

EXAMEN PARCIAL D'EC

20 d'abril de 2021

- L'examen consta de 7 preguntes, que s'han de contestar als mateixos fulls de l'enunciat. No oblidis posar el teu nom i cognoms a tots els fulls.
- La durada de l'examen és de 1:30 hores (90 minuts)
- Les notes, la solució i el procediment de revisió es publicaran al Racó el dia 3 de maig.

Pregunta 1 (1 punt)

Donada la següent subrutina en ensamblador MIPS:

```

acces_aleatori:
    li    $t0, 400
    mult  $t0, $a1
    mflo  $t0
    addu  $t0, $t0, $a0
    lw    $v0, -80($t0)
    jr    $ra

```

Sabem que és la traducció de la següent funció en C (de la qual desconexim els valors dels requadres). Completa els requadres amb les expressions vàlides en C per tal que la traducció sigui correcta:

```

int acces_aleatori (int M[][100], int i)
{
    return M[ ][ ];
}

```

Pregunta 2 (1.25 punts)

Donada la següent funció `foo` en alt nivell correcte (no cal comprovar si se surt de rang), on `M` i `V` són declarades com a variables globals:

```

int M[100][100]; /* suposarem que està inicialitzada */
int V[100];
void foo() {
    int i, aux; /* ocupen els registres $t1 i $t3 respectivament */
    for (i=0; i<97; i++) {
        aux = M[i][i+3];
        M[i+1][i+3] = V[aux];
    }
}

```

Completa el següent codi MIPS omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en alt nivell, tenint en compte que els elements de la matriu s'accedeixen utilitzant la tècnica **d'accés seqüencial**, usant el registre `$t0` com a punter per accedir els elements de la matriu.

```

    la    $t0, M + 
    li    $t1, 0
    li    $t2, 97
    la    $t7, V
bucle: bge $t1, $t2, fibuc
    lw    $t3, ( $t0)
    sll   $t3, $t3, 2
    addu  $t3, $t3, $t7
    lw    $t4, ( $t3)
    sw    $t4, ( $t0)
    addiu $t0, $t0, 
    addiu $t1, $t1, 1
b bucle
fibuc:

```

Donades les següents declaracions de variables locals

i el següent codi en alt nivell

Sabent que les variables a, b, c, d estan guardades en els registres \$t0, \$t1, \$t2, \$t3 respectivament, completa el següent fragment de codi MIPS omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en alt nivell (*nota: posa atenció al tipus de les variables!*):

Donat el següent fragment de codi en ensamblador MIPS:

a) Quin valor representa, en decimal en notació científica normalitzada, el contingut que hi ha a $\$12$? (0.5 punts)

[illegible]

	G	R	S
1,0000000000000000000000000010	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
- 0,000100000000000000000000000000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
xx,xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
$f8 = 0x
```

Pregunta 5 (1.5 punts)

Donada la següent declaració de variables globals, que s'ubica a memòria a partir de l'adreça 0x10010000:

```
.data
v1: .byte    -5
v2: .half    0x80
v3: .dword   0xFEDCBA9876543210
v4: .asciiz  "EC" # codi ASCII 'A' = 0x41
      .align  2
v5: .space   8
v6: .word    v1
```

- a) Omple la següent taula amb el contingut de memòria **en hexadecimal**. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Les posicions de memòria sense inicialitzar es deixen en blanc.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
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) Calcula el valor final del registre \$t0 en hexadecimal després d'executar el següent codi.

```
la    $t1, v1
lb    $t1, 0($t1)
sra   $t2, $t1, 4
sltu  $t0, $t1, $t2
```

\$t0 =

Pregunta 6 (1 punt)

S'executa un programa de test en un ordinador que té 3 tipus d'instruccions (A,B,C), amb CPI diferents. La següent taula especifica el nombre d'instruccions de cada tipus que executa el programa i el seu CPI:

Tipus d'instrucció	Nombre d'instruccions	CPI
A	$2 \cdot 10^9$	1
B	$1 \cdot 10^9$	1
C	$1 \cdot 10^9$	2

- a) Sabem que la freqüència de rellotge és de 1GHz i que la potència dissipada és $P=100W$. Calcula el temps d'execució en segons i l'energia consumida en Joules durant l'execució del programa de test.

$t_{exe} =$

$E =$

- b) S'implementen unes millores al CPU que permeten incrementar la freqüència de rellotge a 2GHz. No obstant, el CPI de les instruccions de tipus B passa a ser de 4 cicles. Quin és el guany (speedup) de rendiment obtingut?

Guany =

Pregunta 7 (2.5 punts)

Donat el següent programa en C:

```
char G[7] = "EXAMEN";

int main() {
    char L[7];
    int i = 0;
    char x = 1;
    while(G[i] != '\0') {
        x = func(&G[i], L, i, x);
        i++;
    }
}

char func(char *a, char P[7], int b, char c) {
    P[b] = *a + c;
    return c + 1;
}
```

Completa la següent traducció a llenguatge ensamblador MIPS omplint les caselles en blanc:

```
.data
G: .asciiz "EXAMEN"
.text
main:
    addiu $sp, $sp, 
    sw    $s0, ($sp)
    
    li    , 0
    li    $v0, 1
while:
    la    $a0, G
    addu   $a0, $a0, 
    lb     $t0, 0($a0)
    beq    $t0, $zero, endwhile
    
    
    move   $a3, $v0
    jal    func
    addiu  , , 1
    b      while
endwhile:
    lw     $s0, ($sp)
    
    addiu  $sp, $sp, 
    jr     $ra

func:
    
    addu   $t0, $t0, $a3
    addu   $t1, , 
    sb     $t0, 0($t1)
    addiu  , $a3, 1
    jr     $ra
```