

## SERIE TD 1 ALGO : PARTIE 2

Exercice 14 : Faire un programme qui saisit une date (jour, mois et année) et qui indique si l'année est bissextile

### PROGRAMME APP14

```
Type date = STRUCTURE
DEBUT
    Jour : 1 .. 31
    Mois : 1 .. 12
    Annee : entier
FIN
Var ladate : date

DEBUT
|   ecrire (« Donner le jour, le mois et l'annee de la          date à
verifier : »)
|   lire ladate.jour, ladate.mois, ladate.annee
|   Si ((ladate.annee mod 4 = 0) ET ((ladate.annee mod 100 > 0) ou
(ladate.annee mod 400 = 0)) alors
|       ecrire (« L'année : », ladate.annee, « est BISSEXTILE ! »)
|   Sinon
|       ecrire (« L'année », ladate.annee, « N'est pas BISSEXTILE »)
|   FinSi
FIN
```

Exercice 15 : Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple si l'on tape 4, l'algorithme doit calculer :  $1 + 2 + 3 + 4 = 10$  Réécrire l'algorithme qui calcule cette fois la moyenne !

### PROGRAMME APP15

```
Var x, i, som : entier
DEBUT
```

```
| Repeter
|     ecrire(« Veuillez saisir un entier > 0 :»)
|     lire x
| Jusqu'à (x > 0)
|     som <- 0
|     Pour i allant de 1 à x faire
|         som += i // som <- som + i
|     FinPour
|     afficher("La somme des nombres de 1 à », x, « est : », som)
FIN
```

[PROGRAMME\\_APP15\\_V2](#)

```
Var x, i, som, cpt : entier
DEBUT
| Repeter
|     ecrire(« Veuillez saisir un entier > 0 :»)
|     lire x
| Jusqu'à (x > 0)
|     som <- 0
|     Pour i allant de 1 à x faire
|         som += i // som <- som + i
|         cpt++
|     FinPour
|     afficher("La Moyenne des nombres de 1 à », x, « est : », som /
cpt)
FIN
```

Exercice 16 : Faire un programme qui calcule et affiche la division de a par b par soustractions successives

[PROGRAMME\\_APP16](#)

```
VAR quotient : entier
DEBUT
|   quotient<-0
|   Tantque a >= b Faire
|       a <- a - b
|       quotient <- quotient + 1
|   FinTantQue
|   afficher(« Le quotient est : », quotient)
FIN
```

Exercice 17: Faire un programme qui calcule le PGCD de deux nombres saisis au clavier en utilisant l'astuce suivante: soustrait le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

#### PROGRAMME\_APP17

```
VAR a, b : entier
DEBUT
|   ecrire(« Veuillez donner les deux nombres entiers : »)
|   lire a, b
|   Repeter
|       Si (a > b) faire
|           a <- a - b
|       Sinon
|           Si (a < b) alors
|               b <- b - a
|           Fin Si
|       FinSi
|   Jusqu'à (a = b)
|   afficher(« Le PGCD de ces deux nombres est : », a)
FIN
```

#### PROGRAMME\_APP17V2

```

VAR a, b : entier
DEBUT
|   ecrire(« Veuillez donner les deux nombres entiers : »)
|   lire a, b
|   Repeter
|       SELONQUE
|           a > b      :   a <- a - b
|           a < b      :   b <- b - a
|       FINSELONQUE
|   Jusqu'à (a = b)
|   afficher(« Le PGCD de ces deux nombres est : », a)
FIN

```

Exercice 18: Faire un programme qui calcule et affiche le PPCM de deux entiers saisis au clavier.

#### PROGRAMME APP18

```

var a , b ,c ,d : entier
DEBUT
|   ecrire (« Entrez les Valeurs de a et b :»)
|   lire a, b
|   c=a
|   d=b
|
|   Tant que a <> b
|       si (a > b) alors
|           b <- b + d
|       sinon si (a < b)
|           a <- a + c
|
|       ecrire(a, « | », b)
|       Fin tant que
|
|   afficher (« Le PPCM de », c, « et », d, « est : », a)
FIN

```

Exercice 19: Ecrire l'algorithme qui affiche la somme des prix d'une suite d'articles en CFA (entiers) saisis par l'utilisateur et se terminant par zéro.

PROGRAMME\_APP19

```
var prixArt, som : entier
```

DEBUT

```
| Repeter
|   ecrire (« Veuillez entrer le prix de l'article : »)
|   Lire prixArt
|   Si (prixArt != 0) alors
|       som <- som + prixArt
|   FinSi
|   Jusqu'à (prixArt = 0)
|   afficher (« La sommes du prix des articles saisi est : », som,
« FCFA »)
FIN
```

Exercice 20 : Ecrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui affiche à la fin le plus grand de ces 10 nombres Et affiche aussi son rang dans la liste saisie

PROGRAMME\_APP20

```
CONST NB_SAISI = 10
var max, pos, nbr, i : entier
DEBUT
|   max <- 0
|   i <- 1
|   Tant que i <= NB_SAISI faire
|       ecrire (« Veuillez donner saisir un nombre : »)
|       lire nbr
|       Si (nbr > max) alors
|           max <- nbr
|           pos <- i
|       FinSi
```

```

|   FINTANTQUE
|   afficher(« Le plus grand nombre est : », max)
|   afficher(« C'est le : », pos, « saisi »)
FIN

```

Exercice 21: Ecrire un algorithme mettant en œuvre le jeu suivant entre deux joueurs : Le premier utilisateur saisi un entier que le second doit deviner. Pour cela, il a le droit à autant de tentatives qu'il souhaite. A chaque échec, le programme lui indique si l'entier est plus grand ou plus petit que sa proposition. Un score est affiché lorsque l'entier est trouvé.

#### PROGRAMME APP20

```

VAR a, nbr, cptT : ENTIER
DEBUT
|   ECRIRE(" Entrez le nombre à deviner")
|   LIRE (a)
|   cptT ← 1
|   REPETER
|       ECRIRE("Entrez un nombre (tentative N° : ",cptT,"")
|       LIRE (nbr)
|       SELONQUE
|           nbr > a : ECRIRE (" Le Nombre cherché est plus petit que
: ",nbr )
|           nbr < a : ECRIRE(« Le Nombre cherché est grand que ",nbr)
|           nbr = a : ECRIRE (" c'est gagné : le nombre de tentatives
est" ,cptT)
|       FINSELONQUE
|       cptT ← cptT + 1
|   JUSQU'A (a = nbr)
FIN

```

Exercice 22 : On propose de saisir N entiers différents entre 1 et 100 (N étant un entier naturel compris entre 10 et 50) puis afficher la plus longue séquence croissante tout en précisant la position du premier nombre de cette séquence.

*Exemple* : Pour N=15

1 2 3 1 2 3 4 5 6 7 8 2 3 4 5

Le programme affiche :

La plus longue séquence est 1 \* 2 \* 3 \* 4 \* 5 \* 6 \* 7 \* 8 \* qui débute à la position 4 et elle est de longueur 7

#### PROGRAMME APP22

```

CONST END_TAB = 50
Type TAB = tableau[1..END_TAB] ENTIER
Var t : TAB
VAR nbr, i, nbVal, pos, j, start, end, keeper : ENTIER
DEBUT
| Répéter
|     ecrire("Combien de valeur voulez-vous saisir ?: »)
|     lire nbVal
|     Jusqu'à (nbVal > 10 ET nbVal < 50)
|     Pour i allant de 1 à nbVal faire
|         Répéter
|             ecrire("Veuillez entrer un nombre entier entre
[1..100] : »)
|             lire t[i]
|             Jusqu'à (t[i] > 1 ET t[i] < 100)
|     FinPour
| Si

```

Exercice 23. La multiplication des lapins. Vous allez faire l'acquisition d'un couple de bébés lapins. Au bout d'un mois ce couple est adulte. Le mois suivant il donne naissance à un couple de bébés lapins : vous avez maintenant 4 lapins. Puis chaque couple engendre tous les mois un nouveau couple deux mois après sa naissance.

Nous avons le schéma ci-contre :

Légende : m : bébé lapin ; M : lapin adulte.

- Mois 0. m m
- Mois 1. M M
- Mois 2. M M m m
- Mois 3. M M M M m m
- Mois 4. M M M M M m m m m

Notons  $F_N$  le nombre de lapins que l'on a au bout du  $N$ -ième mois. On convient que :  $F_0 = 2$ . Nous avons donc  $F_1 = 2$  puis  $F_2 = 4$  et  $F_3 = 6$ . Plaçons-nous au mois  $N + 2$ , nous aurons tous les couples de lapins du mois précédent (le mois  $N + 1$ ) et toutes les progénitures des couples de lapins du mois  $N$ . Nous avons donc la relation :  $F_{N+2} = F_{N+1} + F_N$

Rédiger un programme qui calcule de manière itérative le nombre de lapins au bout d'un an (le mois  $N = 12$ ).

Au bout de combien de mois dépasse-t-on le milliard de lapins ?

Résolution :

PROGRAMME APP23

```
Var nbL, fn1, fn, i: ENTIER
```

DEBUT

```
Fn ← 2
```

```
Pour i allant de 1 à 10 faire
```

```
  Si (i = 1) alors
```

```
    Fn1 ← fn
```

```
    nbL ← fn1 + fn
```

```
  SINON
```

```
    nbL ← nbL + fn1
```

```
    Fn1 ← nbL
```

```
  FINSI
```

```
FINPOUR
```

```
Afficher(« Le nombre de lapin est : »,fn2)
```

```
I<-1 ; nbL ← 0 ; fn1 ← 2 ; fn ← 0
```

```
Repeter
```

```
  nbL ← fn1 + fn
```

```
  fn ← fn1
```

```
  fn1 ← nbL
```

```
  i++
```

```
Jusqu'à (nbL = 1000000000)
```

```
Afficher(« Le nombre de mois pour atteindre 1Mds est : »,i)
```

FIN

Exercice 24: Nombre secret : écrire un programme qui demande à l'utilisateur 1 d'entrer un nombre et à l'utilisateur 2 de le trouver en affichant, à chaque tentative, « trop grand » si le nombre entré est plus grand que le nombre secret, « trop petit » sinon. Le programme s'arrête quand l'utilisateur 2 a trouvé le nombre secret.

PROGRAMME APP24



```

VAR a, n, t : ENTIER
DEBUT
    ECRIRE(" Entrez le nombre à deviner")
    LIRE (a)
    t ← 1
    REPETER
        ECRIRE("Entrez un nombre (tentative N° : ",t,")")
        LIRE (n)
        SELONQUE
            n > a: ECRIRE (" Le nombre cherché est plus petit que : ",n
)
            n < a : ECRIRE (" Le nombre cherché est plus grand que
",n)
            n = a : ECRIRE (" VOUS AVEZ GAGNE : le nombre de tentatives
est" ,t)
        FINSELONQUE
        t ← t+1
    JUSQU'A (a=n)
FIN

```

Exercice 25 : Ecrire le code C du programme qui affiche le texte suivant pour les chiffres de 1 à 10 :

```

1
2 2
3 3 3

```

Ecrire un programme qui saisit un nombre et qui dit si ce nombre est premier ou pas.

#### PROGRAMME APP25\_a

```

#include <stdio.h>
#include <windows.h>

int main(){
    int i,j;
    puts("=== AFFICHAGE DES CHIFFRES ===");
    for(i=1;i<=10;i++){
        for(j=1;j<=i;j++){

```

```
        printf("%d ",i);
    }
    printf("\n");
}
system("PAUSE");
return 0;
}
```

#### PROGRAMME APP25\_B

```
#include<stdio.h>
#include <windows.h>

main()
{
    int i,n,cpt;
    cpt=0;
    printf("➔ veuillez saisir un entier\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            cpt++;
        }
    }
    if(cpt==2)
    {
        printf("ce nombre est un nombre premier\n");
    }
}
```

```
else
{
    printf("ce nombre n'est pas un nombre premier\n");
}

}
```

Exercice 26 : Faire un programme qui saisit une suite de nombre et qui dit les si les nombres saisis sont dans l'ordre croissant ou décroissant ou quelconque.

#### PROGRAMME\_APP26

```
CONST END_TAB = 15
Type TAB = tableau[1..END_TAB] ENTIER
Var t : TAB
Var i : entiere
Var croissant, decroissant : booléen

DEBUT

    croissant ← true, decroissant ← true;
    // REMPLISSAGE DU TABLEAU
    Pour i allant de 1 à END_TAB faire
        Ecrire (« Veuillez saisir un nombre: »)
        Lire t[i]
    FINPOUR
    // ANALYSE DU VECTEUR
    Pour i allant de 1 à END_TAB faire
        SI (t[i] > t[i+1]) alors
            croissant ← false
        FINSI
        SI (t[i] < t[i+1]) alors
            decroissant ← false
```

```
    FINSI
  FINPOUR
  // AFFICHAGE DU RESULTAT
  SI (croissant ET decroissant) alors
    Ecrire("La suite est constante")
  FINSI
  SI (croissant ET !decroissant) alors
    Ecrire ("Le tableau est croissant")
  FINSI
  SI (!croissant && decroissant) alors
    Ecrire ("Le tableau est décroissant")
  FINSI
  SI (!croissant && !decroissant) alors
    Ecrire ("Le tableau est Quelconque")
  FINSI
```

**FIN**