Enoncé et Corrigé type des exercices

Module:Informatique 01

Enseignant :Bezza Youcef .

Semestre : 01

Année Universitaire:2018/2019

TD 1

ENONCE DES EXERCICES

Exercice 1.1

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Var  
  A, B : Entier  
Début  
A ← 1  
B ← A + 3  
A ← 3  
Fin
```

Exercice 1.2

Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

```
Var  
  A, B, C : Entier  
Début  
A ← 5  
B ← 3  
C ← A + B  
A ← 2  
C ← B - A  
Fin
```

Exercice 1.3

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Var  
A, B : Entier  
Début  
A ← 5  
B ← A + 4  
A ← A + 1  
B ← A - 4  
Fin
```

Exercice 1.4

Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

```
Var  
  A, B, C : Entier  
Début  
  A ← 3  
  B ← 10  
  C ← A + B
```

```
B ← A + B
```

```
A ← C
```

```
Fin
```

Exercice 1.5

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

```
Var
```

```
  A, B : Entier
```

```
Début
```

```
A ← 5
```

```
B ← 2
```

```
A ← B
```

```
B ← A
```

```
Fin
```

Les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

Exercice 1.6

Plus difficile, mais c'est un classique absolu, qu'il faut absolument maîtriser : écrire un algorithme permettant d'échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.

Exercice 1.7

Une variante du précédent : on dispose de trois variables A, B et C. Ecrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).

Exercice 1.8

Que produit l'algorithme suivant ?

```
Var
```

```
  A, B, C : CHAINE
```

```
Début
```

```
A ← "423"
```

```
B ← "12"
```

```
C ← A + B
```

```
Fin
```

Exercice 1.9

Que produit l'algorithme suivant ?

```
Var
```

```
  A, B, C : CHAINE
```

```
Début
```

```
A ← "423"
```

```
B ← "12"
```

```
C ← A & B
```

```
Fin
```

TD 1

CORRIGES DES EXERCICES

Exercice 1.1

Après	La valeur des variables est :	
A ← 1	A = 1	B = ?
B ← A + 3	A = 1	B = 4
A ← 3	A = 3	B = 4

Exercice 1.2

Après	La valeur des variables est :		
A ← 5	A = 5	B = ?	C = ?
B ← 3	A = 5	B = 3	C = ?
C ← A + B	A = 5	B = 3	C = 8
A ← 2	A = 2	B = 3	C = 8
C ← B - A	A = 2	B = 3	C = 1

Exercice 1.3

Après	La valeur des variables est :	
A ← 5	A = 5	B = ?
B ← A + 4	A = 5	B = 9
A ← A + 1	A = 6	B = 9
B ← A - 4	A = 6	B = 2

Exercice 1.4

Après	La valeur des variables est :		
A ← 3	A = 3	B = ?	C = ?
B ← 10	A = 3	B = 10	C = ?
C ← A + B	A = 3	B = 10	C = 13
B ← A + B	A = 3	B = 13	C = 13
A ← C	A = 13	B = 13	C = 13

Exercice 1.5

Après	La valeur des variables est :	
A ← 5	A = 5	B = ?
B ← 2	A = 5	B = 2
A ← B	A = 2	B = 2
B ← A	A = 2	B = 2

Les deux dernières instructions ne permettent donc pas d'échanger les deux valeurs de B et A, puisque l'une des deux valeurs (celle de A) est ici écrasée. Si l'on inverse les deux dernières instructions, cela ne changera rien du tout, hormis le fait que cette fois c'est la valeur de B qui sera écrasée.

Exercice 1.6

Début

...

C ← A

A ← B

B ← C

Fin

On est obligé de passer par une variable dite temporaire (la variable C).

Exercice 1.7

Début

...

D ← C

C ← B

B ← A

A ← D

Fin

En fait, quel que soit le nombre de variables, une seule variable temporaire suffit...

Exercice 1.8

Il ne peut produire qu'une erreur d'exécution, puisqu'on ne peut pas additionner des caractères.

Exercice 1.9

...En revanche, on peut les concaténer. A la fin de l'algorithme, C vaudra donc "42312".

TD 2

ENONCE DES EXERCICES

Exercice 2.1

Quel résultat produit le programme suivant ?

```
Var
  val, double Entier
Début
Val ← 231
Double ← Val * 2
Ecrire( Val)
Ecrire( Double)
Fin
```

Exercice 2.2

Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre.

Exercice 2.3

Ecrire un programme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

TD 2

CORRIGES DES EXERCICES

Exercice 2.1

On verra apparaître à l'écran 231, puis 462 (qui vaut $231 * 2$)

Exercice 2.2

```
Var
  nb, carr : Entier
Début
Ecrire( "Entrez un nombre :")
Lire( nb)
carr ← nb * nb
Ecrire( "Son carré est : ", carr)
Fin
```

En fait, on pourrait tout aussi bien économiser la variable carr en remplaçant les deux avant-dernières lignes par :

```
Ecrire( "Son carré est : ", nb*nb)
```

C'est une question de style ; dans un cas, on privilégie la lisibilité de l'algorithme, dans l'autre, on privilégie l'économie d'une variable.

Exercice 2.3

```
Var
  nb, pht, ttva, pttc : Réel
Début
Ecrire( "Entrez le prix hors taxes :")
Lire( pht)
Ecrire( "Entrez le nombre d'articles :")
Lire( nb)
Ecrire( "Entrez le taux de TVA :")
Lire( ttva)
pttc ← nb * pht * (1 + ttva)
Ecrire( "Le prix toutes taxes est : ", pttc)
Fin
```

Là aussi, on pourrait squeezer une variable et une ligne en écrivant directement. :

```
Ecrire( "Le prix toutes taxes est : ", nb * pht * (1 + ttva))
```

C'est plus rapide, plus léger en mémoire, mais un peu plus difficile à relire (et à écrire !)

TD 3

ENONCE DES EXERCICES

Exercice 3.1

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

Exercice 3.2

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas** calculer le produit des deux nombres.

Exercice 3.3

Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.

Exercice 3.4

Ecrire un algorithme qui demande un nombre à l'utilisateur, et l'informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

Exercice 3.5

Ecrire un algorithme qui demande deux nombres à l'utilisateur et l'informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

TD 3

CORRIGES DES EXERCICES

Exercice 3.1

```
Var
n : Entier
Début
Ecrire( "Entrez un nombre : ")
Lire( n)
Si n > 0 Alors
    Ecrire( "Ce nombre est positif")
Sinon
    Ecrire( "Ce nombre est négatif")
Finsi
Fin
```

Exercice 3.2

```
Var
m, n : Entier
Début
Ecrire( "Entrez deux nombres : ")
Lire( m, n)
Si (m > 0 ET n > 0) OU (m < 0 ET n < 0) Alors
    Ecrire( "Leur produit est positif")
Sinon
    Ecrire( "Leur produit est négatif")
Finsi
Fin
```

Exercice 3.3

```
Var
a, b, c : CHAINE
Début
Ecrire( "Entrez successivement trois noms : ")
Lire( a, b, c)
Si a < b ET b < c Alors
    Ecrire( "Ces noms sont classés alphabétiquement")
Sinon
    Ecrire( "Ces noms ne sont pas classés")
Finsi
Fin
```

Exercice 3.4

```
Var
n : Entier
Début
```

```
Ecrire( "Entrez un nombre : ")  
Lire( n)  
Si n < 0 Alors  
    Ecrire( "Ce nombre est négatif")  
SinonSi n = 0 Alors  
    Ecrire( "Ce nombre est nul")  
Sinon  
    Ecrire( "Ce nombre est positif")  
Finsi  
Fin
```

Exercice 3.5

```
Var  
    m, n : Entier  
Début  
Ecrire( "Entrez deux nombres : ")  
Lire( m, n)  
Si m = 0 OU n = 0 Alors  
    Ecrire( "Le produit est nul")  
SinonSi (m < 0 ET n < 0) OU (m > 0 ET n > 0) Alors  
    Ecrire( "Le produit est positif")  
Sinon  
    Ecrire( "Le produit est négatif")  
Finsi  
Fin
```

Si on souhaite simplifier l'écriture de la condition lourde du **SinonSi**, on peut toujours passer par des variables booléennes intermédiaires. Une astuce de sioux consiste également à employer un Xor (c'est l'un des rares cas dans lesquels il est pertinent)

TD 4

ENONCE DES EXERCICES

Exercice 4.1

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

Exercice 4.2

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 6.

Exercice 4.3

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 25.

Exercice 4.4

Réécrire l'algorithme précédent, en utilisant cette fois l'instruction **Pour**

Exercice 4.5

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

```
Table de 7 :  
7 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
...  
7 x 10 = 70
```

Exercice 4.6

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 entre 5, le programme doit calculer :

```
1 + 2 + 3 + 4 + 5 = 15
```

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Exercice 4.7

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée 8!, vaut

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

Exercice 4.8

Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

C'était le nombre numéro 2

TD 4

CORRIGES DES EXERCICES

Exercice 4.1

```
Var
  N : Entier
Debut
  N ← 0
  Ecrire( "Entrez un nombre entre 1 et 3")
  TantQue N < 1 ou N > 3
    Lire( N)
    Si N < 1 ou N > 3 Alors
      Ecrire( "Saisie erronée. Recommencez")
    FinSi
  FinTantQue
Fin
```

Exercice 4.2

```
Var
  N : Entier
Debut
  N ← 0
  Ecrire( "Entrez un nombre entre 10 et 20")
  TantQue N < 10 ou N > 20
    Lire( N)
    Si N < 10 Alors
      Ecrire( "Plus grand !")
    SinonSi N > 20 Alors
      Ecrire( "Plus petit !")
    FinSi
  FinTantQue
Fin
```

Exercice 4.3

On peut imaginer deux variantes, strictement équivalentes :

```
Var
  N, i : Entier
Debut
  Ecrire( "Entrez un nombre : ")
  Lire (N)
  Stop ← N+10
  Ecrire( "Les 10 nombres suivants sont : ")
  TantQue N < Stop
    N ← N+1
    Ecrire( N)
```

```
FinTantQue
```

```
Fin
```

Ou bien :

```
Var
```

```
  N, i : Entier
```

```
Debut
```

```
Ecrire( "Entrez un nombre : ")
```

```
Lire (N)
```

```
i ← 0
```

```
Ecrire( "Les 10 nombres suivants sont : ")
```

```
TantQue i < 10
```

```
  i ← i + 1
```

```
  Ecrire( N + i)
```

```
FinTantQue
```

```
Fin
```

Exercice 4.4

Là encore, deux variantes, correspondant trait pour trait à celles du corrigé précédent :

```
Var
```

```
  N, i : Entier
```

```
Debut
```

```
Ecrire( "Entrez un nombre : ")
```

```
Lire( N)
```

```
Ecrire( "Les 10 nombres suivants sont : ")
```

```
Pour i ← N + 1 à N + 10
```

```
  Ecrire( i)
```

```
Finpour
```

```
Fin
```

Ou bien :

```
Var
```

```
  N, i : Entier
```

```
Debut
```

```
Ecrire( "Entrez un nombre : ")
```

```
Lire( N)
```

```
Ecrire( "Les 10 nombres suivants sont : ")
```

```
Pour i ← 1 à 10
```

```
  Ecrire( N + i)
```

```
Finpour
```

```
Fin
```

WORK HARD