

Laboratorio 6



R. Ferrero

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

This work is licensed under the Creative Commons (CC BY-SA) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Esercizio 1

- Scrivere una subroutine `myUADD8` che implementi la seguente istruzione:

`UADD8 <Rd>, <Rn>, <Rm>`

- Per il funzionamento di `UADD8`, si veda l'esercizio 1 del laboratorio 4.
- I parametri e il valore di ritorno sono passati attraverso registri: `Rd = r4`, `Rn = r0`, `Rm = r1`
- Ad eccezione di `r4`, il valore dei registri deve essere ripristinato al termine della subroutine.

Esercizio 2

- Scrivere una subroutine `myUSAD8` che implementi la seguente istruzione:

`USAD8 <Rd>, <Rn>, <Rm>`

- Per il funzionamento di `USAD8`, si veda l'esercizio 2 del laboratorio 4.
- I parametri e il valore di ritorno sono salvati in un'area di memoria `DATA READWRITE`, il cui indirizzo è passato a `myUSAD8` attraverso `r6`
- La subroutine non deve modificare i registri.

Esercizio 3

- Scrivere due subroutine `mySMUAD` e `mySMUSD` che implementino le seguenti istruzioni:

`SMUAD <Rd>, <Rn>, <Rm>`

`SMUSD <Rd>, <Rn>, <Rm>`

- Per il funzionamento di `SMUAD` e `SMUSD`, si veda l'esercizio 3 del laboratorio 4.
- I parametri e il valore di ritorno sono passati attraverso lo stack.
- Le subroutine non devono modificare i registri

Esercizio 4

Un quadrato magico perfetto è una matrice $N \times N$ che rispetta le seguenti tre condizioni:

1. gli elementi appartengono all'intervallo $[1, N^2]$
2. gli elementi sono tutti distinti fra loro
3. la somma degli elementi in ogni riga, in ogni colonna e in entrambe le diagonali è costante, pari alla costante di magia M :

$$M = \frac{1}{2} N(N^2 + 1)$$

Esercizio 4: esempi

$$\begin{bmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{bmatrix}$$
$$\begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 43 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{bmatrix}$$

Esercizio 4

- Scrivere una subroutine `magicSquare` per determinare se una sequenza di byte salvati in memoria formi un quadrato magico perfetto.
- La subroutine riceve attraverso lo stack l'indirizzo del primo byte e la dimensione N .
- La subroutine restituisce attraverso lo stack un valore booleano: 1 se la sequenza è un quadrato magico perfetto, 0 altrimenti.

Esercizio 4: suggerimento

- La seconda condizione è verificabile con un unico ciclo, utilizzando un vettore di appoggio con N^2 elementi di dimensione byte.
- Il vettore è allocato in un'area DATA READWRITE ed inizializzato a zero.
- Se l'elemento i -esimo del quadrato ha valore x , l'elemento x -esimo del vettore è posto a 1.
- Al termine del ciclo, tutti gli elementi del vettore devono essere uguali a 1.