Signe

1. La fonction Signe

```
' Fonction Signe
Fonction Signe(x : réel) : réel
Début
   SI x > 0 alors
      Retourner 1
   SinonSI x = 0 alors
      Retourner 0
   Sinon
      Retourner -1
   FinSI
FinFonct
```

2. Procédure qui appelle la fonction Signe

Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

```
' Procédure qui appelle la fonction Signe
Procédure Appel_Signe()
Variable y : réel
Début
   Ecrire("Donner la valeur de y : ")
   Lire(y)
   Ecrire("Signe(", y, ") =  ", Signe(y))
FinProc
```

Exercice 05 : Création et affichage d'un tableau à 2 dimensions

Algorithme :

```
Algorithme tableau_2dimensions_création _affichage
// Déclaration du tableau t
Variables t : tableau[3,5] d'entiers
          i, j, k : entiers
Début
   // Initialisation de variable k à 1
   k ← 1
   // Création du tableau t
   Pour i ← 1 jusqu'à 3 faire
      Pour j ← 1 jusqu'à 5 faire
         t[i,j] ← k
         k ← k + 1
      FinPour
   FinPour
   // Affichage du tableau t
   Pour i ← 1 jusqu'à 3 faire
      Pour j ← 1 jusqu'à 5 faire
         Ecrire("t[", i , ", ", j, "] = "  , t[i,j])
      FinPour
   FinPour
Fin
```


Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

Exercice 06 : PPCM

Le PPCM de deux entiers strictement positifs est défini comme étant l'entier positif le plus petit qui soit un multiple de ces deux entiers. Par exemple, PPCM(15, 20) = 60, PPCM(14, 14) = 14, PPCM(2, 4) = 4, PPCM(5, 9) = 45 etc.. Si l'un des deux entiers vaut 0, le PPCM est 0.

Solution 1 :

Pour rechercher le PPCM de deux entiers n1 et n2 strictement positifs, on suit les étapes suivantes :

1. soit p le maximum de n1 et n2 ;
2. vérifier si p est divisible par n1 et n2 ;
3. si oui, PPCM = p et l'algorithme est terminé ;
4. sinon, incrémenter p et reprendre à l'étape 2.

Fonction PPCM :

```
fonction PPCM(n1 : entier, n2: entier) : entier
Variables p: entier
          Trouve: logique
Début
   Si m1 < n2 alors
      p ← n2
   Sinon
      p ← n1
   FinSi
   Trouve ← faux
   TantQue Trouve = faux faire
      Si (p mod n1 = 0) et (p mod n2 = 0) alors
         Trouve ← vrai
      FinSi
      Si Trouve = faux alors
         p ←p + 1
      FinSi
   FinTantQue
   Retourner p
FinFonct
```


Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

Algorithme qui intègre la fonction PPCM :

```
Algorithme calcul_ppcm
Variables m1,  m2 : entiers

// Déclaration de la fonction PPCM
fonction PPCM(n1 : entier, n2: entier) : entier
Variables p: entier
            Trouve: logique
Début
   Si m1 < n2 alors
      p ← n2
   Sinon
      p ← n1
   FinSi
   Trouve ← faux
   TantQue Trouve = faux faire
      Si (p mod n1 = 0) et (p mod n2 = 0) alors
         Trouve ← vrai
      FinSi
      Si Trouve = faux alors
         p ←p + 1
      FinSi
   FinTantQue
   Retourner p

FinFonct

// Partie principale
Début
   Ecrire("Donnez un entier strictement positif m1 : ")
   Lire(m1)
   Ecrire("Donnez un entier strictement positif m2 : ")
   Lire(m2)
   Ecrire(" Le PPCM de ", m1, " et ", m2, " est : ", PPCM(m1, m2)
Fin
```

Solution 2 :

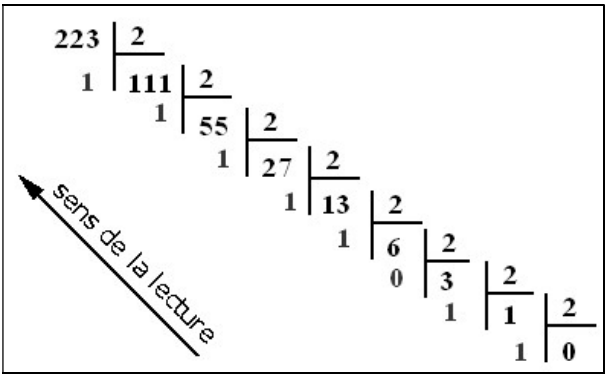
Pour chercher un multiple commun de n1 et n2 avec n1 > n2, on cherche parmi les multiples du plus grand. Le premier multiple qui est divisible par n2 est le PPCM de n1 et n2.

```
fonction PPCM(n1 : entier, n2: entier): entier
Variables p : entier
Début
   p ← n1
   TantQue  p mod n2 ≠ 0 faire   //  n2 ne devise pas P
      p ← p + n1
   FinTanQue
   Retourner p
FinFonct
```

Exercice 07 : Conversion d'un entier naturel en base 2

Pour obtenir l'expression binaire d'un nombre exprimé en décimal, il suffit de diviser successivement ce nombre par 2 jusqu'à ce que le quotient obtenu soit égal à 0. Le nombre cherché est donné par les restes successifs des divisions pris du bas vers le haut.

Exemple : conversion du nombre décimal 223 en binaire



$(223)_{10} = (11011111)_2$

Un entier long est codé sur 32 bits, pour cette raison on va utiliser un vecteur de 32 éléments pour stocker les restes successifs des divisons de l'entier par 2.

Prof. Omar El kharki & Mme Jamila Mechbouh / elkharki@gmail.com

Algorithme :

```
Algorithme conversion_en_base_2
Variables : n, i, j : entier
            code : tableau[0 .. 31] d'entiers
Début
   Ecrire("Donner un entier naturel : ")
   Lire(n)
   Pour i ← 0 jusqu'à 31 faire
      Code[i] ← n Mod 2
      n ←  n Div 2
      Si n = 0 alors
         j ← i
         // Sortir de la boucle Pour
         Quitter Pour
      FinSI
   FinPour
   // Affichage du code binaire de l'entier n
   Pour i ← j jusqu'à 0 pas -1 faire
      Ecrire(Code[i])
   FinPour
Fin
```

Note :

Pour plus d'informations sur la conversion entre les systèmes de numération, on vous conseille de consulter le chapitre 3 de l'ouvrage intitulé :
« ***Initiation à l'Informatique & Réseaux*** »
Cours, Ateliers Pratiques & Examens Corrigés
Auteurs : O. El Kharki, & J. Mechbouh