

# **Serie TP 02 avec Solution**

## **Module AO**

Univ Guelma

Promo 2017-2018

## Table des matières

<b>1</b>	<b>EXO 1</b>	<b>1</b>
1.1	Solution Exo 1 . . . . .	1
<b>2</b>	<b>EXO 2</b>	<b>2</b>
2.1	Solution Exo 2 . . . . .	2
<b>3</b>	<b>EXO 3</b>	<b>3</b>
3.1	Solution Exo 3 . . . . .	3

# 1 EXO 1

## Exercice n°1 :

Ecrire un programme en assembleur Mips qui fait lire trois entiers A, B et C. le programme affiche à la sortie la valeur Minimale et la valeur Maximale.

## 1.1 Solution Exo 1

```

1  .data
2  str:.asciiz "Entrer A: "
3  str1:.asciiz "Entrer B: "
4  str2:.asciiz "Entrer C: "
5  stp:.asciiz "Le max est: "
6  stp1:.asciiz "\nLe min est: "
7  .text
8  li $v0,4
9  la $a0,str
10 syscall
11 li $v0,5
12 syscall
13 move $t1,$v0 # $t1=a
14
15 li $v0,4
16 la $a0,str1
17 syscall
18 li $v0,5
19 syscall
20 move $t2,$v0 # $t2=b
21
22 li $v0,4
23 la $a0,str2
24 syscall
25 li $v0,5
26 syscall
27 move $t3,$v0 # $t3=c
28 #####
29 bgt $t1,$t2 label1 # if a>b
30 move $t4,$t2 # else # max=b
31 j labe2
32 label1:
33 move $t4,$t1 # max=a
34
35 labe2:
36 bgt $t3,$t4 labe3
37 move $t4,$t4
38 j lab
39
40 labe3:

```

```

39
40 labe3:
41 move $t4,$t3
42
43 lab:
44 li $v0,4
45 la $a0,stp
46 syscall
47 move $a0,$t4
48 li $v0,1
49 syscall
50
51
52
53 blt $t1,$t2,labe4
54 move $t5,$t2
55 j labe5
56 labe4:
57
58 move $t5,$t1
59 labe5:
60 blt $t3,$t5 labe6
61 move $t5,$t5
62 j labe
63 labe6:
64 move $t5,$t3
65
66
67 labe:
68 li $v0,4
69 la $a0,stp1
70 syscall
71 move $a0,$t5
72 li $v0,1
73 syscall
74
75
76
77
78

```

## 2 EXO 2

### Exercice n°2 :

Ecrire un programme en assembleur Mips qui calcule la somme de N premiers nombres entiers.

Exemple : N=5 , alors Somme = 5+4+3+2+1.

### 2.1 Solution Exo 2

```
1 .data
2 val_n:.asciiz "Entrer N: "
3 soom:.asciiz "La somme = "
4 .text
5 main:
6 li sv0,4
7 la $a0,val_n
8 syscall
9 la sv0,5 #entre la valeur de N
10 syscall
11 move $t1,sv0 # $t1=n
12
13 li $t2,1 # i=0
14 li $t3,0 # s=0
15 li $t4,1 # 1
16 while:
17 bgt $t2,$t1,done # if i>n ==> done
18
19 add $t3,$t3,$t2 # else s=s+i
20 add $t2,$t2,$t4 # i=i+1
21 j while
22 done:
23 li sv0,4
24 la $a0,soom
25 syscall
26 move $a0,$t3 # affiche la resulta
27 li sv0,1
28 syscall
29
```

Line 29, Column 1 Spaces: 2 MIPS

## 3 EXO 3

### Exercice n°3 :

Ecrire un programme en assembleur Mips qui calcule la somme de N nombres entiers quelconques.

Exemple : N=3, alors Somme = 15+8+36.

### 3.1 Solution Exo 3

```
untitled
1  .data
2  val:.asciiz "Entrer N: "
3  val:.asciiz "Entrer v: "
4  res:.asciiz "La somme = "
5  .text
6  main:
7  li $v0,4
8  la $a0,val
9  syscall
10 la $v0,5
11 syscall
12 move $t1,$v0
13 li $t2,1 # i=1
14 li $t3,0 # s=0
15 li $t4,1 # 1
16 while:
17 bgt $t2,$t1,done
18 li $v0,4
19 la $a0,val # enter val
20 syscall
21 la $v0,5
22 syscall
23 move $t5,$v0
24 add $t3,$t3,$t5 #s=s+val
25 add $t2,$t2,$t4 # i=i+1
26 j while
27 done:
28 li $v0,4
29 la $a0,res
30 syscall
31 move $a0,$t3
32 li $v0,1
33 syscall
34
35 |
```

Line 35, Column 1 Spaces: 2 MIPS