

# **Complément Série TP1 AO Bourbia univ Guelma avec Solution**

© Charaf Tchi

2017-2018

# Table des matières

<b>1</b>	<b>EXO 01</b>	<b>1</b>
1.1	Solution Exo 01 . . . . .	2
<b>2</b>	<b>EXO 02</b>	<b>2</b>
2.1	Solution Exo 02 . . . . .	3
<b>3</b>	<b>EXO 03</b>	<b>4</b>
3.1	Solution Exo 03 . . . . .	4
<b>4</b>	<b>EXO 04</b>	<b>6</b>
4.1	Solution Exo 04 . . . . .	7
<b>5</b>	<b>EXO 05</b>	<b>9</b>
5.1	Solution Exo 05 . . . . .	9
<b>6</b>	<b>EXO 06</b>	<b>9</b>
6.1	Solution Exo 06 . . . . .	10

## 1 EXO 01

### Exercice n°1 :

- Ecrire un programme MIPS qui convertit une distance (en mètres) entrée au clavier et l'affiche en KM.

## 1.1 Solution Exo 01

```
exol.txt
1  #Copyright Charaf Tchi
2  .data
3  Metre: .ascii "Entrez la distance en metre: "
4  Kilometre: .ascii "La distance en KM est: "
5  mille: .double 1000.0
6  .text
7
8  #####
9  la $a0, Metre
10 li $v0, 4      #Affichage -> Entrez la distance en metre:
11 syscall
12 #####
13
14 #####
15 la $v0, 7
16 syscall      #Read double number
17 #####
18
19 ldcl $f4, mille  #$f4=1000
20
21 div.d $f12 $f0, $f4  #$f12=$f0/1000
22
23 #####
24 li $v0, 4
25 la $a0, Kilometre #Affichage -> La distance en KM est:
26 syscall
27 #####
28
29 #####
30 li $v0, 3
31 syscall      #Print double number, should be in $f12 before
32 #####
--
```

## 2 EXO 02

### Exercice n°2 :

- Écrire un programme MIPS qui permet de saisir le prix Hors Taxe d'un article, le nombre d'articles et le taux de TVA, et qui calcule et affiche le prix total TTC correspondant.

## 2.1 Solution Exo 02

```
exo2.txt*
1  #Copyright Charaf Tchi
2
3  #la regle: nombre d'articles*prix Hors Taxe(taux de TVA+1)
4
5  .data
6  str:.asciiz "Enter le prix hors taxe: "
7  str1:.asciiz "\nEntrer le nombre d'article: "
8  str2:.asciiz "\nEntrer le taux de TVA: "
9  str3:.asciiz "\nLe PTTC est: "
10 var:.double 1.0
11 .text
12
13 main:
14 #####
15 li $v0,4
16 la $a0,str    #Print-> Enter le prix hors taxe:
17 syscall
18 #####
19
20 #####
21 la $v0,7
22 syscall #Read double number
23 #####
24
25 mov.d $f2,$f0  #$f2=$f0
26
27 #####
28 li $v0,4
29 la $a0,str1
30 syscall #Print-> Entrer le nombre d'article:
31 #####
32
33 #####
34 la $v0,7
35 syscall #Read double number
36 #####
37
38 movf.d $f4,$f0  #$f4=$f0
39
40 #####
41 li $v0,4
42 la $a0,str2    #Print->Entrer le taux de TVA:
43 syscall
44 #####
45
46 #####
47 la $v0,7
48 syscall #Read double number
49 #####
50
51
52 movf.d $f6,$f0  #$f6=$f0
53
54 ldc1 $f8,var    #$f8=1.0
55
56 ##### add.d(+) mul.d(*) for double numbers we use .d
57 add.d $f12,$f6,$f8  #$f12=$f6+1.0
58 mul.d $f12,$f12,$f4  #$f12=$f12*$f4
59 mul.d $f12,$f12,$f2  #$f12=$f12*$f2
60 #####
```

```

61
62 #####
63 li $v0,4
64 la $a0,str3 #Print->Le PTTC est:
65 syscall
66 #####
67
68 #####
69 li $v0,3
70 syscall #Print double (in $f12)
71 #####

```

## 3 EXO 03

Exercice n°3 :

- Écrire un programme MIPS permettant d'afficher le mois en lettres selon son numéro saisi au clavier.

### 3.1 Solution Exo 03

```

exo4.txt
1 #Copyright Charaf Tchi
2 .data
3 str: .asciiz "Entrer l'ordre de mois svp: "
4 str1: .asciiz "Le mois est Janvier"
5 str2: .asciiz "Le mois esy Fevrier"
6 str3: .asciiz "Le mois est Mars"
7 str4: .asciiz "Le mois est Avril"
8 str5: .asciiz "Le mois est Mai"
9 str6: .asciiz "Le mois est Juin"
10 str7: .asciiz "Le mois est Juillet"
11 str8: .asciiz "Le mois est Aout"
12 str9: .asciiz "Le mois est Septembre"
13 str10: .asciiz "Le mois est Octobre"
14 str11: .asciiz "Le mois est Novembre"
15 str12: .asciiz "Le mois est Decembre"
16 .text
17
18 main :
19 ##### Print-> Entrer l'ordre de mois svp:
20 li $v0,4
21 la $a0,str
22 syscall
23
24 ##### Read the number ($v0,5 for integer)
25 la $v0,5
26 syscall
27
28 move $t1,$v0 # $t1=$v0
29

```

```

30 beq $t1,1,label1 #if $t1=1 then label1 (the same for the rest)
31 beq $t1,2,labe2
32 beq $t1,3,labe3
33 beq $t1,4,labe4
34 beq $t1,5,labe5
35 beq $t1,6,labe6
36 beq $t1,7,labe7
37 beq $t1,8,labe8
38 beq $t1,9,labe9
39 beq $t1,10,label10
40 beq $t1,11,label11
41 beq $t1,12,label12
42
43 label1: #### Print: Janvier
44 li $v0,4
45 la $a0,str1
46 syscall
47 j label13
48
49 labe2: #### Print:Fevrier
50 li $v0,4
51 la $a0,str2
52 syscall
53 j label13
54
55 labe3: #### Print:Mars
56 li $v0,4
57 la $a0,str3
58 syscall
59 j label13
60
61 labe4: #### Print:Avril
62 li $v0,4
63 la $a0,str4
64 syscall
65 j label13
66
67 labe5: #### Print:Mai
68 li $v0,4
69 la $a0,str5
70 syscall
71 j label13
72
73 labe6: #### Print:Juin
74 li $v0,4
75 la $a0,str6
76 syscall
77 j label13
78
79 labe7: #### Print:Juillet
80 li $v0,4
81 la $a0,str7
82 syscall
83 j label13
84
85 labe8: #### Print:Aout
86 li $v0,4
87 la $a0,str8
88 syscall
89 j label13
90

```

```
91  labe9: #### Print:Septembre
92  li $v0,4
93  la $a0,str9
94  syscall
95  j label3
96
97  label10: #### Print:Octobre
98  li $v0,4
99  la $a0,str10
100 syscall
101 j label3
102
103 label11: #### Print:Novembre
104 li $v0,4
105 la $a0,str11
106 syscall
107 j label3
108
109 label12: #### Print:Decembre
110 li $v0,4
111 la $a0,str12
112 syscall
113
114 label13: #### Exit the program
115 li $v0,10
116 syscall
```

## 4 EXO 04

Exercice n°4 :

- Ecrire un programme MIPS qui permet à partir de 2 valeurs numérique de réaliser l'addition (+), la soustraction (-), la multiplication (\*) ou la division selon le choix de l'utilisateur.

**Indication** : il faut que l'utilisateur introduise les deux nombres et l'opérateur.

### 4.1 Solution Exo 04

```
exo3.txt*
1  .data
2  str:.asciiz "Entrer A: "
3  str1:.asciiz "Entrer B: "
4  str2:.asciiz "\nEntrer votre operation +|-|*|/:"
5  res:.asciiz "\nLe resultat est: "
6  .text
7
8  main:
9  ##### Print: Entrer A
10 li $v0,4
11 la $a0,str
12 syscall
13
14 ##### Read A
15 la $v0,5
16 syscall
17
18 move $t1,$v0
19
20 ##### Print: Entrer B
21 li $v0,4
22 la $a0,str1
23 syscall
24
25 ##### Read B
26 la $v0,5
27 syscall
28
29 move $t2,$v0
30
```

```

31 ##### Print: Entrer votre operation
32 li $v0,4
33 la $a0,str2
34 syscall
35
36 ##### Read operator ($v0,12 to read char)
37 la $v0,12
38 syscall
39
40 move $a3,$v0
41
42 ##### if $a3='+' then label (the same for the rest)
43 beq $a3,'+',label1
44 beq $a3,'-',labe2
45 beq $a3,'*',labe3
46 beq $a3,'/',labe4
47
48 label1:#$t3=A+B
49 add $t3,$t1,$t2
50 j label0
51
52 labe2:#$t3=A-B
53 sub $t3,$t1,$t2
54 j label0
55
56 labe3:#$t3=A*B
57 mul $t3,$t1,$t2
58 j label0
59
60 labe4:#$t3=A/B
61 div $t1,$t2
62 mflo $t3
63 j label0
64
65 label0:#print the result & exit the program
66 li $v0,4
67 la $a0,res
68 syscall
69
70 move $a0,$t3
71
72 li $v0,1
73 syscall
74

```

## 5 EXO 05

Exercice n°5 :

- Ecrire un programme MIPS qui lit une **température** en degrés Celsius et affiche son équivalent en Fahrenheit.

### 5.1 Solution Exo 05

```
exo5.txt
1  #Copyright Charaf Tchi
2
3  #Regle: Fahrenheit=(degrés Celsius*1,8)+32
4  .data
5  degree:.asciiz "Entrer la dégree celsius: "
6  f:.asciiz "\nLa valeur en F est: "
7  double1:.double 1.8
8  double2:.double 32.0
9  .text
10 main:
11 #####
12 li $v0,4
13 la $a0,degree #Print->Entrer la dégree celsius:
14 syscall
15
16 ##### Read double number
17 li $v0,7
18 syscall
19
20 ldc1 $f4,double1 # $f4=1,8
21
22 ldc1 $f6,double2 # $f6=32
23
24 mul.d $f12,$f0,$f4 # $f12=$f0*1,8
25
26 add.d $f12,$f12,$f6 # $f12=$f12*32
27
28 ##### Print-> La valeur en F est:
29 li $v0,4
30 la $a0,f
31 syscall
32 ##### Print double number (in $f12)
33 li $v0,3
34 syscall
```

## 6 EXO 06

Exercice n°6 :

- Ecrire un programme MIPS qui permet de convertir et d'afficher en octets, kilo octets, méga octets et giga octets un nombre donné en **bits**.

## 6.1 Solution Exo 06

```
EXO6.txt
1  #Copyright Charaf Tchi
2  #Regle: 1 octet=8 bit      1 ko=1024 octet   1 mb=1024 ko   1 gb=1024 mb
3  .data
4  str: .ascii "Entre la valeur en bits: "
5  str1: .ascii "\nLa valuer en octets est: "
6  str2: .ascii "\nLa valuer en kilo octets est: "
7  str3: .ascii "\nLa valuer en miga octets est: "
8  str4: .ascii "\nLa valuer en giga octets est: "
9  double1: .double 8.0
10 double2: .double 1024.0
11 .text
12
13 main:
14 ##### Print->Entre la valeur en bits:
15 li $v0,4
16 la $a0,str
17 syscall
18
19 ##### Read double number (save in $f0)
20 li $v0,7
21 syscall
22
23 movf.d $f2,$f0 # $f2=$f0
24
25 ldc1 $f4,double1 #load $f4=8
26
27 div.d $f6,$f2,$f4 # $f4=$f2/8
28
29 movf.d $f12,$f6 # $f12=$f6
30
31 ##### print->La valuer en octets est:
32 li $v0,4
33 la $a0,str1
34 syscall
35
36 ##### print result of octect in double
37 li $v0,3
38 syscall
39
40 ldc1 $f8,double2 # load $f8=1024
41
42 div.d $f10,$f6,$f8 # $f10=$f6/1024
43
44 movf.d $f12,$f10 # $f12=$f10
45
46 ##### print-> La valuer en kilo octets est:
47 li $v0,4
48 la $a0,str2
49 syscall
50
51 ##### Print result of K0 in double
52 li $v0,3
53 syscall
54
55 ##### $f14=result of K0($f10)/$f8(1024) then $f12=$f14
56 div.d $f14,$f10,$f8
57 movf.d $f12,$f14
58
```

```

59 ##### Print->La valuer en Mega octets est:
60 li $v0,4
61 la $a0,str3
62 syscall
63
64 ##### Print result of MO in double
65 li $v0,3
66 syscall
67
68 ##### $f16=result of MO($f14)/$f8(1024) then $f12=$f16
69 div.d $f16,$f14 , $f8
70 movf.d $f12,$f16
71
72 ##### Print->La valuer en Gega octets est:
73 li $v0,4
74 la $a0,str4
75 syscall
76
77 ##### Print result of GO in double
78 li $v0,3
79 syscall

```