

# Laboratorio di programmazione

23 novembre 2016

## Primi gemelli e numeri amichevoli

Provate a scrivere una soluzione dei problemi suddetti dell'ultimo laboratorio utilizzando un metodo statico `boolean primo(int)` che restituisce `true` se l'argomento intero è primo, e un metodo statico `int sommadiv(int)` che restituisce la somma dei divisori di un numero dato.

## Garibaldi fu ferito

Scrivete un programma che legga righe in input e la stampa dopo aver sostituito tutte le vocali con delle "u". Il programma si deve fermare quando viene inserita la riga vuota. Suggerimento: utilizzate l'istruzione `switch` per scrivere un metodo statico che accetti un carattere e lo restituisca trasformato opportunamente.

### Esempio di funzionamento

```
mamma mia  
mummu muu  
bella aiuola  
bullu uuuulu  
Garibaldi fu ferito  
Gurubuldu fu furutu
```

Successivamente, modificate il programma in modo che invece di sostituire tutte le vocali con delle "u", le sostituisca con la *prima vocale* che compare in ogni riga (modificate il metodo statico in modo che prenda come argomento la lettera da sostituire).

### Esempio di funzionamento

```
mamma mia  
mamma maa  
bella aiuola  
belle eeeele  
Garibaldi fu ferito  
Garabalda fa farata
```

## Effetti collaterali

Scrivete un programma che dichiari tre variabili intere `x`, `y` e `z`. Fate inserire dall'utente un valore per ciascuna delle tre variabili, e stampate il valore di diverse espressioni con effetti collaterali a vostra scelta. Osservate i risultati e confrontateli con la semantica spiegata a lezione.

Se non avete abbastanza fantasia per creare da soli delle espressioni, ecco alcuni suggerimenti:

```
System.out.println( ( x++ + --z ) * y++ );  
System.out.println( ( x = 2 ) * z-- * y );  
System.out.println( ( z /= 2 ) + ++x + --y );
```

## Alfabeto farfallino

Quando i vostri docenti di laboratorio di programmazione erano bambini, usavano a volte, per comunicare con i loro simili, uno speciale alfabeto, detto *alfabeto farfallino*<sup>1</sup>. L'alfabeto farfallino consiste nel sostituire, a ciascuna vocale, una sequenza di tre lettere della forma vocale-f-vocale. Per esempio, alla lettera *a* viene sostituita la sequenza *afa*, alla lettera *e* la sequenza *efe* e così via.

Dovete scrivere un programma che legge righe scritte dall'utente e ne stampi la traduzione in alfabeto farfallino. Potete assumere che il testo in input in input non contenga lettere maiuscole. Il programma deve fermarsi quando l'utente inserisce una riga vuota.

### Esempio di funzionamento

```
mamma mia  
mafammafa mifiafa  
aiuola  
afaiufuofolafa  
farfalla  
fafarfafallafa
```

## La strana sillabazione

Il professor Precisini, dell'*Accademia della crusca*, sostenendo che le regole di sillabazione della lingua italiana sono troppo complesse e piene di eccezioni, propone un nuovo e originale metodo di sillabazione. Il metodo consiste in questo: una sillaba è una sequenza massimale di caratteri consecutivi che rispettano l'ordine alfabetico. Per esempio, la parola *ambire* viene sillabata come *am-bir-e*: infatti la lettera *a* precede la lettera *m*, e le lettere *b*, *i* e *r* rispettano anch'esse l'ordine. Analogamente, la parola *sotterfugio* viene sillabata come *s-ott-er-fu-gio*.

Dovete scrivere un programma che legge righe scritte dall'utente e ne stampi la versione sillabata. Potete assumere che il testo in input in input non contenga lettere maiuscole. Il programma deve fermarsi quando l'utente inserisce una riga vuota.

### Esempio di funzionamento

```
amore  
amor-e  
medicina  
m-e-di-cin-a  
scafroglia  
s-c-afr-o-gl-i-a
```

<sup>1</sup>Non si può escludere che qualcuno lo faccia ancora, in alcune remote regioni d'Italia...

## Il cifrario di Cesare

Svetonio nella *Vita dei dodici Cesari* racconta che Giulio Cesare usava per le sue corrispondenze riservate un codice di sostituzione molto semplice, nel quale la lettera chiara veniva sostituita dalla lettera che la segue di tre posti nell'alfabeto: la lettera A è sostituita dalla D, la B dalla E e così via fino alle ultime lettere che sono cifrate con le prime.

Più in generale si dice codice di Cesare un codice nel quale la lettera del messaggio chiaro viene spostata di un numero fisso  $k$  di posti, non necessariamente tre.

Dovete scrivere un programma che chieda in input una stringa da cifrare e il numero  $k$  (la chiave di cifratura), e che emetta in output la stringa cifrata; il programma deve cifrare solo le lettere dell'alfabeto, mantenendo minuscole le minuscole, e maiuscole le maiuscole, mentre deve lasciare inalterati gli altri simboli. Assumete che la stringa non contenga lettere accentate.

Per provare se il programma funziona, cifrate un messaggio con una certa chiave  $k$  e poi applicate al risultato una nuova cifratura con chiave  $26 - k$ : il risultato dovrebbe essere la stringa originale.

### Esempio di funzionamento

Frase da cifrare: <b>mamma li Turchi</b> Chiave: <b>5</b> Frase cifrata: rfrrf qn Yzwhmn
--

Frase da cifrare: <b>rfrrf qn Yzwhmn</b> Chiave: <b>21</b> Frase cifrata: mamma li Turchi
---

### Suggerimenti

- Per “spostare” ciclicamente di  $k$  posti un carattere  $c$  potete usare l'espressione

$$(\text{char})( ( ( c - 'A' ) + k ) \% 26 + 'A' )$$

nel caso  $c$  sia maiuscolo,

$$(\text{char})( ( ( c - 'a' ) + k ) \% 26 + 'a' )$$

nel caso  $c$  sia minuscolo (perché?).

- Per stabilire se un carattere è una lettera maiuscola o minuscola potete usare i metodi statici

```
Character.toUpperCase(char)
```

e

```
Character.toLowerCase(char)
```

### Massimo prefisso/suffisso comune

Scrivete un programma che legga due stringhe dalla e ne stampi il massimo prefisso e il massimo suffisso comune. Il *massimo prefisso comune* di due stringhe  $s_0s_1 \dots s_{m-1}$  e  $t_0t_1 \dots t_{l-1}$  è la sequenza  $s_0s_1 \dots s_j$  di massima lunghezza (cioè di massimo  $j$ ) tale che  $s_0 = t_0, s_1 = t_1, \dots, s_j = t_j$ . Analogamente viene definito il massimo suffisso comune.

### Esempio di funzionamento

Prima stringa: <b>mamma</b> Seconda stringa: <b>madre</b> Massimo prefisso: 'ma' Massimo suffisso: ''
--

Prima stringa: <b>pippo</b> Seconda stringa: <b>pappo</b> Massimo prefisso: 'p' Massimo suffisso: 'ppo'
--

Prima stringa: <b>foo</b> Seconda stringa: <b>foo</b> Massimo prefisso: 'foo' Massimo suffisso: 'foo'
--