iterativa

```
list_length: move   $t0,$a0
             move   $v0,$zero
loop:        beqz   $t0,end
             addi   $v0,1
             lw     $t0,($t0)
             b      loop
end:         jr     $ra
```

ricorsiva (ovviamente è più lunga)

```
list_length: beqz   $a0,zero
             addi   $sp,-8
             sw     $ra,($sp)
             sw     $a0,4($sp)

             lw     $a0,($a0)
             jal    length
             addi   $v0,1

             lw     $ra,($sp)
             lw     $a0,4($sp)
             addi   $sp,8
             jr     $ra
zero:        li     $v0,0
             jr     $ra


list_count:  move   $t0,$a2
             add    $t1,$a0,1
zerovect:    sw     $zero,($t0)
             addi   $t0,4
             addi   $t1,-1
             bnez   $t1,zerovect

             move   $t0,$a1
loop:        beqz   $t0,end
             lw     $t1,4($t0)
             sll    $t2,$t1,2
             add    $t2,$a2,$t2   t2 = a2+4*t1, (t2)=vect[t1]
             lw     $t3,($t2)
             addi   $t3,1
             sw     $t3,($t2)
             lw     $t0,($t0)
             b      loop
end:         jr     $ra
```

oppure (ottimizzata nei salti)

```
list_count:  move   $t0,$a2
             add    $t1,$a0,1
zerovect:    sw     $zero,($t0)
             addi   $t0,4
             addi   $t1,-1
             bnez   $t1,zerovect

             move   $t0,$a1
             beqz   $t0,end       check lista vuota!

loop:        lw     $t1,4($t0)
             sll    $t2,$t1,2
             add    $t2,$a2,$t2
             lw     $t3,($t2)
             addi   $t3,1
             sw     $t3,($t2)
             lw     $t0,($t0)
             bnez   $t0,loop
end:         jr     $ra
```