

# TD 1(RAPPEL)

## ENONCE DES EXERCICES

### Exercice 1.1

Écrivez un algorithme remplissant un tableau de 6 sur 13, avec des zéros.

---

### Exercice 1.2

Quel résultat produira cet algorithme ?

```
Tableau X(1, 2) : Entier  
Var  
  
i, j, val : Entier  
Début  
Val ← 1  
Pour i ← 0 à 1  
  Pour j ← 0 à 2  
    X(i, j) ← Val  
    Val ← Val + 1  
  Finpour  
Finpour  
Pour i ← 0 à 1  
  Pour j ← 0 à 2  
    Ecrire( X(i, j))  
  Finpour  
Finpour  
Fin
```

---

### Exercice 1.3

Quel résultat produira cet algorithme ?

```
Tableau X(1, 2) : Entier  
Var  
  
i, j, val : Entier  
Début  
Val ← 1  
Pour i ← 0 à 1  
  Pour j ← 0 à 2  
    X(i, j) ← Val  
    Val ← Val + 1  
  Finpour
```

```
Finpour
Pour j ← 0 à 2
  Pour i ← 0 à 1
    Ecrire( X(i, j))
  Finpour
Finpour
Fin
```

---

### Exercice 1.4

Quel résultat produira cet algorithme ?

**Tableau** T(3, 1) : **Entier**

**Var**

k, m, : **Entier**

**Début**

**Pour** k ← 0 à 3

**Pour** m ← 0 à 1

    T(k, m) ← k + m

  Finpour

Finpour

**Pour** k ← 0 à 3

**Pour** m ← 0 à 1

**Ecrire**( T(k, m))

  Finpour

Finpour

**Fin**

---

### Exercice 1.5

Mêmes questions, en remplaçant la ligne :

T(k, m) ← k + m

par

T(k, m) ← 2 \* k + (m + 1)

puis par :

T(k, m) ← (k + 1) + 4 \* m

---

### Exercice 1.6

Soit un tableau T à deux dimensions (12, 8) préalablement rempli de valeurs numériques.

Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau.

---

### Exercice 1.7

Ecrivez un algorithme qui demande un mot à l'utilisateur et qui affiche à l'écran le nombre de lettres de ce mot (c'est vraiment tout bête).

---

### Exercice 1.8

Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui affiche à l'écran le nombre de mots de cette phrase. On suppose que les mots ne sont séparés que par des espaces (et c'est déjà un petit peu moins bête).

---

### Exercice 1.9

Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui affiche à l'écran le nombre de voyelles contenues dans cette phrase.

On pourra écrire deux solutions. La première déploie une condition composée bien fastidieuse. La deuxième, en utilisant la fonction Trouve, allège considérablement l'algorithme.

---

### Exercice 1.10

Ecrivez un algorithme qui demande une phrase à l'utilisateur. Celui-ci entrera ensuite le rang d'un caractère à supprimer, et la nouvelle phrase doit être affichée (on doit réellement supprimer le caractère dans la variable qui stocke la phrase, et pas uniquement à l'écran).

---

### Exercice 1.11

Considérons les types d'enregistrements suivants :

Type TDate = Enregistrement

    Jour, mois, année : entier ;

    Fin;

TAdresse = Enregistrement

    Numéro : entier ;

    Rue : chaîne [50] ;

    Ville : chaîne [20] ;

    Wilaya : chaîne [20] ;

    Cw : entier ; { Code Wilaya }

    Fin;

THabitant = Enregistrement

    Nom, prenom : chaîne [20] ;

    Date\_naiss : date ;

    Residence : Adresse ;

    Fin;

Ecrire un algorithme permettant de :

- 1- Remplir un tableau T de N habitants ( $N \leq 100$ ).
  - 2- Afficher à partir de T les adresses des habitants nés avant une année de naissance donnée.
  - 3- Afficher les noms et les dates de naissance des habitants de la ville de 'Elhamma' de la wilaya de 'Khenchela'.
  - 4- Editer le nombre d'habitants par wilaya.
-

# TD 1(RAPPEL)

## CORRIGES DES EXERCICES

### Exercice 1.1

**Tableau** Truc(5, 12) : **Entier**

**Debut**

**Pour** i ← 0 à 5

**Pour** j ← 0 à 12

        Truc(i, j) ← 0

**Finpour**

**Finpour**

**Fin**

---

### Exercice 1.2

Cet algorithme remplit un tableau de la manière suivante:

X(0, 0) = 1

X(0, 1) = 2

X(0, 2) = 3

X(1, 0) = 4

X(1, 1) = 5

X(1, 2) = 6

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

---

### Exercice 1.3

Cet algorithme remplit un tableau de la manière suivante:

X(0, 0) = 1

X(1, 0) = 4

X(0, 1) = 2

X(1, 1) = 5

X(0, 2) = 3

X(1, 2) = 6

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

---

### Exercice 1.4

Cet algorithme remplit un tableau de la manière suivante:

```
T(0, 0) = 0
T(0, 1) = 1
T(1, 0) = 1
T(1, 1) = 2
T(2, 0) = 2
T(2, 1) = 3
T(3, 0) = 3
T(3, 1) = 4
```

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

---

### Exercice 1.5

Version a : cet algorithme remplit un tableau de la manière suivante:

```
T(0, 0) = 1
T(0, 1) = 2
T(1, 0) = 3
T(1, 1) = 4
T(2, 0) = 5
T(2, 1) = 6
T(3, 0) = 7
T(3, 1) = 8
```

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

Version b : cet algorithme remplit un tableau de la manière suivante:

```
T(0, 0) = 1
T(0, 1) = 5
T(1, 0) = 2
T(1, 1) = 6
T(2, 0) = 3
T(2, 1) = 7
T(3, 0) = 4
T(3, 1) = 8
```

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

---

### Exercice 1.6

**Var**

```
i, j, iMax, jMax : Entier
Tableau T(12, 8) : Entier
```

Le principe de la recherche dans un tableau à deux dimensions est strictement le même que dans un tableau à une dimension, ce qui ne doit pas nous étonner. La seule chose qui change, c'est qu'ici le balayage requiert deux boucles imbriquées, au lieu d'une seule.

**Debut**

```
...
iMax ← 0
jMax ← 0
Pour i ← 0 à 12
  Pour j ← 0 à 8
    Si T(i, j) > T(iMax, jMax) Alors
      iMax ← i
      jMax ← j
    FinSi
  Finpour
Finpour
Ecrire( "Le plus grand élément est ", T(iMax, jMax))
Ecrire( "Il se trouve aux indices ", iMax, "; ", jMax)
Fin
```

---

### Exercice 1.7

Vous étiez prévenus, c'est bête comme chou ! Il suffit de se servir de la fonction Len, et c'est réglé :

```
Var Mot : Caractère
Var Nb : Entier
Debut
Ecrire( "Entrez un mot : ")
Lire( Mot)
Nb ← Len(Mot)
Ecrire( "Ce mot compte ", Nb, " lettres")
Fin
```

---

### Exercice 1.8

Là, on est obligé de compter par une boucle le nombre d'espaces de la phrase, et on en déduit le nombre de mots. La boucle examine les caractères de la phrase un par un, du premier au dernier, et les compare à l'espace.

```
Variable Bla en Caractère
Variables Nb, i en Entier
```

```

Debut
Ecrire( "Entrez une phrase : ")
Lire( Bla)
Nb ← 0
Pour i ← 1 à Len(Bla)
  Si Mid(Bla, i , 1) = " " Alors
    Nb ← Nb + 1
  FinSi
finpour
Ecrire( "Cette phrase compte ", Nb + 1, " mots")
Fin

```

---

### Exercice 1.9

Solution 1 : pour chaque caractère du mot, on pose une très douloureuse condition composée. Le moins que l'on puisse dire, c'est que ce choix ne se distingue pas par son élégance. Cela dit, il marche, donc après tout, pourquoi pas.

```

Variable Bla en Caractère
Variables Nb, i, j en Entier
Debut
Ecrire( "Entrez une phrase : ")
Lire( Bla)
Nb ← 0
Pour i ← 1 à Len(Bla)
  Si Mid(Bla, i, 1) = "a" ou Mid(Bla, i, 1) = "e" ou Mid(Bla, i,
1) = "i" ou Mid(Bla, i, 1) = "o" ou Mid(Bla, i, 1) = "u" ou
Mid(Bla, i, 1) = "y" Alors
    Nb ← Nb + 1
  FinSi
finpour
Ecrire( "Cette phrase compte ", Nb, " voyelles")
Fin

```

Solution 2 : on stocke toutes les voyelles dans une chaîne. Grâce à la fonction Trouve, on détecte immédiatement si le caractère examiné est une voyelle ou non. C'est nettement plus sympathique...

```

Variables Bla, Voy en Caractère
Variables Nb, i, j en Entier
Debut
Ecrire( "Entrez une phrase : ")
Lire( Bla)
Nb ← 0

```

```

Voy ← "aeiouy"
Pour i ← 1 à Len(Bla)
  Si Trouve(Voy, Mid(Bla, i, 1)) <> 0 Alors
    Nb ← Nb + 1
  FinSi
finpour
Ecrire( "Cette phrase compte ", Nb, " voyelles")
Fin

```

---

## Exercice 1.10

Il n'existe aucun moyen de supprimer directement un caractère d'une chaîne... autrement qu'en procédant par collage. Il faut donc concaténer ce qui se trouve à gauche du caractère à supprimer, avec ce qui se trouve à sa droite. Attention aux paramètres des fonctions Mid, ils n'ont rien d'évident !

```

Variable Bla en Caractère
Variables Nb, i, j en Entier
Début
Ecrire( "Entrez une phrase : ")
Lire( Bla)
Ecrire( "Entrez le rang du caractère à supprimer : ")
Lire( Nb)
L ← Len(Bla)
Bla ← Mid(Bla, 1, Nb - 1) & Mid(Bla, Nb + 1, L - Nb)
Ecrire( "La nouvelle phrase est : ", Bla)
Fin

```

---

## Exercice 1.11

```

Algorithme Habitant ;
Type TDate = Enregistrement
  Jour, mois, année : entier ;
  Fin;
TAdresse = Enregistrement
  Numéro : entier ;
  Rue : chaîne [50] ;
  Ville : chaîne [20] ;
  Wilaya : chaîne [20] ;
  Cw : entier ; { Code Wilaya }
  Fin;
THabitant = Enregistrement
  Nom, prenom : chaîne [20] ;
  Date_naiss : date ;

```

```

ReSidence : Adresse ;
Fin;
Var T :Tableau[1..100] de THabitant ;
TW :Tableau[1..48] de entier ;
I,N,A :entier ;
EH :THabitant ;
Debut
Ecrire('Donner N') ;
Repete Lire(N) ; Jusqu'à N>0 et N≤100 ;
/*lecture du tableau des Habitants
Pour I←1 à N
Faire
Avec EH,EH.Date_naiss,EH.Residence
Faire
Lire(Nom,Prenom) ;
Lire(Jour,Mois,Annee) ;
Lire(Numero,Rue,Ville,Wilaya,CW) ;
Fait ;
T[I]←EH;
Fait;
/*affichage des adresses des habitants nés avant une année A
Ecrire('Donner une Année') ; Lire(A) ;
Pour I←1 à N
Faire
Avec T[I].Date_naiss, T[I].Residence
Faire
Si Annee=A Alors Ecrire(Numero,' ',Rue,' ',Ville,' ',Wilaya) Fsi;
Fait ;
Fait;
/*affichage des noms et adresses des habitants de elhamma
Pour I←1 à N
Faire
Avec T[I],T[I].Date_naiss,T[I].Residence
Faire
Si Ville='elhamma' et Wilaya='Khenchela'
Alors Ecrire(Nom,Prenom,' ',Jour,' ',Mois,' ',Annee) Fsi;
Fait ;
Fait;
/*nombre d'habitants par wilaya ;
/*initialiser à 0 ;
Pour I←1 à 48 Faire TW[I]←0 ; Fait ;
Pour I←1 à N
Faire TW[T[I].ReSidence.CW]← TW[T[I].Residence.CW]+1 ;Fait ;
/*on peut aussi utiliser une variable intermédiaire
/* A← T[I].Residence.CW; TW[A]← TW[A]+1;
/*affichage
Pour I←1 à 48 Faire Ecrire('Wilaya ',I,' Nombre ',TW[I]) ; Fait ;

```

Fin

---