

Architetture dei Sistemi a Elaborazione – a.a. 2014/15

Esercitazione di Laboratorio 1

1. Si scriva un programma in linguaggio Assembler 8086 che inizializzi a FFh tutti i registri general purpose a disposizione.
2. Si scriva un programma in linguaggio Assembler 8086 che esegua le seguenti operazioni:
 - a. Definisca una matrice 5*5 chiamata MATR i cui elementi siano memorizzati su 8 bit, inizializzati a piacimento.
 - b. Senza perdere precisione, si memorizzi in una locazione di memoria scelta la somma di tutti gli elementi della riga 2
 - c. Senza perdere precisione, si memorizzi in una locazione di memoria scelta la somma di tutti gli elementi della colonna 3
 - d. Esegua la differenza dei due risultati ottenuti al punto b. e c. e la si memorizzi in una locazione di memoria scelta.
3. Si scriva un programma in linguaggio Assembler 8086 che esegua le seguenti operazioni:
 - a. Definisca due variabili in memoria, VARW su 16 bit e VARD su 32 bit, inizializzati rispettivamente a FFFFH e 00112233H
 - b. Scambi i due byte più significativi di VARD con VARW
 - c. Esegua la somma di VARW e VARD (valori dopo lo scambio) memorizzandola in VARD

n.b. si utilizzino il metodo di indirizzamento base relative addressing e l'istruzione LEA
4. Si scriva un programma in linguaggio Assembler 8086 che esegua le seguenti operazioni:
 - a. Definisca un vettore VETT di 8 elementi ciascuno su 32 bit, inizializzato a piacimento a numeri positivi e negativi, senza utilizzare la direttiva DUP
 - b. Esegua la somma di tutti numeri positivi contenuti in VETT e depositi il risultato in una variabile VARP su 32 bit
 - c. Esegua la somma di tutti numeri negativi contenuti in VETT e depositi il risultato in una variabile VARN su 32 bit
 - d. *In entrambi i punti b. e c., si tenga conto dell'overflow: in caso di overflow la variabile risultato deve essere posta al valore 0.*

Si esegua il codice scritto variando i valori di VETT più volte, cercando di osservare (tramite i flag) le possibili situazioni di carry e overflow. Vedi esempi:

```
VETT DD 2,4,-3,6,5,800000FFH,0FFFFFFFH,-7
; non da overflow

VETT DD 2,4,-3,6,5,80000000H,0FFFFFFFH,-7
; da overflow sui numeri negativi

VETT DD 2,4,-3,6,7FFFFFFFH,800000FFH,0FFFFFFFH,-7
; da overflow sui numeri positivi

VETT DD 2,4,-3,6,7FFFFFFFH,80000000H,0FFFFFFFH,-7
; da overflow su entrambi
```