

# Programmazione e Algoritmi 1

A.A. 2023/24 — Esercizio aggiuntivo al compito di prova

prof. Gianluca Amato

In questo esercizio aggiuntivo del compito di prova vogliamo mostrare come trattare il caso di esecuzione passo passo di un programma che opera su tipi di dati mutabili (fondamentalmente, le liste). In particolare, consideriamo di voler eseguire passo passo il seguente programma:

```
1 def possum(l, end):
2     for i in range(end):
3         l[i] += i + 1
4     return
5
6 dati = [1, 10]
7 for i in range(1, len(dati)+1):
8     possum(dati,i)
9 print(dati)
```

Utilizzando il solito modulo, otteniamo:

riga programma	valore variabili	note
6	dati=[1, 10]	
7	dati=[1, 10] i=1	
8	dati=[1, 10] i=1	call possum(dati, 1)
1	l=dati=[1, 10] end=1	
2	l=dati=[1, 10] end=1 i=0	
3	l=dati=[2, 10] end=1 i=0	
4	l=dati=[2, 10] end=1 i=0	return
8	dati=[2, 10] i = 1	

È importante osservare le ultime due righe. La funzione `possum` ha modificato la lista `l`. Ma in realtà le liste `l` e `dati` sono la stessa lista. Questo lo evidenzio scrivendo `l=dati=[...]` invece di semplicemente `l=[...]` nel valore delle variabili. Pertanto, la modifica di `l` è visibile anche in `dati`. Per questo motivo, nell'ultima riga, il valore di `dati` è cambiato rispetto al momento prima della chiamata. Continuo adesso da dove avevo interrotto.

riga programma	valore variabili	note
7	dati=[2, 10] i = 2	
8	dati=[2, 10] i = 2	call possum(dati, 2)
1	l=dati=[2, 10] end=2	
2	l=dati=[2, 10] end=2 i=0	
3	l=dati=[3, 10] end=2 i=0	

	2			l=dati=[3, 10] end=2 i=1	
	3			l=dati=[3, 12] end=2 i=1	
	4			l=dati=[3, 12] end=2 i=1	return
8				dati=[3, 12] i = 2	
9				dati=[3, 12] i = 2	print [3, 12]