# Esempi di esercizi per la prova pratica sul linguaggio Python

October 29, 2019

## 1 Python basics

1. Scrivere una funzione che riceve come parametri due liste di numeri interi e restituisce una terza lista contenente gli interi dispari della prima e gli interi pari della seconda. In caso di errore di tipo nelle liste di ingresso la funzione deve restituire `False`.

2. Scrivere una funzione che riceve in ingresso due dizionari `A` e `B` e restituisce un dizionario `C` così formato: (1) le chiavi di `C` sono (tutte e sole) le chiavi presenti sia in `A` che in `B`; (2) se $k$ è una chiave di `C`, allora $\texttt{C}[k] = (v_{Ak}, v_{Bk})$, dove $v_{Ak}$ e $v_{Bk}$ sono rispettivamente di valori corrispondenti a $k$ in `A` e in `B`.

3. Scrivere una funzione che riceve in ingresso due insiemi `A` e `B` di numeri interi positivi e, opzionalmente, un intero `L`; la funzione rimuove da `A` tutti gli elementi maggiori di `L` che appartengono anche a `B`. Se `L` non è specificato, il filtro non si applica. In caso di errore di tipo, la funzione deve stampare un messaggio opportuno e lasciare `A` inalterato.

## 2 Classes

4. I cosiddetti numeri *Catalani* sono così definiti:

$$C_n = \frac{1}{n+1}\binom{2n}{n} \qquad n = 1, 2, \ldots$$

Implementare un iterabile per i primi $n$ numeri Catalani.

5. Implementare un'opportuna sottoclasse di `list` costituita solo da liste di interi e che supporti il confronto così definito: $L_1 == L_2$ se e solo se la somma degli elementi di $L_1$ è uguale della somma degli elementi di $L_2$. Sulla base di questo criterio deve implemntare anche tutti gli altri operatori relazionali.

6. Definire una classe per operare sui numeri complessi. Un numero complesso deve essere visto e trattato come coppia di numeri reali rappresentanti, rispettivamente, parte reale e parte immaginaria. La clase deve supportare (come minimo) le operazioni di somma e prodotto di numeri complesso, oltre al calcolo del coniugato di un numero complesso.

# 3 Function decorators

7. Scrivere un semplice decoratore `greeting` per stampare, all'inizio di ogni chiamata di funzione, il messaggio:

   `Hi, this is` ⟨nome della funzione⟩

8. In relazione all'esercizio 3, definire un decoratore in cui sono inseriti i controlli di tipo