

# Esercizi Assembly 1

M. Sonza Reorda – M. Grosso

Politecnico di Torino  
Dipartimento di Automatica e Informatica

## Esercizio 1

- Siano date tre variabili di tipo byte corrispondenti a tre caratteri alfabetici minuscoli (ASCII)
  - Var1 = 'a'
  - Var2 = 's'
  - Var3 = 'm'
- Si scriva un programma che stampi a video i tre caratteri convertiti in maiuscolo.

## Esercizio 2

- Siano date le seguenti variabili di tipo byte già inizializzate in memoria:
  - `n1 db 10`
  - `n2 db 10h`
  - `n3 db 10b`
- Si calcoli la seguente espressione, il cui risultato dovrà essere salvato nella variabile *byte res*, e si verifichi il risultato:
  - $n1 + n2 - n3$ .

## Esercizio 3

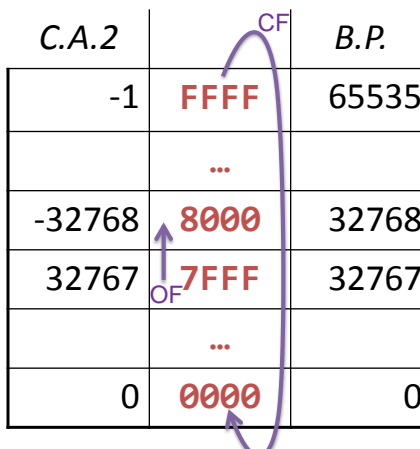
- Siano date le seguenti variabili di tipo word (*con segno*) già inizializzate in memoria:
  - `OPA = -459`
  - `OPB = 470`
  - `OPC = -32756`
  - `OPD = 1`
- Si scriva un programma per l'esecuzione dell'espressione  $OPA + OPB - OPC + OPD$  utilizzando il registro AX
- Si osservino in modalità passo-passo il risultato parziale e il comportamento delle flag (sign, overflow e carry), spiegando quanto visto.

## Esercizio 4

- Siano date le seguenti variabili di tipo word (*unsigned*) già inizializzate in memoria:
  - OPA = 32767
  - OPB = 1
- Si scriva un programma per l'esecuzione dell'espressione OPA+OPB+OPA+OPB utilizzando il registro AX
- Si osservino in modalità passo-passo il risultato parziale e il comportamento delle flag (sign, overflow e carry), spiegando quanto visto.

### 8086: Rappresentazione dei numeri

- Il processore 8086 permette di svolgere operazioni su numeri in *complemento a 2* (con segno) o in *binario puro*
- Analizziamo l'istruzione add:
  - Carry Flag è settata nel passare tra FFFF e 0000
  - Overflow Flag è settata nel passare tra 7FFF e 8000



C.A.2		B.P.
-1	FFFF	65535
	...	
-32768	8000	32768
32767	7FFF	32767
	...	
0	0000	0

- Per valutare condizioni di overflow occorre prestare attenzione al comportamento delle flag tenendo conto del tipo di rappresentazione che si intende utilizzare (il comportamento del processore non cambia).

## Esercizio 5

- Siano date tre variabili di tipo *byte* in memoria, che rappresentino rispettivamente il numero di giorni, ore e minuti passati da un certo istante  $T_0$ . Si calcoli il numero totale di minuti passati da  $T_0$ , e tale valore sia salvato nella variabile di tipo *word* **risultato**.
  - Per estendere l'intervallo di numeri rappresentabili, si richiede di lavorare con una rappresentazione in binario puro
  - In caso di *overflow* della rappresentazione scrivere in **risultato** il valore FFFFh.

## Implementazione

- È necessario eseguire una somma pesata delle variabili di ingresso:
  - Conversione dei giorni in ore
  - Somma delle ore
  - Conversione delle ore in minuti
  - Somma dei minuti
- Operazioni intermedie eseguite su *word*
- Valutare quando è possibile ottenere *overflow* nelle operazioni di somma/moltiplicazione!

## Esercizio 6

- Si scriva un programma che calcoli la media (intera) tra i DIM valori di un vettore di *byte*, e ne salvi il risultato nella variabile `risultato`
- Verificare l'arrotondamento effettuato.