

EC Reavaluació. Sessió 2: Vectors, Matrius, IFs i Bucles

Exercici 1 (Ex. parcial 2016-2017 Q2)

Donada la següent sentència escrita en alt nivell en C:

```
if ((a>b || b<0) && a>0)
    a=0;
else
    a=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 a i b són de tipus int i estan inicialitzades i guardades als registres *t0* i *t1*, respectivament.

```

          ----- $t0, $t1, -----
eti1: ----- $t1, $zero, -----
eti2: ----- $t0, $zero, -----
eti3: move     $t0, $zero
eti4: b        -----
eti5: li       $t0, 1
eti6:
```

Exercici 2 (Ex. parcial 2017-2018 Q1)

Considera la següent subrutina programada en assembleador MIPS:

```
func:
    ble $a0, $a2, eti1
    ble $a1, $a2, eti2
eti1: beq $a2, $zero, eti3
eti2: li  $v0, 0
      b eti4
eti3: li $v0, 1
eti4:
      jr $ra
```

Completa el següent codi escrit en C omplint les caselles en blanc perquè sigui equivalent a l'anterior codi en assembleador:

```
int func(int x, int y, int z)
{
    int res;
    if ( (( _____ ) && ( _____ )) || ( _____ ) ) {
        res = 0;
    } else
        res = 1;
    }
    return res;
}
```

Exercici 3 (Ex. Final 2017-2018 Q1)

Donada la següent declaració en C:

```
short acces_aleatori(short M[][100], int i) {
    return M[i+2][i-1];
}
```

Completa els requadres del següent fragment de codi per tal que sigui la traducció correcta de la funció anterior:

```
acces_aleatori:
    li    $t0, _____
    mult  $t0, _____
    mflo  $t0
    addu  $t0, $t0, _____
    lh    $v0, _____($t0)
    jr    $ra
```

Exercici 4 (Ex. Parcial 2018-2019 Q1)

Donada la següent declaració en C:

```
void func(int mat[][40], int i) {
    mat[i][20-i] = mat[20-i][i];
}
```

Completa els requadres del següent fragment de codi per tal que sigui la traducció optimitzada de la funció func (pista: determina les dues adreces de memòria en funció de i, i observa si contenen algun terme en comú):

```
func:
    li    $t2, _____
    mult  $t2, $a1
    mflo  $t2
    subu  $t1, $a0, $t2
    addu  $t0, $a0, $t2
    lw    $t3, _____($t1)
    sw    $t3, _____($t0)
    jr    $ra
```

Exercici 5 (Ex. Final 2011-2012 Q2)

Considera el següent programa

```
int v[20], m[20][20];

main() {
    int i;

    for (i=19; i>=0; i--)
        v[i] = m[19-i][i];
}
```

Tradueix el programa principal a llenguatge ensamblador MIPS. Només superaran aquesta pregunta aquelles solucions en què cada iteració del bucle tingui 7 o menys línies de codi.

Exercici 6 (Ex. Parcial 2016-2017 Q2)

Considera el següent programa

```
short M[100][100];
void foo() {
    int i; /* ocupa el registre $t0 */
    for (i=50; i>=2; i=i-2){
        M[i+1][100-i] = M[i-1][100-i];
    }
}
```

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 M s'accedeixen utilitzant la tècnica d'accés seqüencial, usant el registre \$t1 com a punter, i que aquest punter s'inicialitza amb l'adreça de l'element M[49][50]:

```
foo:
    la $t1, M + _____ # @M[49][50]
    li $t0, 50 # i = 50
    li $t2, 2
for:
    blt $t0, $t2, endfor
    lh $t4, _____($t1) # M[i-1][100-i]
    sh $t4, _____($t1) # M[i+1][100-i]=...
    addiu $t1, $t1, _____
    addiu $t0, $t0, _____ # i=i-2
    b for
endfor:
    jr $ra
```