

TP2

Archeticture des ordinateurs

1. Ecrire un code MIPS qui permet de calculer la somme 6+8, puis afficher le résultat.

```
li $t0,6
li $t1,8
add $t2,$t0,$t1

li $v0,1          # pour afficher la somme 6+8
move $a0,$t2
syscall
```

2. Ecrire un code MIPS qui réserve un espace mémoire pour les entiers A=30, B=48, calculer la somme A+B puis afficher le résultat.

```
.data
a: .word 30
b: .word 48
.text
lw $t0,a
lw $t1,b
add $t2,$t0,$t1

li $v0,1          # pour afficher la somme des entiers a+b
move $a0,$t2
syscall
```

3. Ecrire un code MIPS qui réserve un espace mémoire pour les réels doubles A=30.3, B=48.1, puis calculer la somme A+B et afficher le résultat.

```
.data
a: .double 30.3
b: .double 48.1
.text
l.d $f0,a
l.d $f2,b
add.d $f4,$f2,$f0

li $v0,3          # pour afficher la somme des doubles a+b
mov.d $f12,$f4
syscall
```

4. Ecrire le code MIPS qui déclare une chaîne de caractère ch='SALEM', puis afficher la chaîne.

```
.data
ch: .asciiz "Salem"
.text
li $v0,4
la $a0,ch
syscall
```

5. Ecrire un code MIPS qui permet de lire un entier et le stocker sur mémoire.

```
.data
    msg : .asciiz "Donnez S.V.P. un entier : "

.text
    li $v0 , 4          # pour afficher msg
    la $a0 , msg        #
    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
```

6. Ecrire un code MIPS qui permet de lire un float et le stocker sur mémoire.

```
.data
    msg : .asciiz "Donnez S.V.P. un float : "

.text
    li $v0 , 4          # pour afficher msg
    la $a0 , msg        #
    syscall             #

    li $v0 , 6          # pour lire un entier
    syscall             # l'entier lu se met automatiquement sur $v0
    mov.s $f10 , $f0    # déplacer le réel lu vers un registre de travail $f10
```

7. Ecrire un code MIPS qui permet de lire une chaîne de caractère et la stocker sur mémoire.

```
.data
    msg : .asciiz "Donnez S.V.P. une chaîne de caractère : "
    msg1 : .asciiz "La chaîne que vous avez saisie est : "
    ch: .space 20

.text
    li $v0 , 4          # pour afficher msg
    la $a0 , msg        #
    syscall             #

    li $v0 , 8          # pour lire une chaîne de caractère
    la $a0, ch          # Mettre l'adresse de la chaîne sur $a0
    li $a1, 20          # la taille max de la chaîne
    syscall             # l'adresse de la chaîne lue se met automatiquement sur $a0

    li $v0 , 4          # pour afficher msg
    la $a0 , msg1       #
    syscall             #

    li $v0, 4           # Afficher la chaîne lue pour vérifier
    la $a0, ch          #
    syscall
```

8. Ecrire un code MIPS qui permet de lire un tableau de 10 entiers, puis stocker toutes les valeurs sur mémoire.

```

.data
tab : .space 40

msg1 : .asciiz "Donnez S.V.P. une valeur : "

.text
    li $t0 , 0
    li $t1 , 10
    la $t2 , tab

loop1: bge $t0 , $t1 , lab_sortir

    li $v0 , 4
    la $a0 , msg1
    syscall

    li $v0 , 5
    syscall

    sw $v0 , 0($t2)          # mettre l'entier sur l'@ : tab , tab+4 , tab+8 ,...

    addi $t0 , $t0 , 1
    addi $t2 , $t2 , 4

    j loop1

lab_sortir:

```