

COGNOMS, NOM:

EXAMEN PARCIAL D'EC

4 de novembre de 2020

L'examen consta de 5 preguntes, que s'han de contestar als mateixos fulls de l'enunciat. No oblideu posar el nom i cognoms a tots els fulls. La durada de l'examen és de 90 minuts. Les notes, la solució i el procediment de revisió es publicaran al Racó.

Problema 1. (2,5 punts)

Donades les següents declaracions en llenguatge C, traduïu a ensamblador MIPS la funció f1 com una subrutina.

```
int f2(char a, long long *b, char c, int d);  
  
int f1(long long w[], int i, char *p, char c) {  
    return f2(*(p+i), &w[i], c, i) + 4*i;  
}
```

COGNOMS, NOM:

Problema 2. (1,5 punts)

Donada la següent sentència escrita en alt nivell en llenguatge C:

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

Completeu 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.

et1:	<input type="text"/>	\$s0, \$zero,	<input type="text"/>
et2:	<input type="text"/>	\$s1, \$zero,	<input type="text"/>
et3:	<input type="text"/>	\$s1, \$s0,	<input type="text"/>
et4:	<input type="text"/>	\$s0, \$zero,	<input type="text"/>
et5:	li	\$s1, 0	# y = 0
et6:	<input type="text"/>	<input type="text"/>	
et7:	li	\$s1, 1	# y = 1
et8:			

COGNOMS, NOM:

Problema 3. (2,5 punts)

Donat el següent programa escrit en llenguatge C, tradueix-lo a ensamblador MIPS amb el menor nombre possible de línies de codi a cada iteració del bucle.

```
short m[100][100];
main() {
int i = 0;
    while (m[i][i] > 0) {
        if (m[i][99-i] > m[i][i]) {
            m[i][99-i] = m[i][i];
        }
        i++;
    }
}
```

COGNOMS, NOM:

Problema 4. (1,5 punts)

- a) Mostreu el contingut de memòria a nivell de byte (amb les posicions d'alineament en blanc) corresponent a aquesta declaració:

```
.data                                # comença a 0x10010000
lin1:    .half      -3, 0xff, 1
lin2:    .word      lin4
lin3:    .byte      'x'                # el codi ascii de la 'x' és 120
lin4:    .asciiz    "xyz"
lin5:    .align     2
lin6:    .space     2
lin7:    .half      -0x7fff
```

0x10010000		0x10010008		0x10010010		0x10010018	
0x10010001		0x10010009		0x10010011		0x10010019	
0x10010002		0x1001000A		0x10010012		0x1001001A	
0x10010003		0x1001000B		0x10010013		0x1001001B	
0x10010004		0x1001000C		0x10010014		0x1001001C	
0x10010005		0x1001000D		0x10010015		0x1001001D	
0x10010006		0x1001000E		0x10010016		0x1001001E	
0x10010007		0x1001000F		0x10010017		0x1001001F	

- b) Indiqueu el valor final a \$t0 després d'executar aquest codi que fa referència a l'anterior declaració:

```
la    $t0, lin2
lw    $t0, 0($t0)
lhu   $t0, 1($t0)
lui   $t1, 0xA5A5
or    $t0, $t0, $t1
sra   $t0, $t0, 8
andi  $t0, $t0, -1
```

\$t0 =

COGNOMS, NOM:

Problema 5. (2 punts)

Feu un programa que detecti si el registre `$t1` conté un valor capicua a nivel de bit, és a dir que el seu contingut es pot expressar com `$t1: abcd efgh ijkl mnop ponm lkji hgfe dcba`, per a qualsevol valor binari de les variables des de `l` a fins `p`. El valor final a `$t1` s'ha de codificar com `CERT=1` i `FALS=0`.