SOLUZIONI ESERCITAZIONE DEL 15/04/2004

- Codifica in linguaggio assembly delle seguenti funzioni:

  1) a= b * sum(c,d,e) con sum(c,d,e) = c + max(d,e) e con max(d,e) = d sse d > e, e viceversa.

  Definizione registri secondo la convenzione:
  a=$s1 b=$s2 c=$s3 d=$s4 e=$s5

  Codice assembly corrispondente. Definisco le procedure:

```
# in $a0 troviamo d ed in $a1 troviamo e
max:        bgtz $a0 $a1 maggiore
            move $v0, $a1
            j fine_max
maggiore:   move $v0, $a0
fine_max:   jr $ra

# in $a0 troviamo c, in $a1 troviamo d ed in $a2 troviamo e
sum:            addi $sp, $sp, -8
            sw $ra, 0($sp)
            sw $a0, 4($sp)
            move $a0, $a1
            move $a1, $a2
            jal max
            lw $a0, 4($sp)
            add $v0, $v0, $a0
            lw $ra, 0($sp)
            addi $sp, $sp, 8
            jr $ra

main:
            .....
            move $a0, $s3
            move $a1, $s4
            move $a2, $s5
            jal sum
            mul $s0, $v0, $s1
            ......
```

  2) …

```
    do
        g = g + A[i];
        i = i + j;
     while (i != h)
     f = (x + y) – (z + q) ;
     a = g + f ;
```

  Definizione registri secondo la convenzione:
  g = $s0
  h = $s1

i = $s2
j = $s3
A[0] = $s4
x = $s5
y = $s6
z = $s7
q = $s8
registri temporanei:
$t0, $t1, $t2, $t3, $t4, $t5

Codice assembly corrispondente:

```
        add $s2, $zero, $zero
Loop:   muli $t1, $s2
        add $t1, $t1, $s4
        lw $t0, 0($t1)
        add $s0, $s0, $t0
        add $s2, $s2, $s3
        bne $s2, $s1, Loop
        add $t2, $s7, $s8
        add $t3, $s5, $s6
        sub $t4, $t3, $t2
        add $t5, $s0, $t4
```

- Traduzione in linguaggio macchina di istruzioni assembly:

L1: lw $t0, 8($t1) 8

Op	Rs	Rt	Indirizzo
100011	$t1	$t0	8
100011	01001	01000	000...1000

Beq $t0, $zero, L1 24

Op	Rs	Rt	Indirizzo
000100	8	0	-5
000100	01000	00000	111...1011

J L1 32

Op	Indirizzo
000010	8
00010	0000...1000

...
add $t1, $s1, $s2 48

Op	Rs	Rt	Rd	Shamt	Funct
000000	17	18	9	0	32
000000	10001	10010	01001	00000	100000

...

L2: lw $t2, 32($s3) 60

Op	Rs	Rt	Indirizzo
100011	$s3	$t2	32
100011	10011	01010	000...100000