

## Série de TD n°2

---

### Questions :

- a) Quels sont les caractéristiques principales du processeur MIPS R3000 ?
- b) Quels sont les différents types d'instructions manipulés dans MIPS ?
- c) Quels sont les directives utilisées pour déclarer des données en mémoire ?
- d) Donnez l'instruction de déclaration pour chaque requête suivante :
  - Réserve d'un entier de 32 bits
  - Réserve de plusieurs mots consécutifs initialisés
  - Réserve d'une chaîne de caractères terminée par un caractère nul
  - Réserve de plusieurs octets sans initialisation

### Exercice 1 : Que fait ce programme ?

```
.data
    Var1: .asciiz "Entrez la .....: " #
    Var2: .asciiz "Entrez la .....: " #
    result1: .asciiz "..... est: " #

.text
main:
    li $v0, 4          #
    la $a0, prompt1   #
    syscall
    li $v0, 5          #
    syscall
    move $t0, $v0      #
    li $v0, 4          #
    la $a0, prompt2   #
    syscall
    li $v0, 5          #
    syscall
    move $t1, $v0      #
    add $t2, $t0, $t1  #
    add $t2, $t2, $t2  #
    li $v0, 4          #
    la $a0, result1   #
    syscall
    move $a0, $t2      #
    li $v0, 1          #
    syscall
    li $v0, 10         #
    syscall
```

**Exercice n°2** : Corriger le programme et dites ce qu'il fait ?

```
.data
prompt: .word "Entrez un entier: "      #
result1: .asciiz ".....: "          #
.data
main:
    li $v0, 6          #
    la $a2, prompt    #
    system call
    li $v0, 5          #
    system call
    move $t10, $v0     #
    Add $t1, $t0, $t0  #
    li $v1, 4          #
    la $a1, result1   #
    system call
    move $a1, $t1      #
    li $v0, 1          #
    system call
    li $v0, 1          #
    system call
```

**Exercice n°3** : Ecrire un programme *Operations* en assembleur MIPS permettant de saisir deux entiers et d'afficher successivement leur somme, leur différence, leur produit et leur quotient.

---