

SERIE TD 1 ALGO

Exercice 13 : Faire un programme qui saisit une date (jour, mois et année) et qui indique si la date est valide

PROGRAMME APP13

Type date = STRUCTURE

DEBUT

Jour : 1 .. 31

Mois : 1 .. 12

Annee : entier

FIN

Var ladata : date

DEBUT

| ecrire (« Donner le jour, le mois et l'année de la date à
verifier : »)

| lire ladata.jour, ladata.mois, ladata.annee

| Si ((ladata.jour > 1 ET ladata.jour < 31) ET (ladata.mois > 1 ET
ladata.mois < 12)) alors

| Si (ladata.mois = 2 ET ladata.jour > 29) alors

| afficher (« DATE INVALIDE ! »)

| Sinon

| afficher (« LA DATE SAISIE EST VALIDE ! »)

| FinSi

| Sinon

| afficher (« LA DATE SAISIE N'EST PAS VALIDE ! »)

| Fin

FIN

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 ladata : date

DEBUT

| ecrire (« Donner le jour, le mois et l'annee de la date à
verifier : »)

| lire ladata.jour, ladata.mois, ladata.annee

| Si ((ladata.annee mod 4 = 0) ET ((ladata.annee mod 100 > 0) ou
(ladata.annee mod 400 = 0)) alors

| ecrire (« L'année : », ladata.annee, « est BISSEXTILE ! »)

| Sinon

| ecrire (« L'année », ladata.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 se 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 = 100
```

```
Type TAB = tableau[1..END_TAB] ENTIER
```

```
Var t : TAB
```

```
VAR nbr, i, nbVal, pos : 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 (nbr > 1 ET nbr < 100)  
|     FinPour
```