



Corrigé du TP N°1 – Architecture des ordinateurs

Exercice 1 :

Ecrire le code MIPS qui permet de :

1. calculer la somme 5+7 ;
2. afficher le résultat.

```
1 # Solution Exercice 1
2 # Il n'y aura pas de données à déclarer donc pas de section data, on utilise Le mode immédiat
3 .text
4 main:
5 li $t0,5      # charger la valeur 5 dans Le registre t0
6 li $t1,7      # charger la valeur 7 dans Le registre t1
7 add $t2,$t1,$t0 # Faire L'addition de t0+t1 et mettre Le résultat dans Le registre t2
8 move $a0,$t2  # Le contenu de t2 est copié dans Le registre a0
9 li $v0,1      # charger Le registre v0 avec Le code 1 pour afficher entier print_int
10 syscall      # appel système pour L'opération affichage entier
11 #Fin du programme
12 li $v0,10
13 syscall
```

Exercice 2 :

Ecrire le code MIPS qui permet de :

1. réserver un espace mémoire pour les entiers A=15, B=35 ;
2. déclarer la chaîne de caractères Str=« La somme de A=15 et B=35 est : »
3. calculer la somme A+B ;
4. afficher la chaîne de caractère Str sur la console ;
5. afficher le résultat.

```
1 # Solution Exercice 2
2 .data        # Section données
3 A: .word 15  # Déclaration de variable A=15
4 B: .word 35  # Déclaration de variable B=35
5 Str: .asciiz "La somme de A=15 et B=35 est: " # Déclaration de chaîne de caractère Str
6 .text
7 main:       # Section Code
8 lw $t0,A    # charger La valeur de La variable A dans Le registre t0
9 lw $t1,B    # charger La valeur de La variable B dans Le registre t1
10 add $t2,$t1,$t0 # Faire L'addition de t0+t1 et mettre Le résultat dans Le registre t2
11 li $v0,4    # Charger Le registre $v0 avec Le code 4 pour affichage caractère (print_string)
12 la $a0,Str  # Mettre L'adresse de La chaîne de caractère Str dans Le registre $a0
13 syscall    # Appel système pour L'opération affichage chaîne de caractère
14 move $a0,$t2 # Le contenu de t2 est copié dans Le registre a0
15 li $v0,1    # Charger Le registre v0 avec Le code 1 pour afficher entier (print_int)
16 syscall    # Appel système pour L'opération affichage entier
17 #Fin du programme
18 li $v0,10
19 syscall
```

Exercice 3 :

Ecrire le code MIPS qui permet de :

1. réserver un espace mémoire pour les réels double A=15.4, B=35.2 ;
2. déclarer une chaîne de caractères Str=« La somme de A=15.4 et B=35.2 est : »
3. calculer la somme A+B ;
4. afficher la valeur de la chaîne de caractères Str, ensuite le résultat.

```
1 # Solution Exercice 3
2 .data # Section données
3 A: .double 15.4 # Déclaration de variable A=15.4
4 B: .double 35.2 # Déclaration de variable B=35.2
5 Str: .asciiz "La somme de A=15.4 et B=35.2 est: " # Déclaration de chaîne de caractère Str
6 .text
7 main: # Section Code
8 l.d $f0,A # charger la valeur de la variable A dans le registre t0
9 l.d $f2,B # charger la valeur de la variable B dans le registre t1
10 add.d $f12,$f2,$f0 # Faire l'addition de f0+f2 et mettre le résultat dans le registre f12
11 li $v0,4 # Charger le registre $v0 avec le code 4 pour affichage caractère (print_string)
12 la $a0,Str # Mettre l'adresse de la chaîne de caractère Str dans le registre $a0
13 syscall # Appel système pour l'opération affichage chaîne de caractère
14 li $v0,3 # Charger le registre v0 avec le code 3 pour afficher double (print_double)
15 syscall # Appel système pour l'opération affichage double
16 #Fin du programme
17 li $v0,10
18 syscall
```

Exercice 4 :

Ecrire le code MIPS qui permet de :

1. déclarer les chaînes de caractères Message1=« Donner un nombre entier : », Message2=«L'entier que vous avez donné est : »
2. afficher la chaîne de caractère Message1 sur la console ;
3. lire un nombre entier au clavier ;
4. afficher la chaîne de caractère Message2 sur la console suivie de la valeur de l'entier saisi au clavier

```
1 # Solution Exercice 4
2 .data # Section données
3 Message1: .asciiz "Donner un nombre entier : " # Déclaration de chaîne de caractère Message1
4 Message2: .asciiz "L'entier que vous avez donné est : " # Déclaration de chaîne de caractère Message2
5 .text
6 main: # Section Code
7 li $v0,4 # Charger le registre $v0 avec le code 4 pour affichage caractère (print_string)
8 la $a0,Message1 # Mettre l'adresse de la chaîne de caractère Message1 dans le registre $a0
9 syscall # Appel système pour l'opération affichage chaîne de caractère
10 li $v0,5 # Charger le registre $v0 avec le code 5 pour Lire entier (read_int)
11 syscall # Appel système pour lecture
12 move $t0,$v0 # Mettre la valeur de l'entier lu ($v0) dans le registre $t0
13 li $v0,4 # Charger le registre $v0 avec le code 4 pour affichage caractère (print_string)
14 la $a0,Message2 # Mettre l'adresse de la chaîne de caractère Message2 dans le registre $a0
15 syscall # Appel système pour l'opération affichage chaîne de caractère
16 move $a0,$t0 # Mettre la valeur du registre $t0 dans le registre $a0
17 li $v0,1 # Charger le registre $v0 avec le code 1 pour afficher entier (print_int)
18 syscall # Appel système pour l'opération affichage entier
19 #Fin du programme
20 li $v0,10
21 syscall
```

Exercice 5 :

Ecrire le code MIPS qui permet de :

1. déclarer les chaînes de caractères Chaine=« Donner un nombre flottant : », Sortie=« La valeur saisie est : »
2. afficher la chaîne de caractère Chaine sur la console ;
3. lire un nombre flottant au clavier ;
4. afficher la chaîne de caractère Sortie sur la console suivie de la valeur du nombre flottant saisi au clavier.

```
1 # Solution exercice5
2 .data # Section données
3 chaine:.asciiz"donner un nombre flottant:"
4 sortie:.asciiz"la valeur saisie est :"
5 .text
6 main: # Section Code
7 la $a0,chaine # Mettre l'adresse de la chaîne de caractère chaine dans le registre $a0
8 li $v0,4 # Charger le registre $v0 avec le code 4 pour affichage chaîne de caractère (print_string)
9 syscall # Appel système pour l'opération affichage chaîne de caractère
10 li $v0,6 # Charger le registre $v0 avec le code 6 pour lire flottant (read_float)
11 mov.s $f0,$f12 # Mettre la valeur de du flottant lu ($f12) dans le registre $f0
12 syscall # Appel système
13 la $a0,sortie # Mettre l'adresse de la chaîne de caractère sortie dans le registre $a0
14 li $v0,4 # Charger le registre $v0 avec le code 4 pour affichage chaîne de caractère (print_string)
15 syscall # Appel système
16 mov.s $f12,$f0 # Mettre le contenu du registre $f0 dans le registre $f12
17 li $v0,2 # Charger le registre $v0 avec le code 2 pour afficher flottant (print_float)
18 syscall # Appel système pour affichage
19 #Fin du programme
20 li $v0,10
21 syscall
```