

## Solution TP Travail à domicile Architecture des ordinateurs

---

### Exercice 1 (10 points)

Écrire le code MIPS qui lit de l'utilisateur une valeur. Le programme doit afficher si la valeur est positive ou négative.

```
.data
msg1 : .asciiz "Donnez S.V.P. une valeur : "
msg2 : .asciiz "La valeur est positive."
msg3 : .asciiz "La valeur est négative."
.text
li $v0 , 4 # pour afficher msg1
la $a0 , msg1 #
syscall #
li $v0 , 5 # pour lire un entier
syscall # l' entier lu se met automatiquement sur $v0
move $t0 , $v0 # déplacer l' entier lu vers un registre de travail
$t0
bltz $t0 , pos # vérifier si ($t0<0) ?, puis branchement si vrai
li $v0 , 4 # si ($t0>=0)
la $a0 , msg2 # afficher msg2
syscall
j end_if
pos : li $v0 , 4 # afficher msg3 [ si ($t0<0) ]
la $a0 , msg3
syscall
end_if:
li $v0 , 10 # mieux ajouter cette partie qui indique fin d'exécution
syscall
```

### Exercice 2 (5 points)

Écrire le programme MIPS qui lit de l'utilisateur une valeur, et affiche ensuite les 5 premiers multiples de cette valeur.

```
.data
msg1 : .asciiz "Donnez S.V.P. une valeur : "
msg2 : .asciiz "Les multiples sont : "
msg3 : .asciiz " "
.text
li $v0 , 4
la $a0 , msg1
syscall
li $v0 , 5
syscall
move $t9 , $v0
li $t0 , 1
li $t2 , 0
li $t1 , 5
li $v0 , 4
la $a0 , msg2
```

```

        syscall
loop :   bgt $t0 , $t1 , lab
        add $t2 , $t2 , $t9
        addi $t0 , $t0 , 1
        li $v0 , 1
        move $a0 , $t2
        syscall
        li $v0 , 4
        la $a0 , msg3
        syscall
        j loop
lab :   li $v0 , 10
        syscall

```

### Exercice 3 (5 points)

Ecrire le programme MIPS qui lit de l'utilisateur 5 valeurs et les sauvegarde dans un tableau, puis calcule et affiche la somme des valeurs du tableau.

```

.data
tab : .space 20
msg1 : .asciiz "Donnez S.V.P. une valeur : "
msg2 : .asciiz "La somme est : "

.text
li $t0,0      # indice debut
li $t1,5      # indice fin
la $t2,tab

loop1:        bge $t0,$t1, pos1 # une fois $t0 devient égale à $t1, alors faire
un saut au pos1
li $v0,4      # pour affficher msg1
la $a0,msg1
syscall

li $v0,5      # lire entier
syscall

sw $v0, 0($t2)
addi $t0,$t0,1 # incrimerter l'indice
addi $t2,$t2,4 # passser à la prochaine case du tableau
j loop1

pos1: li $t0,0 # reinitialiser l'indice au debut
la $t2,tab

loop2:        bge $t0,$t1, pos2
lw $s0, 0($t2)
add $s2,$s2,$s0
addi $t0,$t0,1 # incrimerter l'indice
addi $t2,$t2,4 # passser à la prochaine case du tableau
j loop2

pos2: li $v0,4 # pour affficher msg2
la $a0,msg2
syscall

li $v0,1
move $a0,$s2
syscall

```