

# Tema 2. Enunciats dels problemes

Joan Manuel Parcerisa



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Facultat d'Informàtica de Barcelona



# Problema 2.10

2.10. Convertiu els següents números enters decimals als següents formats de representació binària per a 8 bits. Escriviu el resultat en hexadecimal:

	Ca1	Ca2	Signe i Magnitud	Excés a $2^7-1$
-78				
125				
0				
-1				

# Problema 2.19

**2.19.** Donades les següents declaracions:

```
        .data
u1:     .byte  '9'
u2:     .byte  '3'
d3:     .byte  0,  0
```

Cada dada *u1* i *u2* conté un caràcter ASCII que representa un dígit decimal. Escriu un programa en ensamblador que sumi els naturals representats per *u1* i *u2* deixant el resultat en *d3* (que té un dígit de més per poder representar qualsevol suma sense desbordar-se). Feu servir l'algorisme típic de la suma, dígit a dígit.

# Problema 2.20

2.20. Escriu en hexadecimal el contingut de la memòria ocupat per les següents declaracions, suposant que comencen a emmagatzemar-se a partir de l'adreça 0x10010000:

```
.data
a:   .half  54
b:   .byte  'C'
c:   .half  0xAAAA
d:   .word  0x4F3C
e:   .byte  -18
f:   .dword 0xFFFFEEEE
```

Fes servir el següent format, canviant els zeros pels valors corresponents en hexadecimal (recordeu que per defecte està activa l'alineació automàtica de les variables a memòria):

ADRECES	CONTINGUTS (hexadecimal, byte per byte)
0x10010000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10010010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

# Problema 2.23

2.23. Convertiu les següents instruccions de llenguatge ensamblador a llenguatge màquina, seguint el format de les instruccions MIPS:

Ll. Assemblador MIPS	Ll. Màquina MIPS (Bit a bit)	Ll. Màquina MIPS (Hexa)
<code>addu \$t4, \$t3, \$t5</code>		
<code>addiu \$t7, \$t6, 25</code>		
<code>lw \$t3, 0(\$t2)</code>		
<code>sw \$t0, 0(\$t1)</code>		

# Problema 2.30

2.30. Donades les següents declaracions:

```
char a;
int b;
long long int c;
main()
{
    char *p;           /* punter guardat en $t0 */
    int *q;            /* punter guardat en $t1 */
    long long int *h;  /* punter guardat en $t2 */
    ...
}
```

Suposant que els punters  $p$ ,  $q$  i  $h$  ocupen els registres  $\$t0$ ,  $\$t1$  i  $\$t2$ , tradueix a assemblador MIPS les següents sentències en C, pertanyents a la funció *main*:

- a) `q = q + 1;`
- b) `a = *p;`
- c) `h = &c;`
- d) `b = *(q + b);`
- e) `*h = *(h + b);`
- f) `p[*q + 10] = a;`
- g) `h = &h[*p];`

# Problema 2.32

2.32. Donades les següents declaracions:

```
char indx[100];           /* enters de 1 byte */
short meitat[100];       /* enters de 2 bytes */
int val[100], vec[100];  /* enters de 4 bytes */
main()
{
    char c;               // $t2
    int i, j;             // $t0, $t1
    ...
}
```

Tradueix les següents sentències en C suposant que les variables  $i$ ,  $j$ , i  $c$  ocupen els registres \$t0, \$t1, i \$t2:

- a) `i = val[5] + vec[10];`
- b) `i = vec[10 + val[5]];`
- c) `c = indx[i + val[j]];`
- d) `c = indx[meitat[i]];`
- e) `i = meitat[vec[i] + val[j]];`
- f) `i = val[meitat[i]] + vec[j];`
- g) `i = vec[indx[i]] - val[j];`
- h) `i = meitat[indx[val[j]]];`