

Examen de fin de semestre 1

Questions de cours : (06 points)

1. Les instructions dans les architectures RISC sont codées sur le même nombre de bits. Dites pourquoi cela est-il avantageux?
2. Quelle est l'utilité des registres : \$cause, \$Pc, \$s6, \$v1
3. Cette instruction est de format R. Donnez le code du registre \$Rd.

00000001011011000111000000100000

4. Dans un tableau on veut stocker 30 éléments à partir de l'adresse 0x15C02001. Quelle est l'adresse du 8^{ème} élément ?

Exercice 1 : (04 points)

Ecrivez un programme en MIPS R3000 qui **lit** un nombre ensuite **calcule** et **affiche** sa factorielle.

Nb : la factorielle de 7 est notée 7 ! et vaut : 7x6x5x4x3x2x1

Exercice 2 : (05 points)

```
.data
x: .word 3
y: .byte 5
answer: .int 0

.text
main
lw $a0, x
lw $a1, y
jr power
sw $v0, answer
li $v0, 10
.end main

power:
li $v0, 0
li $t0, 0
bcl:
mul $v0,$v0, $a0
add $t0, $t0, 1
blt $t0, $a1, power
jr $a0
.end power
```

- Ce code en MIPS contient des erreurs. Corrigez-les pour obtenir un programme correct qui calcule X puissance Y.
- Modifiez-le pour obtenir un autre qui **lit** les valeurs X et Y et **affiche** ensuite le résultat sur la console.

Exercice 3 : (05 points)

Le programme suivant permet de remplir une pile à partir d'un tableau.

- Ajoutez les commentaires manquants.
- Affichez les éléments du tableau sur la console.
- Ecrivez la boucle **PopLoop** qui permet de retirer les éléments **positifs** de la pile et de les mettre dans ce tableau.

```
.data
    array: .word 1, -3, 2, 14, 9, -11, 18, -6, -17, 22
    length: .word 10
.text
main:

la $t0, array      # .....
li $t1, 0          # .....
lw $t2, length

pushLoop:
lw $t4, ($t0)      # .....
subu $sp, $sp, 4
sw $t4, ($sp)      # .....
add $t1, $t1, 1
add $t0, $t0, 4
blt $t1, $t2, pushLoop # .....

li $v0, 10
syscall           # .....

.end main
```

Bon Courage