

Laboratorio di programmazione

16 novembre 2016

- Lo scheletro dei vostri programmi deve essere

```
import java.io.*;
import java.util.*;

public class ... {
    public static void main( String[] args ) throws Exception {
        ...
    }
}
```

dove i due gruppi di puntini devono essere sostituiti dal nome scelto per la classe e dal suo codice. Per il nome potete usare lettere maiuscole, minuscole e numeri, purché il nome inizi con una lettera. Ricordatevi che il nome del file sorgente *deve* essere identico al nome della classe.

Per leggere dati da tastiera dovete creare uno `Scanner`:

```
Scanner scanner = new Scanner( System.in );
```

Una volta creato lo `Scanner` (va creato *una sola volta*) potete leggere l'input da tastiera dell'utente in vari modi:

- Per leggere una riga: `String line = scanner.nextLine();`
- Per leggere una parola: `String line = scanner.next();`
- Per leggere un intero: `int x = scanner.nextInt();`
- Per leggere un long: `long l = scanner.nextLong();`
- Per leggere un double: `double d = scanner.nextDouble();`

- Per stampare, potete usare `System.out.println(...)`, che va a capo, e `System.out.print(...)`, che non va a capo.
- Se avete più stringhe, le potete concatenare con l'operatore `+`, che si occupa anche di convertire in stringa qualunque altro oggetto o variabile:

```
Scanner scanner = new Scanner( System.in );
System.out.print( "Scrivi qualcosa: " );
String input = scanner.nextLine();
System.out.println( "Hai scritto " + input );
```

- Se volete leggere un numero:

```
Scanner scanner = new Scanner( System.in );
System.out.print( "Scrivi un numero: " );
int x = scanner.nextInt();
int quadrato = x * x;
System.out.println( "Hai scritto " + x + " che ha quadrato " + quadrato );
```

Numeri perfetti

Un numero è *perfetto* se è uguale alla somma dei suoi divisori propri. Per esempio, $6 = 1 + 2 + 3$ è perfetto. Scrivete un programma che, dato un intero in input, scriva i numeri perfetti minori dell'intero dato.

Esempio di esecuzione

```
Limite? 1000
6
28
496
```

Spaziate

Scrivete un programma che legga una riga introdotta dall'utente e ne stampi i caratteri a spaziatura doppia.

Esempio di input

```
Ma Ugo, non l'abbiamo mai fatto!
```

Esempio di output

```
M a U g o , n o n l ' a b b i a m o m a i f a t t o !
```

Palindrome

Una parola è *palindroma* se è uguale quando viene letta da destra a sinistra e da sinistra a destra. Quindi "enne" è palindroma, ma "papa" non lo è. Scrivete un programma che legga una stringa da tastiera e dica se è palindroma o no.

Esempio di input

```
abba
```

Esempio di output

```
abba è palindroma
```

Esempio di input

```
pippo
```

Esempio di output

```
pippo non è palindroma
```

Primi gemelli

Due primi p e q sono *gemelli* se $p = q + 2$. Scrivete un programma che stampi i primi gemelli minori di un intero fornito dall'utente.

Esempio di esecuzione

```
Numero? 20  
3 5  
5 7  
11 13  
17 19
```

La prova del nove

La prova del nove è un meccanismo per il controllo dei calcoli basato sull'aritmetica modulare. Sfrutta il fatto che dovendo controllare se $x \cdot y = z$, possiamo controllare se $x \cdot y \bmod 9 = z \bmod 9$, dove \bmod è il resto della divisione intera (in C, %). Scrivete un programma che enumeri le terne di interi x , y , z minori di un intero dato dall'utente e che stampi quelle per cui la prova del nove non funziona (cioè $x \cdot y \neq z$ ma $x \cdot y \bmod 9 = z \bmod 9$).

Esempio di esecuzione

```
Limite? 5  
3 3 0  
3 4 3  
4 3 3
```

Lunghezza media delle parole

Scrivete un programma che legga una frase (cioè, un'intera riga) e che calcoli quante parole contiene e la loro lunghezza media.

Esempio di funzionamento

```
Frase: S'i' fosse foco, arderei 'l mondo  
La frase contiene 7 parole, con lunghezza media 3.43.
```

Suggerimenti

- Per contare quante parole contiene una stringa potete controllare quanti sono i caratteri alfabetici seguiti da caratteri non alfabetici; per vedere se un carattere c è alfabetico potete usare il metodo statico

```
Character.isLetter(char)
```

Il metodo restituisce vero quando il suo argomento è alfabetico. Dovete ovviamente stare attenti a quel che fate sull'ultimo carattere.

- Per stampare un numero double con un certo numero cifre decimali, anziché usare `System.out.println()` potete usare `System.out.printf()`, che accetta una stringa di formato. Per esempio,

```
System.out.printf("La frase contiene %d parole, con lunghezza media %.2f\n",  
n, media);
```

assumendo che la variabile intera `n` contenga il numero di parole e che la variabile double `media` contenga la media.

Numeri amichevoli

Due numeri interi (x, y) sono detti *amichevoli* se la somma dei divisori di ciascuno è uguale all'altro (fra i divisori è compreso l'1 escluso il numero stesso). Ad esempio $(220, 284)$ è una coppia di amichevoli, essendo

$$\begin{aligned} 284 &= 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 \text{ (che sono i divisori di 220)} \\ 220 &= 1 + 2 + 4 + 71 + 142 \text{ (che sono i divisori di 284)}. \end{aligned}$$

Gli antichi greci ritenevano che le coppie di numeri amichevoli avessero certe proprietà mistiche.

Scrivete un programma che, dato un numero n inserito dall'utente, stampi tutte le coppie di numeri amichevoli (x, y) con $x \leq y \leq n$

Esempio di funzionamento

```
Limite: 1000  
6 6  
28 28  
220 284  
496 496
```