

# Esercizi Assembly 2

M. Sonza Reorda – M. Grosso – M. Monetti

Politecnico di Torino  
Dipartimento di Automatica e Informatica

## Esercizio 1

- 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.

## Esercizio 2

- Si scriva un programma che stampi a video il valore decimale di un intero nell'intervallo  $[0, 2^{16}-1]$  memorizzato in un'opportuna variabile.

## Implementazione

- Si utilizza un algoritmo in due passi:
  - Scomposizione del numero nelle sue cifre tramite divisioni successive per 10, salvando i resti e ripetendo l'operazione sul quoziente sino a che questo è diverso da zero
  - Visualizzazione delle cifre così ottenute in ordine inverso a quello di generazione, utilizzando lo stack
  - N.B.: le cifre devono essere convertite in caratteri ASCII prima della stampa.

## Esercizio 3

- Si scriva un programma che richieda all'utente un intero positivo (eventualmente composto da più cifre, e concluso con ENTER) e lo salvi in una variabile di tipo *word*. L'inserimento di valori troppo grandi deve segnalare un errore.
  - Approfondimento: Acquisire 5 interi positivi separati da ENTER e memorizzarli in un vettore di *word*.

## Implementazione

- Si utilizza un algoritmo in due passi:
  - nel primo si acquisiscono i caratteri ASCII;
  - nel secondo passo si convertono in intero, valutando la presenza eventuale di overflow
- I due passi possono essere svolti nello stesso ciclo.

## Esercizio 4

- 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 (*unsigned*)
  - 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 5

- Si scriva un programma in linguaggio Assembly 8086 che scriva in un vettore definito di 20 elementi di tipo *word* i primi 20 valori della serie di Fibonacci.
- Serie di Fibonacci
  - $\text{vet}[i] = \text{vet}[i-1] + \text{vet}[i-2] \Rightarrow \text{vet} = 1, 1, 2, 3, 5, 8, \dots$