

**Examen Architecture des Ordinateurs**

**Questions (2p)**

1. Soit une mémoire de 512 Mo. Donner la taille du bus d'adresse ? **29**.....
2. Soit une mémoire de 8200 octets. Donner la taille du bus d'adresse ? **14**.....

**Exercice 1 (6p)**

1. Utilisant un processeur MIPS, donner en hexadécimale les adresses de :
  - (a) 1<sup>er</sup> octet **0x0000 0000**.....
  - (b) dernier octet **0x ffff ffff**.....
2. Donner l'adresse de chaque ligne du code MIPS suivant :

<pre>.data     a: .word 5     tableau: .asciiz "batna"     b: .word 8     c: .word  .text     lw \$t1,a     lw \$t2,b     add \$a0,\$t1,\$t2     sw \$a0,c</pre>	<p>Les adresses:</p> <p><b>0x 1001 0000</b> .....</p> <p><b>0x 1001 0004</b> .....</p> <p><b>0x 1001 000c</b> .....</p> <p><b>0x 1001 0010</b> .....</p> <p><b>0x 0040 0000</b> .....</p> <p><b>0x 0040 0004</b> .....</p> <p><b>0x 0040 0008</b> .....</p> <p><b>0x 0040 000c</b> .....</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Que fait le code précédent? **c=a+b / ou calcul la somme de a et b dans c / ou autre réponse équivalente**.....

**Exercice 2 (6p)**

Sachant que les instructions consécutives : **mult \$r0, \$r1** puis **mflo \$r2** permettent de faire la multiplication des registres **\$r0, \$r1** et mettre le résultat sur le registre **\$r2**.

1. Compléter le program MIPS au-dessous qui calcule le polynôme  **$5x^2 + 12x + 97$**  et met le résultat sur la variable **y**.
2. Donner les modifications nécessaires pour que le programme puisse **lire** la valeur de **x** et **afficher** l'image **y** (sans passer par la mémoire).

<pre>.data     x : .word 4     y : .word  .text 1.      lw \$t0 , x 2.      mult \$t0, \$t0 3.      mflo \$t1 4.      li \$t2,5 5.      mult \$t1, \$t2 6.      mflo \$t1 7.      li \$t2,12 8.      mult \$t0, \$t2 9.      mflo \$t3 10.     add \$t1,\$t1,\$t3 11.     addi \$t1,\$t1,97 12.     sw \$t1,y</pre>	<p>Remplacer la ligne 1 par :</p> <pre>li \$v0 , 5 syscall move \$t0,\$v0</pre> <p>Remplacer la ligne 12 par :</p> <pre>li \$v0 , 1 move \$a0 , \$t1 syscall</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Exercice 3 (6p)

Donner le program C correspondant au code MIPS suivant :

```
.data
tab : .space 20
msg1 : .asciiz "Donnez S.V.P. une valeur : "
msg2 : .asciiz "La somme est : "

.text
li $t0 , 0      # indice i
li $t1 , 5      # taille tableau
la $t2 , tab    # pointer la case i
li $t3 , 0      # somme

loop1: bge $t0 , $t1 , lab1

    li $v0 , 4
    la $a0 , msg1
    syscall

    li $v0 , 5
    syscall

    sw $v0 , 0($t2)

    addi $t0 , $t0 , 1
    addi $t2 , $t2 , 4

    j loop1

lab1: li $t0 , 0
     la $t2 , tab

loop2 : bge $t0 , $t1 , lab2

    lw $t4 , 0($t2)
    add $t3 , $t3 , $t4

    addi $t0 , $t0 , 1
    addi $t2 , $t2 , 4

    j loop2

lab2 : li $v0 , 4
     la $a0 , msg2
     syscall

    li $v0 , 1
    move $a0 , $t3
    syscall
```

```
int main()
{
    int tab[5];

    int i=0;

    while(i<5)
    {
        printf("Donnez S.V.P. une valeur :");
        scanf("%d",&tab[i]);
        i++;
    }

    int somme =0;

    i=0;

    while(i<5)
    {
        somme += tab[i];
        i++;
    }

    printf("La somme est : %d",somme);
    return 0;
}
```