

Serie TP 03 avec solution

Module AO

Univ Guelma

Promo 2017-2018

1 Exercice 01 (Juste introduction)

Exercice n°1: Que fait ce programme ?

```
.data
    list: .word 2,3,5,7,11,13,17,19,23,29
    size: .word 10

.text
main:
    lw $t3,size
    la $t1,list
    li $t2,1
print_loop:
    beq $t2,$t3,print_loop_end
    lw $a3, 0($t1)
    li $v0,1
    move $a0,$a3
    syscall
    addi $t2,$t2,1
    addi $t1,$t1,4
    j print_loop
print_loop_end:
    li $v0,10
    syscall
```

2 Exercice 02

Exercice n°2 :

Écrire un programme en assembleur Mips déclarant un tableau initialisé de 10 entiers et en recherchant le plus petit élément. [Solution : `Tab_Min.asm`]

2.1 Solution exercice 02

```
.data
list:.word 202,10,5,7,11,13,17,19,23,29
size:.word 10
affiche :.asciiz "LE MIN EST :"
.text
main:
lw $t3,size
la $t1,list
li $t2,2
lw $t4,0($t1) # min = $a2

print_loop:
bgt $t2,$t3,print_loop_end

lw $a3,0($t1)
ble $a3,$t4,lab

addi $t2,$t2,1
addi $t1,$t1,4

j print_loop

lab:
move $t4,$a3
addi $t2,$t2,1
addi $t1,$t1,4
j print_loop
syscall

print_loop_end:
li $v0,4
la $a0,affiche
syscall

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

3 Exercice 03

Exercice n°3 :

Ecrire un programme en assembleur Mips déclarant un tableau initialisé de 10 entiers et calcule la somme de n élément du tableau. [Solution : **Tab_Som.asm**]

3.1 Solution exercice 03

```
.data
list:.word 2,3,5,7,11,13,17,19,23,29
size:.word 10
somme:.asciiz " la somme est :"
.text

main:
lw $t3,size
la $t1,list
li $t2,1
li $t6,0
print_loop:
bgt $t2,$t3,print_loop_end
lw $t5,0($t1)
add $t6,$t6,$t5
addi $t2,$t2,1
addi $t1,$t1,4
j print_loop
print_loop_end:
li $v0,4
la $a0,somme
syscall

move $a0,$t6
li $v0,1
syscall
```

4 Exercice 04

Exercice n°4 :

Ecrire un programme en assembleur Mips déclarant un tableau initialisé de 10 entiers et calcule le produit de n élément du tableau. [Solution : **Tab_Prod.asm**]

4.1 Solution exercice 04

```
.data
list:.word 2,3,5,7,11,13,17,19,23
size:.word 9
pro:.asciiz " produit est :"
.text

main:
lw $t3,size
la $t1,list
li $t2,1
li $t6,1
print_loop:
bgt $t2,$t3,print_loop_end
lw $t5,0($t1)
mul $t6,$t6,$t5
addi $t2,$t2,1
addi $t1,$t1,4
j print_loop

print_loop_end:
li $v0,4
la $a0,pro
syscall

move $a0,$t6
li $v0,1
syscall
```