

Alcuni (simpatici) esercizi di Assembly

1. Sommatore BCD con ripple carry

Scrivere un programma assembly che simuli un sommatore a 4 cifre in base 10. Il programma dovrà comportarsi come segue:

1. Preleva due numeri naturali, X e Y, su 4 cifre in base 10.
2. Esegue la somma dei due numeri come se ci fosse un montaggio ripple carry, come visto a lezione, di sommatore ad una cifra in base 10.
3. Mostra i risultati in forma tabulare, come nell'esempio di cui sotto.
4. Mostra il risultato finale.

Successivamente, si vuole parametrizzare il programma per mezzo della direttiva .EQU in modo che funzioni per una base generica BETA compresa tra 2 e 10. Si faccia attenzione che l'input sia sempre consistente

```
X? 5639
Y? 1903

i   x   y   cin |   s   cout
-----|-----
0   9   3   0   |   2   1
1   3   0   1   |   4   0
2   6   9   0   |   5   1
3   5   1   1   |   7   0

X+Y = 7542   cout = 0
```

2. Conversione da base 7 in CR a base 10 in CR

Scrivere un programma assembly che prelevi un numero in complemento alla radice in base 7 e lo converta nel corrispondente numero in base 10 su due cifre, anch'esso in complemento alla radice. Poiché l'operazione non è sempre fattibile, si realizzi anche un controllo di overflow, con il quale si esce dal programma. Ci si aspetta un output del seguente formato:

```
X? 100
b10_2: 49
X? 666
b10_2: 99
X? 565
b10_2: overflow
```

3. Conversione da base 16 in CR a base 10 in MS

Scrivere un programma assembly che prelevi un numero in base 16 considerato come la rappresentazione in complemento alla radice di un intero x. Si stampi x in modulo e segno in base 10. Nel caso in cui il carattere inserito sia 'Q', il programma

termina. Si svolga senza utilizzare l'istruzione MUL. Ci si aspetta un output del seguente formato:

X? A0

b10z: +32

X? 10

b10z: -112