

# Corso di Programmazione 1

## Quinta Esercitazione di Laboratorio

### Esercizio 1

Scrivere un programma Java che legga da tastiera un intero positivo  $n$  e stampi a video la più piccola potenza di 2 maggiore di  $n$ .

### Esercizio 2

Scrivere un programma Java che, lette da tastiera due stringhe  $a$  e  $b$ , calcoli la lunghezza del massimo prefisso comune. Ad esempio, se  $a = \text{cavallo}$  e  $b = \text{cavallerizza}$ , il programma stamperà a video:

```
Il massimo prefisso comune tra le stringhe "cavallo"  
e "cavallerizza" ha lunghezza 6
```

in quanto `cavall` è la parte iniziale in comune ad entrambe le stringhe.

### Esercizio 3

Scrivere un programma Java che legga da tastiera un intero positivo e stampi a video la scomposizione in fattori (primi) del numero. Ad esempio, se il numero letto è 12, il programma stamperà:

```
Numero letto: 12  
Fattori: 2 (2 volte), 3 (1 volta)
```

### Esercizio 4

Scrivere un programma Java che calcoli la somma di due numeri interi positivi letti da tastiera, utilizzando solo operazioni di post-incremento e post-decremento.

### Esercizio 5

La *congettura di Collatz* è un famoso problema matematico tuttora irrisolto. Tale congettura afferma che, partendo da un qualsiasi numero intero positivo  $n$ , applicando ripetutamente le seguenti operazioni si ottiene una sequenza di interi che termina sempre con 1:

- se  $n = 1$ , termina
- se  $n$  è pari, poni  $n = n / 2$
- se  $n$  è dispari (maggiore di 1), poni  $n = 3n + 1$

Scrivere un programma Java che, letto da tastiera un intero positivo  $n$ , stampi a video la sequenza di numeri ottenuti applicando ripetutamente le operazioni indicate sopra. Si noti che il programma termina solo se la congettura di Collatz vale per il numero  $n$  inserito.

**Ad esempio**, se  $n = 10$ , il programma stamperà a video la sequenza: 10, 5, 16, 8, 4, 2, 1.

**Variante:** Scrivere un programma Java che, letto da tastiera un numero intero positivo `numeroMassimo`, determini se la congettura di Collatz vale per ogni numero compreso tra 1 e `numeroMassimo`.

**Ad esempio**, se `numeroMassimo = 5`, il programma stamperà a video:

```
Provo la congettura per n = 1 ... Ok, vale.  
Provo la congettura per n = 2 ... Ok, vale.  
Provo la congettura per n = 3 ... Ok, vale.  
Provo la congettura per n = 4 ... Ok, vale.  
Provo la congettura per n = 5 ... Ok, vale.
```

### Esercizio 6

Scrivere un programma Java che, lette da tastiera due stringhe *genoma* e *proteina*, entrambe formate dalle lettere A, C, G, T (che indicano rispettivamente *Adenina*, *Citosina*, *Guanina* e *Timina*, le quattro basi azotate che compongono il DNA) stampi a video tutte le occorrenze della stringa *proteina* all'interno della stringa *genoma*, **senza usare i metodi `substring()` e `indexOf()` della classe `String`**.

**Ad esempio**, se *genoma* = ACCAGTCAGTGCAATC e *proteina* = AGT, il programma stamperà a video:

La sequenza AGT compare in ACCAGTCAGTGCAATC nelle  
seguenti posizioni: 4, 8

Si noti che la prima posizione ha indice 1.

**Altro esempio:** se *genoma* = AAAAAAA e *proteina* = AAA, il programma stamperà a video:

La sequenza AAA compare in AAAAAAA nelle seguenti  
posizioni: 1, 2, 3, 4, 5

### Esercizio 7

Scrivere un programma Java che, lette da tastiera due stringhe composte dalle cifre da 0 a 9, che rappresentano due numeri interi  $n$  e  $m$ , calcoli il prodotto  $n*m$  usando l'algoritmo imparato alle scuole elementari (e stampandone a video i passaggi). Ad esempio, se  $n = 2431$  e  $m = 523$ , abbiamo:

```
  2431
   523
  ----
  7293
 4862
12155
-----
1271413
```

Quindi il programma stamperà a video (non necessariamente allineate come mostrato sopra) le stringhe: 7293, 4862, 12155, 1271413.