

Esercizio premiante Assembly

Matricola no. 900411 – Cattaneo Francesco

Codice Assembly:

```
.data
numeri: .word 9, 2, -11, 0, 99

.text
main:
    la $t0, numeri #inizializza $t0, puntatore ai numeri in .data
    li $t1, 0 #inizializza $t1, contatore

loop:
    lw $a0, 0($t0) #preleva il numero puntato
    beq $a0, $zero, esci #se si trova lo zero in .data, esegui il branch a
    esci
    addi $t1, $t1, 1 #aggiungi 1 al contatore
    addi $t0, $t0, 4 #procedi al prossimo numero in memoria
    j loop #se non si è trovato uno zero, esegui di nuovo il controllo

esci:
    move $v0, $t0 #$v0 punta alla prima parola contenente il valore 0
    move $t0, $t1
    li $v0, 10 #carica il valore 10 → exit
    syscall
```

Spiegazione:

Il programma scorre i numeri in memoria a partire da "numeri" fino a trovare il primo 0, contando quante cifre lo precedono. È flessibile e utilizza i registri \$t0 e \$t1 come puntatore e contatore, rispettivamente. Una volta che i registri sono stati inizializzati, il loop può partire prelevando il numero puntato, e salvandolo in a0. Una volta salvato il numero puntato, possiamo verificare se è diverso da zero: se lo è, allora si continua con l'esecuzione del loop, altrimenti il programma esegue un branch alla label "esci", che salva l'indirizzo di memoria dove è salvato lo zero nel registro \$v0. Continuando nel loop, se non trova lo zero nell'indirizzo di memoria selezionato, allora aggiunge 1 al contatore, aggiunge 4 al registro \$t0 (aggiungendo 4, possiamo passare alla prossima area di memoria, ed effettivamente passare al prossimo numero salvato in memoria), e riprende l'esecuzione da capo del loop, in modo da poter verificare di nuovo se nella memoria è presente uno zero. Una volta che lo zero è stato trovato in memoria, il programma memorizza il valore dell'indirizzo di memoria dove lo zero era presente, e successivamente esce dal programma, salvando il valore 10 nel registro \$v0, ed eseguendo una syscall.