

Laboratorio di Architetture degli elaboratori (n. 11, 10-12/02/04)

Programma 1

Scrivere un programma Assembly MIPS che verifica se un vettore di n numeri a 32 bit contiene almeno una occorrenza del numero 0. In caso positivo il programma deve assegnare al byte meno significativo del registro V0 il valore 1, altrimenti -1. Mettere l'indirizzo base del vettore in T0 e la lunghezza n in T1. Si termini il codice con la seguente chiamata di sistema per visualizzare il risultato:

```
end:      move    $a0,$v0
          li      $v0,1
          syscall          #chiamata di sistema
```

Programma 2

Scrivere un programma Assembly che converte una stringa di caratteri numerici nel corrispondente valore intero. Si assuma che nella stringa ci siano solo caratteri da '0' a '9' (oltre lo 0 finale). Si ricordi che nella codifica ASCII i caratteri suddetti sono contigui quindi per convertire una cifra nel suo valore numerico basta sottrarvi il valore '0'. Mettere l'indirizzo base della stringa in T0. Si dichiari la stringa in memoria con la direttiva `.asciiz`:

```
string:  .asciiz "1423" #una stringa di prova
```

Programma 3

Scrivere un programma che verifica se un vettore di n numeri a 8 bit contiene solamente numeri compresi fra 0 e 16. In caso positivo il programma deve assegnare al byte meno significativo del registro V0 il valore 255 *unsigned*, altrimenti il valore 0. (Mettere l'indirizzo base del vettore in T0 e la lunghezza n in T1)

Si consiglia di strutturare i programmi secondo lo schema usuale:

```
        .text
        CODICE
        .data
Vett    .word  DATI
        .byte  DATI
```