

Nom :		/ 20
Prénom :		
Matricule :		

### Exercice 1 : (4 points = 2 x 2 points par instruction et commentaire correctes)

Le langage assembleur natif du MIPS n'inclut pas l'instruction « nand » (not and)

```
nand $rd, $rs, $rt    # $rd <-- !($rs & $rt)
```

Montrer comment l'assembleur pourrait remplacer le pseudo-code ci-dessus par une séquence d'instructions assembleurs natives. Utilisez le registre \$at comme registre de travail (temporaire) et expliquez la logique de votre conception par des commentaires clairs.

Remarque aucune solution proposée ne doit modifier des registres autres que le registre \$rd et, éventuellement le registre \$at.

and \$at, \$rs, \$rt	# \$at <-- \$rs & \$rt	ou	and \$at, \$rs, \$rt	# \$at <-- \$rs & \$rt
nor \$rd, \$at, \$at	# \$rd <-- !(\$at   \$at)		xori \$rd, \$at, -1	# \$rd <-- \$at ^ 0xFFFFFFFF
		ou	and \$at, \$rs, \$rt	# \$at <-- \$rs & \$rt
			nor \$rd, \$at, \$zero	# \$rd <-- !(\$at   0)

parmi d'autres combinaisons possibles : \$rd <-- !(\$rs | \$rs) | !(\$rt | \$rt) ou \$rd <-- !(\$rs | \$0) | !(\$rt | \$0)

### Exercice 2 : (16 points)

Quand le code MIPS ci-dessous est assemblé dans MARS, nous obtenons le code machine à droite :

```
.data
Sz: .word 5
List: .word 10, 20, 30, 40, 50

.text
main:
la $s0, List    #1 $s0 := 1001 0004 (l'adresse de List)
lw $s1, ($s0)  #2 $s1 := le mot à l'adresse 1001 0004
lw $s2, 4($s0) #3 $s2 := le mot à l'adresse 1001 0008

li $v0, 10
syscall
```

	adresse	Valeurs (mots)
Sz	1001 0000	0000 0005
List	1001 0004	0000 000A
	1001 0008	0000 0014
	1001 000C	0000 001E
	1001 0010	0000 0028
	1001 0014	0000 0032

a/ (6 points = 3 x (1 point pour une case correcte + 1 point pour un commentaire correcte) )

Quelles valeurs sont placées dans chacun des registres suivants quand le programme est exécuté ? Donnez votre réponse en hexadécimal (comme indiqué ci-dessus) et commentez les lignes #1, #2 et #3.

\$s0 :  (2 points)

\$s1 :  (2 points)

\$s2 :  (2 points)

b/ (2 points)

Quelle valeur serait placée dans le registre \$s3 une fois l'instruction suivante est exécutée ?

la \$s3, \$s2

1001 0000

c/ (8 points = 2 points pour la logique + 3 points pour les instructions + 3 points pour les commentaires)

Supposons que les instructions #1 à #3 viennent d'être exécutées. Insérer le code assembleur MIPS pour calculer l'adresse du premier octet de la mémoire à la fin du tableau *List* et sauvegarder cette adresse dans le registre \$s4.

Votre solution doit fonctionner indépendamment de la taille du tableau *List*, c.-à-d. l'écriture de la valeur 0x10010018 dans \$s4 N'EST PAS une solution acceptable.

<code>lw \$s4, \$s2</code>	# récupérer le nombre d'éléments dans le tableau <i>List</i>
<code>sll \$s4, \$s4, 2</code>	# multiplier par 4 pour avoir la taille du tableau en octets
<code>addu \$s4, \$s0, \$s4</code>	# additionner la taille du tableau à son adresse de base # qui est stocké dans le registre \$s0