

# Simulazione esame di Programmazione

Ing. Cristiano Gregnanin

May 19, 2015

## 1 Esercizio

Un professore salva gli esiti degli esami (max 100) in un file di testo *esiti.txt* con i seguenti dati:

- *cognome* dello studente (stringa di max 20 caratteri, senza spazi)
- *mese* in cui si è sostenuto l'appello (int)
- *voto* conseguito dallo studente (int)

Esempio di *esiti.txt*:

Rossi	10	18
Rossi	1	30
Bianchi	2	30
Verdi	7	25
Bianchi	10	27

Si desidera sapere quanti appelli ha sostenuto ogni studente. Si scriva un programma che calcola questa informazione, utilizzando il seguente algoritmo:

1. Si legga, tramite una funzione *leggi*, il file *esiti.txt* e se ne porti il contenuto in un' opportuna lista di dizionari. La funzione deve:
  - (a) mostrare a video il contenuto della lista opportunamente formattata.
  - (b) ritornare la lista degli esiti.
2. Nel programma, si invochi una funzione *elabora* (che verrà definita al punto 3), che:
  - (a) prende in ingresso la lista generata al punto 1 (ed eventualmente altri parametri)
  - (b) aggiorni una lista di dizionari *studenti* contenente per ogni studente il nome ed il numero di appelli sostenuti
3. Si implementi la funzione *elabora* utilizzata al punto 2. Per fare questo, si scorra la lista generata al punto 1 e, per ciascun elemento della lista, si invochi una funzione *inserimento* (che verrà definita al punto 4). La funzione *inserimento* prende in ingresso:

- (a) un esito (cognome, mese, voto)
- (b) la lista di strutture *studenti* descritta al punto 2b (più, eventualmente altri parametri).

Questa funzione aggiorna la lista *studenti* con i dati dell'appello dato come parametro. Ad esempio, se la lista *studenti* è la seguente:

Cognome	Numero
Rossi	1
Bianchi	2

e l'esito contiene (Rossi,3,25) , nella lista *studenti* si dovrà incrementare di 1 il numero di appelli sostenuti da Rossi, cioè:

Cognome	Numero
Rossi	2
Bianchi	2

Se a questo punto si inserisse l'esito (Verdi,4,28) , la lista diventerebbe:

Cognome	Numero
Rossi	2
Bianchi	2
Verdi	1

4. Si implementi la funzione *inserimento* utilizzata al punto 3. Si utilizzi il seguente algoritmo. Si cerchi nella lista *studenti* lo studente il cui cognome compare nell'esito dell'appello. Questo viene effettuato con una funzione *ricerca* (che verrà implementata al punto 5) che prende in ingresso la lista *studenti* ed il cognome di uno studente e fornisce:

- la posizione (indice della lista) in cui compare lo studente
- oppure -1 se lo studente non è presente nella lista

Se lo studente compare nella lista (ovvero, se *ricerca* ha fornito un valore maggiore o uguale a 0), la funzione *inserimento* incrementa il numero di appelli di quello studente. Altrimenti (se la *ricerca* ha fornito -1), la funzione *inserimento* aggiunge lo studente in fondo alla lista *studenti*, con numero di appelli sostenuto uguale a 1.

5. Si implementi la funzione *ricerca* , utilizzata al punto 4.
6. Scrivere una funzione di ordinamento *bubblesort* che prende in ingresso la lista *studenti* e restituisce la lista ordinata per cognome. L'algoritmo da utilizzare per ordinare è il bubblesort.
7. Nel programma principale si stampi la lista *studenti* opportunamente formattata.

8. Nel programma principale si stampi la lista *studenti* ordinata per cognome (usare la funzione *bubblesort* implementata al punto 6) opportunamente formattata.

## 2 Osservazioni

- La creazione del file di testo *esiti.txt* è a cura dello studente.
- Usare la funzione `split()` per estrarre il cognome, il mese e il voto. Si testi in console il suo funzionamento. Ad esempio si testi in console: `"Bianchi 12 30".split()`
- La comparazione fra stringhe avviene nello stesso modo della comparazione fra numeri: `"Rossi" > "Bianchi"` restituisce *True*. Lo si testi eventualmente in console.
- Con *opportunamente formattata* si intende una stampa a video che consenta di visualizzare l'output in maniera ordinata e facilmente leggibile. Non utilizzare istruzioni del tipo `print(lista)` perchè non mostrano l'output ben formattato. Scrivere piuttosto una funzione che iteri sulla lista e ne stampi il contenuto elemento per elemento.