

TD4

Exercice 1 :

Ecrire un programme en assembleur qui lit un nombre grand de secondes au clavier, il le convertit ensuite en heures, minutes et secondes restantes, vous deviez utiliser les registres suivants :

\$t1 : pour lire le nombre de secondes en entrée.

\$t2 : contient le nombre d'heures calculé.

\$t3 : contient le nombre de minutes calculé.

\$t4 : contient le nombre de secondes calculé.

\$t5 : contient la constante 60

L'exécution doit ressembler à ça :

```
donnez le nombre de secondes SVP:4063
le nombre d'heures est:1
le nombre de minutes est:7
le nombre de secondes est:43
-- program is finished running (dropped off bottom) --
```

Exercice 2 :

Ecrire un programme en assembleur qui convertit 216548 bits en kilo octets, octets et bits restants, l'exécution doit ressembler à ça :

```
le nombre de bits restants est:4
le nombre d'octets est:444
le nombre de kilos octets est:26
```

Exercice 3 :

Ecrire un programme en assembleur qui lit un entier au clavier, et il affiche à l'écran la somme des entiers de 1 jusqu'à ce nombre, l'exécution doit ressembler à ça :

```
donnez le nombre SVP:6
la somme est :21
-- program is finished running (dropped off bottom) --
```

Exercice 4 :

Ecrire un programme en assembleur qui permet de lire 10 valeurs entières entrées au clavier, de vérifier si la constante 23 existe dans cette suite ou non ; il affiche le message suivant : "la valeur 23 existe dans cette suite" si la valeur 23 existe dans les valeurs entrées ou bien le message suivant : "la valeur 23 n'existe pas dans cette suite" dans le cas contraire. L'exécution du programme ressemble à ça :

```
insérer l'élément SVP:20
insérer l'élément SVP:5
insérer l'élément SVP:13
insérer l'élément SVP:3
insérer l'élément SVP:9
insérer l'élément SVP:23
insérer l'élément SVP:18
insérer l'élément SVP:22
insérer l'élément SVP:31
insérer l'élément SVP:2
la valeur 23 existe pas dans cette suite
```

```
insérer l'élément SVP:20
insérer l'élément SVP:5
insérer l'élément SVP:13
insérer l'élément SVP:3
insérer l'élément SVP:9
insérer l'élément SVP:18
insérer l'élément SVP:22
insérer l'élément SVP:31
insérer l'élément SVP:2
insérer l'élément SVP:64
la valeur 23 n'existe pas dans cette suite
```

Indication :

Vous pouvez utiliser 4 registres :

\$t0 : pour la variable compteur.

\$t2 : pour les valeurs entrées au clavier.

\$t3 : pour stocker la constante 23.

\$t4 : valeur booléenne qui prend la valeur 0 si 23 n'existe pas dans la suite et 1 si elle existe.

seq \$t1, \$t2, \$t3 : si \$t2 est égal à \$t3 alors mettre 1 dans \$t1 sinon 0.

beqz \$t1, label : se brancher à label si \$t1 est égal à 0.

TP4

Refaites tous les exercices précédents sur machine et vérifiez leurs exécutions.

Correction TP4

Solution exercise 1 :

```
.data
entree: .asciiz "donnez le nombre de secondes SVP:"
heure: .asciiz "\nle nombre d'heures est:"
minute: .asciiz "\nle nombre de minutes est:"
seconde: .asciiz "\nle nombre de secondes est:"
```

```
.text
li $t5, 60
```

```
li $v0, 4
la $a0, entree
syscall
```

```
li $v0, 5
syscall
move $t1, $v0
```

```
div $t1, $t5
mfhi $t4
mflo $t1
```

```
div $t1, $t5
mfhi $t3
mflo $t2
```

```
li $v0, 4
la $a0, heure
syscall
```

```
move $a0, $t2
li $v0, 1
syscall
```

```
li $v0, 4
la $a0, minute
syscall
```

```
move $a0, $t3
li $v0, 1
syscall
```

```
li $v0, 4
la $a0, seconde
syscall
```

```
move $a0, $t4
```

```
li $v0, 1
syscall
```

Solution exercise 2 :

```
.data
bits: .asciiz "le nombre de bits restants est:"
octets: .asciiz "\nle nombre d'octets est:"
kiloctets: .asciiz "\nle nombre de kilos octets est:"
nombre1 : .byte 8
nombre2 : .half 1024
nombre3 : .word 216548
```

```
.text
lb $t1, nombre1
lh $t2, nombre2
lw $t3, nombre3
```

```
div $t3, $t1
```

```
li $v0, 4
la $a0, bits
syscall
```

```
mfhi $a0
```

```
li $v0, 1
syscall
```

```
mflo $t4
```

```
div $t4, $t2
```

```
li $v0, 4
la $a0, octets
syscall
```

```
mfhi $a0
```

```
li $v0, 1
syscall
```

```
li $v0, 4
la $a0, kiloctets
syscall
```

```
mflo $a0
li $v0, 1
syscall
```

Solution exercise 3 :

```
.data
nombre :.word 1
entree:.asciiz "donnez le nombre SVP:"
sortie:.asciiz "la somme est :"
.text
lw $t3, nombre
lw $t4, nombre

li $v0, 4
la $a0, entree
syscall
```

```
li $v0, 5
syscall
move $t2, $v0
```

```
boucle:
beq $t3, $t2, fin
addi $t3, $t3, 1
add $t4, $t4, $t3
b boucle
```

```
fin :
li $v0, 4
la $a0, sortie
syscall
```

```
li $v0, 1
move $a0, $t4
syscall
```

Solution exercise 4 :

```
.data
entree :.asciiz "insérer l'élément SVP:"
sortie1:.asciiz "la valeur 23 n'existe pas dans cette suite"
sortie2:.asciiz "la valeur 23 existe pas dans cette suite"
.text
addi $t0, $zero, 0
addi $t3, $zero, 23
```

```
boucle:
beq $t0, 10, fin
li $v0, 4
la $a0, entree
syscall
li $v0, 5
syscall
move $t2, $v0
```

```
beq $t2, 23, poserbool
addi $t0, $t0, 1
b boucle
poserbool:
seq $t4, $t2, $t3
addi $t0, $t0, 1
b boucle
```

```
fin:
beqz $t4, afficher1
li $v0, 4
la $a0, sortie2
syscall
li $v0, 10
syscall
afficher1:
li $v0, 4
la $a0, sortie1
syscall
```