Esercizio 1

a) Partendo dal programma riportato nel file "esercizio1_a.s" in "Lab08Esercizi.zip", realizzare un programma che utilizzando un ciclo, legga da input una sequenza di interi positivi terminata da un numero negativo e conti quanti interi pari sono presenti nella sequenza. Infine, il programma stampa tale conteggio.
b) Partendo dal programma riportato nel file "esercizio1_b.s" in "Lab08Esercizi.zip", generalizzare il programma realizzato per il punto a) e scrivere un programma che letto da input un intero X e una sequenza di interi positivi terminata da un numero negativo, calcoli e stampi il numero di interi nella sequenza che sono divisibili per X.

Suggerimento: si tratta di varianti degli esempi di programmi visti nell'ultima lezione. Per la parte b) si può prendere spunto dal programma "divisibili.s".

Esercizio 2

Completare il programma "esercizio2.s" riportato in "Lab08Esercizi.zip" implementando la funzione ***diff*** che ricevuti due interi A e B (tramite i registri X0 e X1) restituisca (impostando opportunamente il registro X0):

- A-B, se A>B,
- B-A, altrimenti.

Esercizio 3

Partendo dallo scheletro di programma riportato in "esercizio3.s" in "Lab08Esercizi.zip", realizzare un programma che legga da input tre numeri A, B, C, maggiori di 1, e stampi **SI** se C è un divisore sia di A e che di B (ovvero A%C == 0 e B%C == 0), **NO** altrimenti. Per verificare se due numeri sono divisibili, si implementi una funzione ***divisibili*** che ricevuti due interi X e Y (tramite i registri W0 e W1) restituisca (impostando opportunamente il registro W0):

- 1 se Y è divisore di X, ovvero X%Y == 0,
- 0 altrimenti.

Suggerimento: Anche per questo esercizio si può prendere spunto dal programma "divisibili.s".

Esercizio 4

Partendo dallo scheletro di programma riportato in "esercizio4.s" in "Lab08Esercizi.zip", realizzare un programma che letto da input un numero N > 1, stampi tutti i numeri primi minori o uguali a N. A tal proposito, si implementi una funzione ***primo*** che ricevuto un numero restituisca 1 se tale numero è primo, 0 altrimenti.

Suggerimento: per verificare se un numero X > 1 è primo è sufficiente controllare se esiste almeno un numero compreso tra 2 ed X-1 divisibile per X. Ricordiamo che il numero 1 non è primo.

Esempi.

- Se N fosse: 3: il programma dovrebbe stampare 2 e 3.
- Se N fosse: 30: il programma dovrebbe stampare 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

Esercizio 5

La congettura di Collatz, chiamata anche congettura 3N+1, afferma che dato un qualsiasi numero positivo N, il seguente algoritmo termina sempre:

- Se N è 1, termina;
- Se N è pari, dividi N per 2;
- Se N è dispari, moltiplica N per 3 e somma 1.

Ad esempio: se N = 6, si ottiene: 6, 3, 10, 5, 16, 8, 4, 2, 1. La sequenza così ottenuta ha lunghezza 9 (considerando anche il 6 iniziale e l'1 finale): in altre parole, l'algoritmo effettua 9 iterazioni prima di terminare.

Nel programma riportato in "esercizio5.s" in "Lab08Esercizi.zip" la funzione main è già implementata, mentre la funzione ***collatz*** è da implementare. Il main legge un numero positivo da input, invoca la funzione ***collatz*** e stampa tale lunghezza.

Si implementi la funzione ***collatz*** in modo ricorsivo: dato un numero positivo la funzione dovrebbe restituire la lunghezza della corrispondente sequenza di collatz

Suggerimento: In Python, l'implementazione della funzione ***collatz*** sarebbe la seguente:

```
def collatz(N):
    if(N==1):
        return 1
    elif(N%2==0):
        return collatz(N/2)+1
    else:
        return collatz(N*3+1)+1
```

Esercizio 6

Partendo dagli esempi "automatic-variables.s" e "automatic-variables2.s" visti a lezione e riportati in "Lab08Esercizi.zip" realizzare due programmi come specificato di seguito.

a) Un programma che legga da input due array di interi, come variabili automatiche con dimensione fissata a 5 e stampi la somma ottenuta sommando tutti gli interi contenuti nei due array.
b) Un programma che legga da input due array di interi, come variabili automatiche con dimensione letta da input e stampi la somma ottenuta sommando tutti gli interi contenuti nei due array.

Esercizio 7

Il programma riportato in "esercizio7.s" in "Lab08Esercizi.zip" definisce alcune *format string* utili per allineare testo o numeri a destra o sinistra, con e senza "padding" (ovvero, riempimento):

- `fmt_str_dx` permette di stampare una stringa allineata a destra,
- `fmt_str_sx` permette di stampare una stringa allineata a sinistra,
- `fmt_int_32_dx` permette di stampare un intero a 32 bit allineato a destra,
- `fmt_int_32_sx` permette di stampare un intero a 32 bit allineato a sinistra,
- `fmt_int_32_dx_padding` permette di stampare un intero a 32 bit allineato a destra r in cui lo zero è usato come carattere "riempitore",
- `fmt_int_64_dx` permette di stampare un intero a 32 bit allineato a destra,
- `fmt_int_64_sx` permette di stampare un intero a 64 bit allineato a sinistra,
- `fmt_int_64_dx_padding` permette di stampare un intero a 64 bit allineato a destra r in cui lo zero è usato come carattere "riempitore",
- `fmt_str` permette di stampare due stringhe: una allineata a destra e una a sinistra,
- `fmt_int_32` permette di stampare tre interi a 32 bit: uno allineato a destra, uno allineato a sinistra e uno allineato a destra con riempimento,
- `fmt_int_64` permette di stampare tre interi a 64 bit: uno allineato a destra, uno allineato a sinistra e uno allineato a destra con riempimento.

Nel main vengono utilizzate tali *format string* per effettuare alcune stampe. Eseguire il programma e successivamente modificarlo per fare in modo di stampare:

- il numero 123 come numero a 32 bit allineato a destra,
- il numero 1000 come numero a 64 bit allineato a sinistra,
- il numero 1999 come numero a 64 bit allineato a destra con riempimento,
- la stringa "ciao" allineata a destra seguita dalla stringa "hello" allineata a sinistra (usando `fmt_str`).

Realizzare ora una *format string* che permetta di stampare una stringa centrata con 10 spazi a destra e 10 a sinistra e stampare la stringa "ciao" tramite tale *format string*; si dovrà cioè ottenere la stampa:

```
|          ciao          |
```

Per maggiori informazioni sulle *format string* e sull'uso con la funzione *printf*, si consiglia di consultare la documentazione: https://www.cplusplus.com/reference/cstdio/printf