Université d'Alger 1

Faculté des Sciences
Département Mathématiques et Informatique

Rattrapage L2-ADO
11-Septembre-2021

Nom : Prénom : Groupe : Matricule :		/ 20

Exercice 1 : (10 points)

Pour chacune des pseudo-instructions suivantes, donnez une **séquence minimale** d'instructions MIPS réelles pour accomplir la même chose. Vous ne pouvez utiliser que le registre $at en tant que registre temporaire pour vos transformations

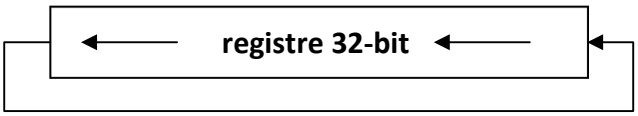
a)
```
abs   $s1, $s2          # $s1 = abs( $s2 )
addu $s1, $zero, $s2
bgez $s2, next
subu $s1, $zero, $s2
next:
```

b)
```
addiu $s1, $s2, imm32   # imm32 est un immediat 32-bits
lui  $at, upper16
ori $at, $at, lower16
addu $s1, $s2, $at
```

c)
```
bleu $s1, $s2, Label    # branch si inférieur ou égal (unsigned)
sltu $at, $s2, $s1
beq  $at, $zero, Label
```

d)
```
bge $s1, imm32, Label   # imm32 est un immediat 32-bits
lui $at, upper16
ori $at, $at, lower16
slt $at, $s1, $at
beq $at, $zero, Label
```

e)
```
rol $s1, $s2, 5         # rol = rotation à gauche de $s2 par 5 bits
srl $at, $s2, 27
sll $s1, $s2, 5
or  $s1, $s1, $at
```



Exercice 2 : (10 points)

Le code suivant traite deux tableaux et produit un résultat important dans le registre $v0. Supposons que chaque tableau se compose de N mots, les adresses de base des tableaux A et B sont stockées dans les registres $a0 et $a1 respectivement, et leurs tailles sont stockées dans les registres $a2 et $a3, respectivement.

```
          sll     $a2, $a2, 2
          sll     $a3, $a3, 2
          addu    $v0, $zero, $zero
          addu    $t0, $zero, $zero
outer:    addu    $t4, $a0, $t0
          lw      $t4, 0($t4)
          addu    $t1, $zero, $zero
inner:    addu    $t3, $a1, $t1
          lw      $t3, 0($t3)
          bne     $t3, $t4, skip
          addiu   $v0, $v0, 1
skip:     addiu   $t1, $t1, 4
          bne     $t1, $a3, inner
          addiu   $t0, $t0, 4
          bne     $t0, $a2, outer
```

1/ Décrivez ce que fait le code ci-dessus et ce qui sera retourné dans le registre $v0.

Ce code compare chaque élément du premier tableau à tous les éléments du second tableau. Il compte le nombre d'éléments correspondants entre les deux tableaux et retourne cette valeur dans. $v0.

2/ Donnez une version en langage C du code ci-dessous

```
/*
int  i <-> $t1
int  j <-> $t2
int  N <-> $a2
int  M <-> $a3
int* A <-> $a0
int* B <-> $a1
int count <-> $v0
*/

count = 0;
for(i=0 ; i<N ; ++i)
{
    for(j=0 ; j<M ; ++j)
    {
          if( A[i] == B[i] )
          {
               count++;
          }
    }
}
```