

Esercizio 1

Considerare il programma riportato nel file “esercizio1.s” e indicare commenti, etichette, istruzioni e direttive di cui si compone. Assemblare e linkare il programma con il comando:

```
aarch64-linux-gnu-gcc -o esercizio1 -static esercizio1.s
```

Eeguire quindi il programma con il seguente comando. Cosa si ottiene in output?

```
qemu-aarch64 esercizio1
```

N.B.: I comandi su riportati si riferiscono al caso in cui si utilizza l’emulatore suggerito a lezione (già preinstallato nella macchina virtuale fornita).

Esercizio 2

Considerare il seguente programma riportato nel file “esercizio2.s” in cui vengono dichiarate alcune variabili: a destra sono riportate le equivalenti dichiarazioni in Python.

<pre>.global main .data list: .word 0, 1, 2, 3 bool: .byte 0 n: .word 123</pre>	<pre>list = [0, 1, 2, 3] bool = False n = 123</pre>
--	---

- a. Analizzare l’allineamento della memoria con i comandi:

```
aarch64-linux-gnu-gcc -c esercizio2.s
aarch64-linux-gnu-objdump -s esercizio2.o
```

- b. Arricchire il programma delle direttive .align che si ritengono necessarie e verificare nuovamente se la memoria risulta allineata.
- c. Ripetere tutti i passaggi, cambiando la definizione di n come segue.

```
n: .quad 123
```

N.B.: I comandi su riportati si riferiscono al caso in cui si utilizza l’emulatore suggerito a lezione (già preinstallato nella macchina virtuale fornita).

Esercizio 3

Considerare il seguente programma riportato nel file “esercizio3.s”, in cui vengono dichiarate alcune variabili: a destra sono riportate le equivalenti dichiarazioni in Python.

<pre>.global main .data a: .byte 0, 1, 2 b: .hword 15 c: .byte 'A' d: .word 5, 2 e: .quad 9, 4 f: .asciz "Ciao"</pre>	<pre>a = [0, 1, 2] b = 15 c = 'A' d = [5, 2] e = [9, 4] f = "Ciao"</pre>
--	--

- a. Arricchire il programma delle direttive `.align` che si ritengono necessarie facendo in modo di limitare il più possibile il numero di bit 0 aggiunti.
- b. Analizzare l'allineamento della memoria con i comandi:

```
aarch64-linux-gnu-gcc -c esercizio3.s  
aarch64-linux-gnu-objdump -s esercizio3.o
```

Quanti bit sono stati "sprecati"?

Quanti ne verrebbero sprecati se si usassero soltanto direttive `.align 3`?

- c. Riordinare le direttive in modo tale che non sia necessario aggiungere alcun `.align`.

N.B.: I comandi su riportati si riferiscono al caso in cui si utilizza l'emulatore suggerito a lezione (già preinstallato nella macchina virtuale fornita).

Esercizio 4

Considerare il programma riportato nel file "esercizio4.s":

- a. Indicare la parte di programma che corrisponde alla dichiarazione di una macro.
- b. Indicare la/le invocazioni della macro presenti nel programma.
- c. Assemblare ed eseguire il programma con i seguenti comandi e verificare cosa si ottiene in output:

```
aarch64-linux-gnu-gcc -o esercizio4 -static esercizio4.s  
qemu-aarch64 esercizio4
```

- d. Modificare il programma in modo tale da stampare soltanto il numero 15.

N.B.: I comandi su riportati si riferiscono al caso in cui si utilizza l'emulatore suggerito a lezione (già preinstallato nella macchina virtuale fornita).

Esercizio 5

Considerare il programma riportato nel file "esercizio5.s":

1. Assemblare ed eseguire il programma con i seguenti comandi e verificare cosa si ottiene in output:

```
aarch64-linux-gnu-gcc -o esercizio5 -static esercizio5.s  
qemu-aarch64 esercizio5
```
2. Cosa cambierebbe se `.equ CONSTANT, 1` fosse sostituito con `.equ CONSTANT, 10` ?
3. Commentare `.equ CONSTANT, 1` quindi assemblare ed eseguire nuovamente il programma. Il programma si comporta in modo diverso: a cosa è dovuto questo comportamento?
4. Aggiungere una seconda costante `CONSTANT2` e modificare il programma in modo tale che stampi 1 quando `CONSTANT` e `CONSTANT2` sono definite, 0 altrimenti.

N.B.: I comandi su riportati si riferiscono al caso in cui si utilizza l'emulatore suggerito a lezione (già preinstallato nella macchina virtuale fornita).