COGNOMS:	GRUP:
NOM:	

Pregunta 5. (2,50 punts)

Donada la següent declaració de variables globals d'un programa escrit en llenguatge C:

```
char a[] = "DECA";
int b = 0;
int *c = &b;
short d[3] = {-14, 43, 5};
long long e[1000];
```

a) (0,50 p) Tradueix-la al llenguatge assemblador del MIPS

```
        .data
a:      .asciiz   "DECA"
b:      .word 0
c:      .word b
d:      .half -14, 43, 5
        .align 3
e:      .space 8000
```

b) (0,50 p) Completa la següent taula amb el contingut de memòria en hexadecimal de les primeres 24 posicions de memòria. Tingues en compte que el codi ascii de la 'A' és el 0x41. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Les posicions de memòria sense inicialitzar es deixen en blanc .

@Memòria	Dada
0x10010000	44
0x10010001	**45**
0x10010002	**43**
0x10010003	**41**
0x10010004	**00**
0x10010005	
0x10010006	
0x10010007	

@Memòria	Dada
0x10010008	**00**
0x10010009	**00**
0x1001000A	**00**
0x1001000B	**00**
0x1001000C	**08**
0x1001000D	**00**
0x1001000E	**01**
0x1001000F	**10**

@Memòria	Dada
0x10010010	**F2**
0x10010011	**FF**
0x10010012	**2B**
0x10010013	**00**
0x10010014	**05**
0x10010015	**00**
0x10010016	
0x10010017	

c) (0,50 p) Quin és el valor de $t0 en hexadecimal, després d'executar el següent fragment de codi?

```
li      $t1, 0x77777777
li      $t2, 0x55555555
la      $t3, d + 2
lh      $t0, 0($t3)
or      $t0, $t0, $t2
xor     $t0, $t0, $t1
```

$t0= **0x 22222208**

d) (0,50 p) Tradueix a llenguatge assemblador del MIPS la següent sentència en C:

```
*c = 18;
```

```
li      $t0, 18
la      $t1, c
lw      $t2, 0($t1)
sw      $t0, 0($t2)
```

e) (0,50 p) Tradueix a llenguatge assemblador del MIPS la següent sentència en C:

```
e[5] = 0;
```

```
la      $t0, e
sw      $zero, 40($t0)
sw      $zero, 44($t0)
```

Pregunta 6. (1 punt) Condicional

Donada la següent sentència escrita en alt nivell en C:

```
if (((x==0)&&(y!=0)) || ((y>x)&&(x<=0)))
        x=0;
else
        x=1;
```

Completa el següent fragment de codi MIPS, que tradueix l'anterior sentència, escrivint en cada calaix un mnemònic d'instrucció o macro, una etiqueta, o un registre. Les variables `x` i `y` són de tipus `int` i estan inicialitzades i guardades als registres `$s0` i `$s1`, respectivament.

```
        bne     $s0, $zero,  etiq1

        bne     $s1, $zero,  etiq2

etiq1:  ble     $s1, $s0,    etiq3

        bgt     $s0, $zero,  etiq3

etiq2:  move    $s0, $zero              # x=0
        b       etiq4

etiq3:  li      $s0,1                   # x=1

etiq4:
```