# CORRECTION EXERCICES ALGORITHME 1

Mr KHATORY (GIM 1° A)

1

```
EXERCICES ALGORITHME
```

```
Ecrire un algorithme permettant de résoudre une équation du second degré. Afficher les solutions !
```

```
ax^2 + bx + c = 0; solution: x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
```

#### Solution:

FINSI

FIN

```
ALGORITHME seconddegré
VAR a, b, c, delta: REEL
DEBUT
             ECRIRE (" saisissez les valeurs a, b et c de l'équation ax²+bx+c=0 : ")
             LIRE (a, b, c)
             SI (a=0)
             ALORS
                           ECRIRE (" équation du premier degré ")
                           ALORS ECRIRE ("solution est ", -c/b)
                           SINON ECRIRE (" Pas de solution")
                           FINSI
             SINON
                                                                                            Fonction
                           delta \Leftarrow b*b-4*a*c
                                                                                            standard
                           Si (delta > 0)
                           ECRIRE "les solutions sont ", -b-racine(delta) * et *, -b+racine(delta)
                                         SI delta =0 ALORS ECRIRE ( "Solution est", -b/(2a))
                                         SINON ECRIRE ("pas de solutions réelles !!")
                                         FINSI
                           FINSI
```

#### Ecrire le même algorithme avec des selon-que :

```
ALGORITHME seconddegré
VAR a, b, c, delta: REEL DEBUT
                 ECRIRE ("saisissez les valeurs de a, b et c de l'équation ax²+bx+c ")
                LIRE (a, b, c)
                Si (a=0)
                 ALORS
                                 ECRIRE ("équation du premier degré ")
                                 SI (b⇔0)
                                 ALORS ECRIRE ("solution est ", -c/b)
                                 SINON ECRIRE (" Pas de solution")
                                 FINSI
                SINON
                                 delta ⇔ b*b-4*a*c
                                 SELONQUE
                                 delta = 0 : ECRIRE ("la solution unique est:", -b/(2a)
                                 delta > 0: ECRIRE \left( \begin{tabular}{ll} les \ deux \ solutions \ \ sont \ \ ", & \  \  \, \frac{-b - racine(delta)}{2 \bullet a} \  \  \, ," \ et \ ", & \  \  \, \frac{-b + racine(delta)}{2 \bullet a} \  \  \, \right)
                                 SINON ECRIRE (" pas de solution réelle ")
                                 FINSELON
                FINSI
FIN
```

3

#### **EXERCICES ALGORITHME**

Ecrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée.

On considère que le départ et l'arrivé ont lieu le même jour!

```
Données: h1,m1,h2 et m2
```

On suppose que h2 > h1 !!

Cas possibles pour m1 et m2 2 cas ( m1<m2 ou m1 >m2)

Ecrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée.

On considère que le départ et l'arrivé ont lieu le même jour

#### Solution:

```
ALGORITHME DuréeVol
VAR h1, h2, m1, m2: ENTIER
    hd, md : ENTIER
DEBUT
            ECRIRE (" entrer horaire de départ: h min")
            LIRE (h1, m1)
            ECRIRE (" entrer horaire d'arrivée: h min")
            LIRE (h2, m2)
             SI (m2 > m1)
            ALORS
                         hd ⇔ h2-h1
                         md ⇔ m2-m1
                         ECRIRE (" la durée de vol est : ", hd , ' : ', md)
            SINON
                         hd ⇔ h2-h1-1
                         md ← m2+60-m1
                         ECRIRE (" la durée de vol est : ", hd , ' : ', md)
            FINSI
FIN
```

5

#### **EXERCICES ALGORITHME**

Ecrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée.

On considère que le départ et l'arrivé ont lieu le même jour

#### Solution n° 2:

```
ALGORITHME DureeVol1
```

```
VAR h1, h2, m1, m2: ENTIER
hd, md: ENTIER

DEBUT:

ECRIRE (" entrer horaire de départ: h min")

LIRE (h1, m1)

ECRIRE (" entrer horaire d'arrivée: h min")

LIRE (h2, m2)

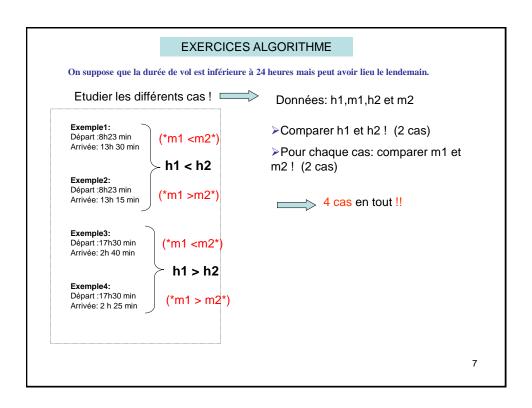
md ⇔ [h2*60+m2] - [h1*60+m1]

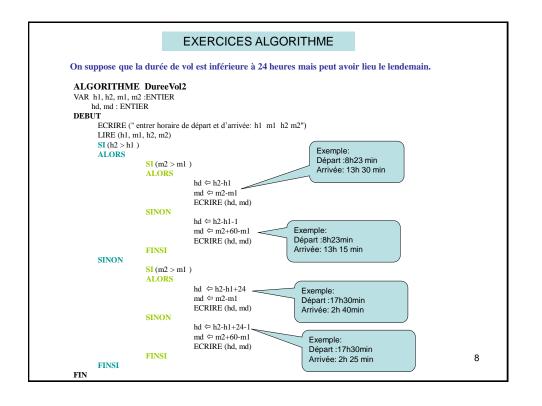
hd ⇔ md div 60 (* division entière (/)*)

md ⇔ md mod 60 (*reste de la division entière (%)*)

ECRIRE (" la durée de vol est: ", hd , ': ', md)

FIN
```

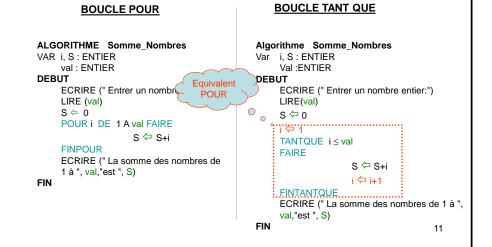




```
EXERCICES ALGORITHME
Ecrire un algorithme qui lit trois valeurs entières (A, B et C) et qui permet de les
trier par échanges successifs Et enfin les afficher dans l'ordre
ALGORITHME TriSuccessif
VAR A, B, C : ENTIER
DEBUT
ECRIRE (" entrer Les valeurs A, B et C ")
LIRE(A,B,C)
                                            ici A < B reste à vérifier B ? C
SI (A > B) ALORS
          echange (A,B)
          SI B > C ALORS
                                                   ici B < C ET A < C (reste A?B)
                    echange (B,C) -
                    SI A > B ALORS
                               echange (A,B)
                    FINSI
                                                    Finalement A < B < C
          FINSI
SINON
                                                   Ici B < C ET A < C ((reste A ? B)
          SIB > CALORS
                    echange (B,C)
                     SI A >B ALORS
                               echange (A,B)
                                                           Ici A <B ET B<C \Rightarrow A<B<C
                    FINSI
          FINSI
FINSI
                                    Finalement A < B < C
ECRIRE ("Les valeurs A, B et C sont (d
                                                                                            9
FIN
```

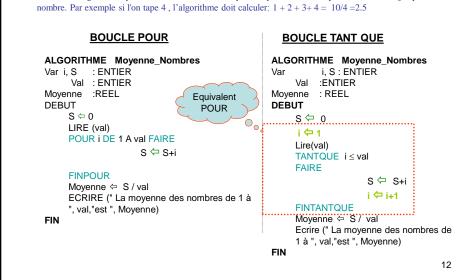
```
EXERCICES ALGORITHME
Ecrire un algorithme calculatrice permettant la saisie du premier entier (a) de l'opération (+
ou - ou * ou / : sont des caractères) et du deuxième entier (b) et qui affiche le résultat
ALGORITHME calculatrice
VAR a, b: ENTIER
   op: CARACTERE
DEBUT
     ECRIRE (" saisissez le premier entier ")
     LIRE (a)
      ECRIRE (" saisissez l'opération ")
     LIRE (op)
     ECRIRE (" saisissez le deuxième entier")
     LIRE (b)
      SELONQUE:
           Op = '+':
                       ECRIRE ("la somme de ",a, "et de ",b, "est égale",a+b)
           Op = '*':
                       ECRIRE ("le produit de ",a, "et de ",b, "est égale",a*b)
           Op = '/':
                       SI (b= 0) ALORS ECRIRE (" division impossible ")
                       SINON ECRIRE ("la division de ",a, "par ",b, "est égale", a/b)
           Op = '-': ECRIRE ("la soustraction de ",a, "et de ",b, "est égale", a-b)
      SINON: ECRIRE((" Opération invalide ")
      FINSELONQUE
FIN
                                                                                             10
```

1. 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



#### **EXERCICES ALGORITHME**

1. Ecrire un algorithme qui demande un nombre de départ, et qui calcule la **moyenne** des entiers jusqu'à ce nombre. Par exemple si l'on tape 4. l'algorithme doit calculer: 1 + 2 + 3 + 4 = 10/4 = 2.5



Ecrire l'algorithme qui affiche la somme des prix d'une suite d'articles en DH (entiers) saisies par l'utilisateur et se terminant par zéro.

#### **BOUCLE TANTQUE**

```
ALGORITHME Somme_Prix

VAR p, S: ENTIER

DEBUT

S ⇔ 0

ECRIRE("Entrer le prix du 1° article:")

LIRE (p)

TANTQUE (p≠0)

FAIRE

S ⇔ S+p

ECRIRE("Entrer le prix de l'article suivant( 0 si Fin):")

LIRE(p)

FINTANTQUE

ECRIRE (" La somme des prix des articles est ", S)
```

#### **BOUCLE REPETER**

```
ALGORITHME Somme_Prix

VAR p, S : ENTIER

DEBUT

S ⇔ 0

ECRIRE("Entrer le prix du 1° article:")

LIRE(p)

REPETER

S ⇔ S+p

ECRIRE("Entrer le prix de l'article suivant( 0 si Fin):")

LIRE(p)

JUSQU'A (p = 0)

ECRIRE (" La somme des prix des articles est ", S)

FIN
```

13

#### **EXERCICES ALGORITHME**

Ecrire l'algorithme qui affiche la somme des prix d'une suite d'articles en DH (entiers) saisies par l'utilisateur et **se terminant par zéro**.

#### Cas d'entrée à la boucle:

Si au départ p= 0

→ choisir la boucle TANTQUE

```
On peut changer l'algorithme :

ALGORITHME Somme_Prix
VAR p, S : ENTIER
DEBUT
S \( \to 0

REPETER

ECRIRE("Entrer le prix de l'article ( 0 si Fin):")
LIRE(p)
S \( \to S + p

JUSQU'A p =0
ECRIRE (" La somme des prix des articles est ", S)
```

Cas particulier (p=0) (Boucle REPETER):

```
ALGORITHME Somme_Prix

VAR p, S: ENTIER

DEBUT
S ⇔ 0

ECRIRE("Entrer le prix du 1° article:")

LIRE(p)

REPETER
S ⇔ S+p
ECRIRE("Entrer le prix de l'article suivant( 0 si Fin):")

LIRE(p)

JUSQU'A p = 0

ECRIRE (" La somme des prix des articles est ", S)

FIN

14
```

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 aussi son rang

#### Exemple:

Entrez le nombre numéro 1 : 13
Entrez le nombre numéro 2 : 17
....
Entrez le nombre numéro 10 : 5
Le plus grand de ces nombres est : 17
C'était le 2 ème nombre saisi

```
EXERCICES ALGORITHME
ALGORITHME Somme_10Nombres
                                                 POUR indice DE 2 A NBRE
CONST
          NBRE=10;
                                                      Ecrire("Entrez le nombre numéro", indice)
          indice, val: ENTIER
VAR
                                                      Lire (Val)
          Indice_grand, PLUSGRAND :ENTIER
                                                      Si val > PLUSGRAND alors
DEBUT
                                                           Indice_grand ← indice
     ECRIRE("Entrez\ le\ 1^{er}\ nombre\ :\ ")
                                                           PLUSGRAND ← val
     LIRE (Val)
                                                      Finsi
     PLUSGRAND ← val
                                                FINPOUR
     Indice_grand ← 1
     TANTQUE (indice \leq NBRE)
     FAIRE
          ECRIRE("Entrez le nombre numéro : ", indice)
          LIRE (Val)
          SI (val > PLUSGRAND) ALORS
                     indice_grand ← indice
                     PLUSGRAND ← val
          FINSI
          indice ← indice+1
     FINTANTQUE
     ECRIRE("le plus grand de ces nombres est:", PLUSGRAND)
     ECRIRE(" c'était le ",indice_grand, " ème nombre saisi)
                                                                                          16
FIN
```

Ecrire un programme mettant en œuvre le jeu suivant :

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 cherché est plus grand ou plus petit que sa proposition.

Un score indiquant le nombre de coups joués est mis à jour et affiché lorsque l'entier est trouvé.

```
ALGORITHME devinette
VAR a, n, t: ENTIER
DEBUT
ECRIRE(" Entrez le nombre à deviner")
LIRE (a)
ECRIRE("entrez le nombre ( premier essai")
t⇔1
TANTOUE (a≠n )
FAIRE
           SI n>a ALORS ECRIRE (" nombre cherché plus petit que : ",n )
           SINON ECRIRE (" nombre cherché plus grand que ",n)
           FINSI
           t ← t+1
           ECRIRE("entrez un autre nombre (tentative N° ",t,")")
           LIRE (n)
FINTANTQUE
ECRIRE (" c'est gagné : le nombre de tentatives est" ,t)
```

17

#### **EXERCICES ALGORITHME**

Ecrire un programme mettant en œuvre le jeu suivant :

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 cherché est plus grand ou plus petit que sa proposition.

Un score indiquant le nombre de coups joués est mis à jour et affiché lorsque l'entier est trouvé.

```
ALGORITHME devinette 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 (" nombre cherché plus petit que: ",n)

n <a: ECRIRE (" nombre cherché plus grand que ",n)

n=a: ECRIRE (" c'est gagné: le nombre de tentatives est",t)

FINSELONQUE

t ⇔ t+1

JUSQU'A (a=n)

FIN
```

18

**BOUCLE REPETER** 

Écrire une fonction F\_PGCD qui retourne le PGCD de deux nombres en utilisant l'astuce suivante: soustrait le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

#### **Exemple:**

```
a=24
b=36
Le PGCD ??
Boucle:
1. \mathbf{a} < \mathbf{b} (24 < 36) \rightarrow \mathbf{b} = 36-24=12
2. \mathbf{b} < \mathbf{a} (12 < 24) \rightarrow \mathbf{a} = 24-12=12
```

a=b=12 on s'arrête donc le PGCD est 12

19

#### **EXERCICES ALGORITHME**

Écrire une fonction F\_PGCD qui retourne le PGCD de deux nombres en utilisant l'astuce suivante: soustrait le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

```
FONCTION F_PGCD(a,b :ENTIER) :ENTIER

DEBUT

REPETER

SELONQUE

a>b: a \( \phi \)a-b

a < b: b \( \phi \)b-a

FINSELON

JUSQU"A a=b

retourner(a)

FIN
```

Ecrire la même fonction en utilisant l'algorithme d'Euclide: Utiliser les structures TANTQUE puis REPETER JUSQU'A

# Boucle TANT QUE:

# FONCTION F\_PGCD(a,b :ENTIER) :ENTIER VAR r :ENTIER DEBUT $r \hookrightarrow a\%b /^*$ reste de la division entière\*/ TANTQUE $r\neq 0$ FAIRE $a \hookrightarrow b$ $b \hookrightarrow r$

 $r \hookleftarrow a\%b$ FINTANTQUE
retourner(b)

FIN

# Boucle REPETER JUSQU'A:

FONCTION F\_PGCD(a,b :ENTIER) :ENTIER VAR r :ENTIER DEBUT REPETER  $\begin{matrix} r \Leftrightarrow a\%b \\ a \Leftrightarrow b \\ b \Leftrightarrow r \end{matrix}$  JUSQU'A r=0 retourner(b) FIN

21

Merci & FIN