# Laboratorio di programmazione

30 novembre 2016

## Fattoriale

Scrivete un programma che tramite un metodo statico ricorsivo calcoli il fattoriale $n! = 1 \cdot 2 \cdot 3 \cdot \cdots \cdot (n-1) \cdot n$.

## Potenze

Scrivere un programma che tramite un metodo statico ricorsivo dati un reale $\alpha$ e un intero nonnegativo $k$ calcoli $\alpha^k$.

## La funzione misteriosa

Dove scrivere un programma che calcoli la seguente funzione ricorsiva definita sugli interi strettamente positivi:

$$f(x) = \begin{cases} 0 & \text{se } x = 1; \\ 1 + f(\lfloor x/2 \rfloor) & \text{altrimenti.} \end{cases}$$

Il programma deve chiedere all'utente un input e stampare il risultato del calcolo di $f$. Utilizzate il programma per calcolare la funzione su vari input. Cercate di indovinare di che funzone si tratta.

## Partizioni di un intero

Una *partizione di n* è una sequenza di interi $a_1, \ldots, a_k$ con $a_1 \leq a_2 \leq \cdots \leq a_k$ e $a_1 + a_2 + \cdots + a_k = n$ (i singoli $a_i$ vengono chiamati *parti* della partizione). Per esempio, le partizioni di 4 sono le seguenti: (1, 1, 1, 1), (1, 1, 2), (1, 3), (2, 2), (4). Quindi 4 ha 5 partizioni distinte. Indicate con $\pi(n)$ il numero di partizioni dell'intero $n$.

Dovete scrivere un programma che, preso come argomento (sulla linea di comando) $n$, calcoli e stampi $\pi(n)$.

### Esempio di funzionamento

```
java Partizioni
6
11
```

Infatti le partizioni di 6 sono: (1, 1, 1, 1, 1, 1), (1, 1, 1, 1, 2), (1, 1, 1, 3), (1, 1, 2, 2), (1, 1, 4), (1, 2, 3), (1, 5), (2, 2, 2), (2, 4), (3, 3), (6).

### Suggerimenti

Predisponete un metodo statico ricorsivo:

```
int npart(int n, int k)
```

che conta quante sono le partizioni dell'intero $n$ aventi parti tutte maggiori o uguali a $k$ (si assume sempre che $k \leq n$). Notate che:

- se $n = 0$, il metodo deve restituire 1 (esiste una sola partizione dell'intero 0);
- se $n > 0$, una partizione di $n$ avente parti $\geq k$ può essere ottenuta scegliendo la prima parte della partizione come un numero $i = k, k + 1, \ldots, n$, e contando quante sono le partizioni di $n - i$ aventi parti $\geq i$.

## Il codice di Vigénère

Un sistema di cifratura molto diffuso fin dal XVI secolo è il cosiddetto *codice di Vigénère*, una variante polialfabetica del cifrario di Cesare.

Supponete di avere un *testo in chiaro*, costituito semplicemente da una sequenza di caratteri alfabetici. Per applicare il codice di Vigénère, occorre anche avere una *chiave di cifratura*, spesso chiamata "verme".

Il testo in chiaro e il verme vengono scritti uno sopra l'altro (il verme viene, se necessario, ripetuto più volte e/o troncato, in modo che le due sequenze di caratteri abbiano la stessa lunghezza). Quindi i due testi vengono sommati lettera per lettera. In pratica, questo corrisponde a identificare ogni lettera dell'alfabeto con un numero fra 0 e 25, e nell'effettuare le somme modulo 26.

Ad esempio, se il testo fosse `ARRIVANOIRINFORZI` e il verme fosse `VERME`:

```
ARRIVANOIRINFORZI
VERMEVERMEVERMEVE
VVIUZVRFUVDRWAVUM
```

Infatti A+V=V (essendo A la 0-esima lettera e V la 21-esima lettera, A+V=0+21=21=V), R+E=V (R è la 17-esima lettera e E la quarta, R+E=17+4=21=V) eccetera.

Notate inoltre che lo stesso sistema si può usare anche per decifrare un testo cifrato: è sufficiente sostituire al verme usato per cifrare quello "opposto" (sostituendo a ogni A una A, a ogni B una Z, a ogni C una Y, a ogni D una X ecc.).

Dovete scrivere un programma che legga da tastiera un verme (costituito solo da lettere maiuscole), poi legga il testo in chiaro (anch'esso costituito solo da lettere maiuscole) e stampi il testo cifrato.

Dopo aver scritto e provato il programma, provate a decodificare il seguente messaggio (sapendo che era stato codificato con il verme `CANE`, il cui opposto è `YANW`):

```
KNIMCTRGKVRRVIXKFIAYVEYPC
```

#### Esempio di funzionamento

```
java Vigenere
Verme: VERME

Testo in chiaro: ARRIVANOIRINFORZI

Testo cifrato: VVIUZVRFUVDRWAVUM
```

```
java Vigenere
Verme: FWJOW

Testo in chiaro: VVIUZVRFUVDRWAVUM

Testo cifrato: ARRIVANOIRINFORZI
```

### Suggerimenti

- Se il verme è lungo $m$ caratteri, il carattere di posto $i$ del messaggio deve essere cifrato utilizzando il carattere di posto $i \bmod m$ del verme.

## Frazioni continue

Se $a_0$ è un numero intero qualsiasi, e $a_1, a_2, \ldots, a_n$ sono interi positivi, la notazione $[a_0, a_1, \ldots, a_n]$ sta per l'espressione

$$[a_0, a_1, \ldots, a_n] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}},$$

e viene chiamata *frazione continua*. Ad esempio,

$$[-1, 5, 2, 4] = -1 + \cfrac{1}{5 + \cfrac{1}{2 + \frac{1}{4}}}.$$

Ovviamente, $[a_0, a_1, \ldots, a_n]$ è un numero razionale, e si può dimostrare che per ogni numero razionale esiste una sola frazione continua che lo rappresenti a meno di eventuali 1 in fondo (poiché, ad esempio, $[-1, 5, 2, 4] = [-1, 5, 2, 3, 1]$). Inoltre, per ogni numero irrazionale esiste un'unica frazione continua infinita che lo rappresenta.

Scegliete un valore di $n$ (p.es., $n = 4$) e scrivete un programma che riceva in input $a_0, a_1, \ldots, a_n$ e stampi in output $[a_0, a_1, \ldots, a_n]$. In particolare, usate questo programma per calcolare $[1, 1, \ldots, 1]$. Che valore ottenete? Riuscite a immaginare che "celebre" valore rappresenta la frazione infinita $[1, 1, 1, \ldots]$?

**Suggerimento:** Se chiamate $x$ il valore della frazione continua $[1, 1, 1, \ldots]$, vale che

$$x = 1 + \frac{1}{x},$$

e quindi. . .