# Correction de l'examen 03

## Exercice 01 : L'équivalent d'une suite d'instructions

Il suffit d'appliquer la loi de Morgan et de changer l'ordre des instructions a ← a + 1 et a ← a – 1.

```
Si a <= b  Et c = False Alors
   a ← a – 1
Sinon
   a ← a + 1
Finsi
```

## Exercice 02 : Concaténation des chaînes des caractères

La valeur de variable c est : "12312"

## Exercice 03 : Structure répétitive

```
Algorithme Boucle
Variable note : réel
Debut
   note ← -1
   TantQue note < 0 ou note > 20 faire
      Ecrire("Entrez une note entre 0 et 20 : ")
      Lire(note)
      Si note < 0 ou note > 20 Alors
         Ecrire( "Saisie erronée. Recommencez")
      FinSi
   FinTantQue
Fin
```


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

## Exercice 04 : Table de multiplication d'un nombre

```
Algorithme Table_de_multiplication
Variable n, i : entiers
Début
   Ecrire("Entrer un nombre n : ")
   Lire(n)
   Ecrire("La table de multiplication de n est : ")
   Pour i ← 0 jusqu'à 10 faire
      Ecrire(n, " * ", i, " = ", n*i)
   FinPour
Fin
```

## Exercice 05 : Fonction

1. La ***Fonction1*** retourne le maximum de deux nombres.
2. La valeur de la variable i est 100 et la valeur de la variable j est 120.

## Exercice 06 : Mot_de_passe

```
Algorithme Mot_de_passe
Variables, Mot_Saisi, MotDePasse : chaînes
Début
   MotDePasse ← "Agadir2010"
   Mot_Saisi ← ""
   TantQue Mot_Saisi <> MotDePasse faire
      Ecrire("Saisir un mot de passe correct : ")
      Lire(Mot_Saisi)
   FinTantQue
   Ecrire("Bienvenue")
Fin
```

## Exercice 07 : Tri d'un tableau

1. Il s'agit de l'algorithme de tri à bulle (Bublle Sort).
2. Voilà Voilà une mise en œuvre simple du tri à bulles sur un


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

tableau d'entiers en C :

```
#define TRUE 1
#define FALSE 0
 void tri_a_bulle(int t[], int const n)
{
   /* Variable de boucle */
   int j   = 0;
   /* Variable de stockage temporaire */
   int tmp = 0 ;
   /* Booléen marquant l'arrêt du tri si le tableau est ordonné */
   int en_desordre = TRUE ;
   /* Boucle de répétition du tri et le test qui arrête le tri dès que le
   tableau est ordonné(en_desordre=FALSE */
   while(en_desordre)
   {
      /* Supposons le tableau ordonné */
      en_desordre = FALSE ;
      /* Vérification des éléments des places j et j+1 */
      for(j = 0 ; j < n- 1 ; j++)
      {
         /* Si les 2 éléments sont mal triés */
         if(t[j] > t[j+1])
         {
            /* Inversion des 2 éléments */
            tmp = t[j+1] ;
            t[j+1] = t[j] ;
            t[j] = tmp ;
            /* Le tableau n'est toujours pas trié */
            en_desordre = TRUE ;
         }
      }
   }
}
```


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