

Esercizio 1

Considerare il codice riportato nel file “esercizio1.s” in “Lab 08 - esercizi.zip” ed utilizzare il debugger per:

- a) Indicare il contenuto di ciascuno dei registri w19, w22 e w24 **dopo l'esecuzione** delle istruzioni alle righe 16, 21, 28.
- b) Indicare tutte le istruzioni dopo le quali il valore contenuto in w19 passa da essere minore di quello in w24, ad esserne maggiore.

Suggerimento: eseguire il programma una istruzione alla volta tramite *ni* o *Step Over* (F10 su Visual Studio Code) ed individuare quali istruzioni fanno in modo che il valore di w19 passa da essere minore di w24, ad esserne maggiore.

- c) Indicare il valore esadecimale contenuto nei registri w19, w22 e w24 **dopo l'esecuzione** dell'istruzione a riga 27.

Esercizio 2

Eseguire il codice riportato nel file “esercizio2.s” in “Lab 08 - esercizi.zip” utilizzando il debugger per identificare gli errori e correggerli.

Il programma dovrebbe consentire di depositare o prelevare denaro da un conto corrente, inizialmente vuoto.

Esercizio 3

Eseguire il codice riportato nel file “esercizio3.s” in “Lab 08 - esercizi.zip” utilizzando il debugger per identificare gli errori e correggerli.

Il programma richiede in input la dimensione di un array ed i suoi elementi e dovrebbe verificare se l'array contiene una progressione aritmetica.

Una progressione aritmetica è una successione di numeri tali che la differenza tra ciascun elemento della successione e il suo precedente sia costante. Tale differenza viene detta ragione della progressione. Per esempio: 3, 5, 7, 9 è una progressione aritmetica con *ragione* 2.

Esercizio 4

Eseguire il codice riportato nel file “esercizio4.s” in “Lab 08 - esercizi.zip” utilizzando il debugger per identificare gli errori e correggerli.

Il programma richiede di inserire in input degli anni e dovrebbe verificare se sono bisestili.

Un anno è bisestile se è divisibile per 4 ma non per 100, oppure se è divisibile per 400.

Esercizio 5

Eseguire il codice riportato nel file “esercizio5.s” in “Lab 08 - esercizi.zip” utilizzando il debugger per identificare gli errori e correggerli.

Il programma richiede in input due interi positivi: base m ed esponente n e dovrebbe calcolare ricorsivamente m^n . Tale calcolo ricorsivo si basa sulla seguente osservazione:

$$m^n = (m^{n-1}) * m$$

Ad esempio, se n fosse 3 ed m fosse 2, allora $2^3 = 2^2 * 2 = 2^1 * 2 * 2 = 2 * 2 * 2$.