

Nom :		/ 20
Prénom :		
Matricule :		

Exercice 1 : (8 points)

Corriger et compléter le code assembleur MIPS ci-dessous.

```

1  .data
2  Str: .asciiz "-> "
3  Size: .word 3
4  value: .word 0x0CB0CEFA, 0xE5A55ED1, 0xEFBEADDE
5
6  .text
7  main:
8      la $s1, Str          # $s1 = l'adresse de Str          (1 point)
9      lw $s0, Size        # $s0 = Size[0]; la taille du tableau "value" (1 point)
10
11     sll $s0, $s0, 2      # $s0 = $s0 * 4                    (0.5 point)
12
13     DO: addi $s0, $s0, -4 # $s0 = $s0 - 4                    (1 point)
14         blt $s0, $zero, DONE # if($s0 < 0) goto DONE
15
16         lw $t0, value($s0) # $t0 = value[s0]
17
18         addi $sp, $sp, -8   # allocation d'espace sur la pile (0.5 point)
19         sw $t0, ($sp)      # sauvegarde de $t0 dans la pile (0.5 point)
20         sw $ra, 4($sp)     # sauvegarde de $ra dans la pile (0.5 point)
21         add $a0, $zero, $t0 #
22         jal func          # appel de la fonction "func"
23         add $t1, $zero, $v0 #
24         lw $ra, 4($sp)    # restaure $ra depuis la pile (0.5 point)
25         lw $t0, ($sp)     # restaure $t0 depuis la pile (0.5 point)
26         addi $sp, $sp, 8   # libérer l'espace dans la pile (0.5 point)
27
28         add $a0, $zero, $t0 # affichage d'un entier en utilisant
29         addi $v0, $zero, 34 # le format hexadécimal
30         syscall           #
31
32         add $a0, $zero, $s1 # affichage de la chaine de (0.5 point)
33         addi $v0, $zero, 4  # caractères "-> "
34         syscall           #
35
36         add $a0, $zero, $t1 # affichage d'un entier en utilisant
37         addi $v0, $zero, 34 # le format hexadécimal
38         syscall           #
39
40         add $a0, $zero, '\n' # affichage du caractère
41         addi $v0, $zero, 11 # '\n' (saut de ligne) (1 point)
42         syscall           #
43
44         j DO              # saut à DO (boucle)
45     DONE: li $v0, 10      # exit()
46         syscall         #

```

Exercice 2 : (4 points)

Compléter le code assembleur MIPS de la fonction "func" ci-dessous.

```
func:
    addi $sp, $sp, -12           # Allocation dans la pile pour la sauvegarde      (0.5 point)
    sw $s0, 0($sp)              # Sauvegarde dans la pile de la variable locale $s0      (0.5 point)
    sw $s1, 4($sp)              # Sauvegarde dans la pile de la variable locale $s1      (0.5 point)
    sw $s2, 8($sp)              # Sauvegarde dans la pile de la variable locale $s2      (0.5 point)

    add $s0, $zero, $a0          # $s0 = $a0
    add $s1, $zero, $zero        # $s1 = 0
    addi $s2, $zero, 4           # $s2 = 4

BCL:  addi $s2, $s2, -1           # $s2 = $s2 - 1
      blt $s2, $zero, QUIT       # if( $s2 < 0) goto QUIT

      and $t0, $s0, 0xFF          # $t0 = $s0 and 0xFF
      srl $s0, $s0, 8             # décalage à droite logique de $s0 de huit positions
      sll $s1, $s1, 8            # décalage à gauche logique de $s1 de huit positions
      or  $s1, $s1, $t0          # $s1 = $s1 or $t0

      j BCL                       # goto BCL

QUIT: add $v0, $zero, $s1        # $v0 = $s1

      lw $s0, 0($sp)             # restaurer depuis la pile la valeur de $s0      (0.5 point)
      lw $s1, 4($sp)             # restaurer depuis la pile la valeur de $s1      (0.5 point)
      lw $s2, 8($sp)             # restaurer depuis la pile la valeur de $s2      (0.5 point)
      addi $sp, $sp, 12          # Libérer l'espace dans la pile                  (0.5 point)

      jr $ra                       # retour à la fonction appelante
```

Exercice 3 : (8 points)

Supposons que la fonction "func" de l'exercice 2 soit appelée à la ligne 22 par la fonction "main" de l'exercice 1.

1/ Donner les messages affichés par les instructions des lignes 28 à 38 dans la fonction "main" (3 points)

```
0xEFBEADDE -> 0xDEADBEEF      (1 point)
0xE5A55ED1 -> 0xD15EA5E5      (1 point)
0x0CB0CEFA -> 0xFACEB00C      (1 point)
```

2/ Que fait le programme ? (2 points)

Le programme inverse l'ordre des octets dans un « mot ».

3/ Quand est-ce qu'on utilise la fonction "func" ? (3 points)

On utilise la fonction "func" quand on veut permuter entre les modes « little-endian et big-endian » pour le stockage de « mots » dans la mémoire.