

Pregunta 2. (1,8 punts)

Donades les següents declaracions en C:

```
unsigned int func2(int x, char M[][4], int *y);
unsigned int *pglob;
int func1(int par1, char par2[][4], int par3, int *par4) {
    int loc1;
    loc1 = par1 + par3;
    *pglob = func2 (*par4, &par2[3][0], par4);
    return *par4 - loc1;
}
```

A continuació es mostra una traducció de la funció `func1` a llenguatge MIPS que està incompleta. Llegiu-la amb atenció, a fi de contestar correctament a les preguntes de més avall.

`func1:`

```
addiu    $sp, $sp, -12
sw       $ra, 8($sp)
sw       $s0, 4($sp)
sw       $s1, 0($sp)
move     $s1, $a3                                # copia del punter par4

# loc1 = par1 + par3
addu     $s0, $a0, $a2

# *pglob = func2 (*par4, &par2[3][0], par4);
```

```
# CAIXA 1
# (pas de paràmetres a func2)
```

```
jal func2
```

```
# CAIXA 2
# (emmagatzema resultat de func2)
```

```
# return *par4 - loc1;
```

```
# CAIXA 3
# (sentència return)
```

```
lw       $ra, 8($sp)
lw       $s0, 4($sp)
lw       $s1, 0($sp)
addiu    $sp, $sp, 12
jr       $ra
```

- a) Completa la CAIXA 1, en llenguatge ensamblador de MIPS, amb el pas de paràmetres per a la crida a la funció `func2` corresponent a la següent sentència del cos de la subrutina `func1`.

```
... = func2 (*par4, &par2[3][0], par4);
```

```
lw      $a0, 0($a3)
addiu   $a1, $a1, 12
move    $a2, $a3
```

```
jal     func2
```

COGNOMS:

GRUP:

NOM:

- b) Completa la CAIXA 2, en llenguatge ensamblador de MIPS, amb la recollida del resultat de la crida i el seu emmagatzematge a l'adreça apuntada per la variable `pglob`, corresponent al següent fragment de sentència del cos de la subrutina `func1`: `*pglob = func2(...`

```
jal    func2
la     $t0, pglob
lw     $t1, 0($t0)
sw     $v0, 0($t1)
```

- c) Completa la CAIXA 3, en llenguatge ensamblador de MIPS, amb la traducció de la darrera sentència: `return *par4 - loc1;`

```
lw     $v0, 0($s1)
subu   $v0, $v0, $s0
```

Pregunta 3. (0,8 punts)

Donada la següent sentència escrita en alt nivell en C:

```
if ((a%2==0) || (b!=0)) && ((b<=a) || (a>0))
    z=a;
else
    z=b;
```

Completa el següent fragment de codi MIPS, que tradueix l'anterior sentència, escrivint en cada calaix un mnemònic d'instrucció o macro, etiqueta, registre o immediat. Les variables `a`, `b` i `z` són de tipus `int` i estan inicialitzades i guardades als registres `$t0`, `$t1` i `$t2`, respectivament.

```
andi    $t3, $t0, 1
etq1:    beq    $t3, $zero, etq3
etq2:    beq    $t1, $zero, etq6
etq3:    ble    $t1, $t0, etq5
etq4:    ble    $t0, $zero, etq6
etq5:    move   $t2, $t0
        b       etq7
etq6:    move   $t2, $t1
etq7:
```