

EC Reavaluació. Sessió 3: Subrutines

Exercici 1 (Ex. Final 2012-2013 Q1)

Tradueix a llenguatge ensamblador MIPS la subrutina func1:

```
int func2(int X, int Y);

int func1(int a, int *b, int c) {
    *b = func2(a, c);
    return (*b+c);
}
```

Exercici 2 (Ex. Parcial 2018-2019 Q1)

Tradueix la funció f2 de llenguatge C a una subrutina MIPS:

```
void f1(int v[], int x);

int f2(int a, int b)
{
    int tmp[2];

    f1(tmp, a);
    return tmp[0] + tmp[1] + b;
}
```

Exercici 3 (Ex. Parcial 2013-2014 Q1)

Tradueix a llenguatge ensamblador MIPS la subrutina func1:

```
short func2(short a, char *b, short *c);

short func1 (int x, short *y) {
    char V1[7];
    short V2[7];
    short res;
    res=func2(*y,&V1[x],V2);
    if (x>0)
        res++;
    return res+(*y);
}
```

Exercici 4 (Ex. Parcial 2012-2013 Q1)

Tradueix a llenguatge ensamblador MIPS la subrutina func1:

```
int func2(int X[], int Y);
int func3(int *d);

int func1(int a, int *p) {
    int V[5];
    int b;
    func3(&b);
    if (a>b)
        { *p = func2(V,V[*p]); }
    return *(p+1);
}
```

Exercici 5 (Ex. Parcial 2018-2019 Q2)

Donada la següent funció en C, tradueix-la a llenguatge MIPS:

```
float inc_f(float *x) {  
    return *x+1.0;  
}
```

Exercici 6 (Ex. Final 2013-2014 Q1)

Es demana que facis les correccions necessàries per tal que els codis generats siguin correctes. Escribeu TOT el codi correcte de cada subrutina a l'espai indicat a la dreta del codi erroni.

a) void subr2(char *p1, int p2) {
 char vl[4] = "123"; // codi ASCII '0' = 0x30
 *p1 = vl[p2%4];
}

Traducció ERRÒNIA:

```
subr2:  
    addiu    $sp, $sp, -4  
    li      $t0, 0x00333231  
    sw      $t0, 0($sp)  
    sra     $t1, $a1, 2  
    addu    $t1, $sp, $t1  
    lw      $t2, 0($t1)  
    lw      $t3, 0($a0)  
    sw      $t2, 0($t3)  
    addiu    $sp, $sp, 4  
    jr      $ra
```

b) short subr3(short p0, short* p1) {
 short res;

 res = *p1;
 if (p0>1) {
 res += subr3(p0-1, p1+1);
 }
 return res;
}

Traducció ERRÒNIA:

```
subr3:  
    addiu    $sp, $sp, -4  
    sw      $ra, 0($sp)  
    move     $t0, $a1  
    li      $t1, 1  
    ble     $a0, $t1, fi3  
    addiu    $a0, $a0, -1  
    addiu    $a1, $a1, 2  
    jal     subr3  
    move     $v0, $t0  
fi3:  
    lw      $ra, 0($sp)  
    addiu    $sp, $sp, 4  
    jr      $ra
```