



Nom :		<h1>/ 20</h1>
Prénom :		
Groupe :		
Matricule :		

### Exercice 1 : (8 points)

Considérer les définitions de données suivantes :

```
.data
var1:      .byte      3, -2, 'A'
var2:      .half      1, 256, 0xffff
var3:      .word      0x3delc74, 0xff
.align 3   # aligner l'instruction suivante à une adresse multiple de 2^3
str1:      .asciiz    "ICS233"
```

- 1/ Donner le contenu de chaque octet de la mémoire allouée **en hexadécimal** pour les définitions de données ci-dessus. L'ordre **Little Endian** est utilisé pour les octets des mots et demi-mots (half). (4 points)
- 2/ Remplir la table des symboles en indiquant **toutes les étiquettes** et leurs **adresses de début**. (2 points)
- 3/ Indiquez par un **X** dans la case les octets ignorés (mais alloués) dans le segment de données. (1 point)

#### Segment de données

Adresse	Octet 0	Octet 1	Octet 2	Octet 3
0x10010000	0x03	0xfe	0x41	X
0x10010004	0x01	0x00	0x00	0x01
0x10010008	0xff	0xff	X	X
0x1001000C	0x74	0x1c	0xde	0x03
0x10010010	0xff	0x00	0x00	0x00
0x10010014	X	X	X	X
0x10010018	0x49	0x43	0x53	0x32
0x1001001C	0x33	0x33	0x00	
0x10010020				
0x10010024				
0x10010028				
0x1001002C				

#### Table des symboles

Etiquette	Adresse
var1	0x10010000
var2	0x10010004
var3	0x1001000C
str1	0x10010018

- 4/ Combien d'octets sont alloués au total (incluant les octets ignorés) dans le segment de données ? (1 point)

<b>31</b>	octets
-----------	--------

### Exercice 2 : (6 points)

Considérons un cache de type 1-associatif de 128 lignes (blocs). Une ligne de ce cache (un bloc) contient 32 octets de données.

1/ Donnez le nombre de bits d'étiquette, de bit d'index et de bits d'offset (mot + octet)

Etiquette :  (0,5 point)      Index :  (0,5 point)      Offset :  (0,5 point)

2/ Donnez le nombre nécessaire de bits pour stocker tous les bits de validité et d'étiquette pour ce cache

(0,5 point)

3/ Supposons que le cache soit initialement vide et que nous accédions aux adresses suivantes, dans cette ordre :

0x4e20, 0x4e24, 0x4e28, 0x4e30, 0x5e2c, 0x5e30, 0x5e34, 0x5e38

Donner la valeur en décimal de l'index et de l'étiquette pour chaque adresse et indiquer si l'accès est un succès ou un échec.

Adresse	Etiquette	Index	Succès ou Echec	
0x4e20	4	113	Echec (vide)	(0,5 point)
0x4e24	4	113	Succès	(0,5 point)
0x4e28	4	113	Succès	(0,5 point)
0x4e30	4	113	Succès	(0,5 point)
0x5e2c	5	113	Echec	(0,5 point)
0x5e30	5	113	Succès	(0,5 point)
0x5e34	5	113	Succès	(0,5 point)
0x5e38	5	113	Succès	(0,5 point)

### Exercice 3 : (6 points)

Pour chacune des pseudo-instructions suivantes, donner une séquence minimale d'instructions MIPS réelles pour accomplir la même tâche. Vous ne pouvez utiliser que le registre \$at comme registre temporaire.

1/ `addiu $s1, $s2, imm32`      # imm32 est un nombre sur 32-bit

`lui $at, imm3232-16`  
`ori $at, $at, imm3215-0`  
`addu $s1, $s2, $at`      (2 points)

2/ `bleu $s1, $s2, Label`      # branchement si inférieur ou égal (nombres non signés)

`sltu $at, $s2, $s1`  
`beq $at, $0, Label`      (2 points)

3/ `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`



(2 points)