

Programmazione I

Prova di programmazione – 8 Febbraio 2019 – 2 ore

Scrivere un programma per la gestione della ricerca e del download di brani musicali da un catalogo virtuale online. Il sistema memorizza i brani disponibili in catalogo tramite un file *brani.txt* (fornito). Ogni riga di tale file corrisponde ad un brano ed è memorizzata nel seguente modo:

artista titolo anno genere downloads

dove **artista** e **titolo** sono delle stringhe (senza spazi) aventi lunghezza massima prefissata, **anno** indica la data di uscita, **downloads** indica il numero di volte che il brano è stato scaricato. Il **genere** è memorizzato sul file come un singolo carattere 'p' (pop), 'c' (classica), 'r' (rock), 'h' (hip-hop), 't' (trap). I brani disponibili in catalogo sono fisicamente ospitati in diversi *server*, uno per ogni genere musicale. Un secondo file di testo *servers.txt* (fornito) indica l'elenco dei server. Ogni riga del file descrive un server nel seguente formato:

nomeServer tipoServer

dove *nomeServer* è una stringa (senza spazi) di lunghezza prefissata e *tipoServer* indica il genere dei brani ospitati su quel server, ed usa la stessa codifica vista per il file dei brani (singolo carattere $\in \{ 'p', 'c', 'r', 'h', 't' \}$).

Si definisca un'opportuna struttura dati **Bran** che possa ospitare i dati relativi ad un brano memorizzati nel file *brani.txt* (si scelga il tipo di dato ritenuto più opportuno per ciascun campo). Si definisca anche una struttura dati **Server** per rappresentare i dati registrati nel file *server.txt*. Quest'ultima struttura deve anche contenere un campo intero **totDownloads** (che verrà utilizzato nell'esercizio 4).

Il programma deve fornire le seguenti funzionalità:

1. Lettura dei database di brani e di server (max 12 pt). Si scrivano due funzioni

Bran * leggiBrani (char fileName[], int &dim);

Server * leggiServer (char fileName[], int &dim);

che, ricevuto in ingresso il nome di un file (contenente i dati dei brani nel primo caso, e quelli dei server nel secondo), leggano i dati relativi e li memorizzino in un array (di elementi di tipo **Bran** nel primo caso, e di tipo **Server** nel secondo). Si noti che la prima riga di ognuno dei files forniti contiene un intero che indica quanti elementi vi sono contenuti (brani, server). Le funzioni devono allocare dinamicamente la memoria necessaria per gli array, che verranno ritornati dopo esser stati popolati. Le funzioni devono anche ritornare la dimensione logica degli array tramite il parametro formale **dim**.

2. Stampa di una lista di brani (max 6 pt). Si scriva una funzione

void stampaBrani (**Bran** b[], int dim, char filtro);

che, ricevuto in ingresso un array **b** di oggetti di tipo **Bran** e la sua dimensione **dim**, ne stampi a video la descrizione (usando formattazione opportuna). Il parametro **filtro** permette di stampare i soli elementi di **b** per cui il campo **genere** è uguale a **filtro**. Se **filtro** è uguale a '_' si stampino tutti gli elementi di **b**.

3. Trova brano per intervallo di date (max 6 pt). Si realizzi una funzione

Brano * cercaNelleDate (Brano b[], int dim, int y1, int y2, int &dimR);

che preso in ingresso un array **b** di tipo **Brano** e dimensione **dim**, e due valori interi **y1** e **y2** rappresentanti due anni (si assuma $y1 < y2$) crei e ritorni un secondo array di elementi di tipo **Brano** contenente la lista di brani usciti nell'arco di tempo tra le due date inserite (estremi inclusi). La funzione ritorni anche la dimensione logica del nuovo array tramite il parametro **dimR**.

4. Classifica dei brani per numero di download (max 9 pt). Il candidato realizzi una funzione

void classifica (Server s[], int dimS, Brano b[], int dimB);

che, presi in ingresso gli array dei server (**s**) e dei brani (**b**) con le relative dimensioni (**dimS**, **dimB**) popoli il campo **totDownloads** di ogni elemento dell'array **s** col valore cumulativo di tutti i download dei brani appartenenti al genere di quel server (si ricorda che ogni server può contenere solo brani di un dato genere. Ad esempio, se nell'array **b** fosse presente un primo brano di genere *classica* con 12 downloads e un secondo brano *classica* con 6 downloads, **totDownloads** per il server *classica* sarebbe pari a 18).

La funzione quindi riordini l'array *server* per numero di **totDownloads** decrescente (prima i server con maggior numero di download complessivi) e stampi a video, per ogni server, (a) il nome del server; (b) il numero totale di downloads; (c) l'elenco dei film in esso presenti. Per il punto (c), si sfrutti **stampaBrani**.

Realizzare nel main opportune chiamate per validare il corretto funzionamento di tutte le funzioni scritte (si veda il possibile esempio di output in appendice).

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'overflow e l'inserimento di dati in formato errato da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati
 - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale

Possibile esempio di output:

Esercizio 1-2)

```
-----  
Lista dei brani in catalogo..  
-----  
BRANO (0)      Queen   Bohemian_Rhapsody      (1975 - r).   Downloads = 60.  
BRANO (1)      Pollini Chopin_Nocturnes (2005 - c).   Downloads = 20.  
...  

```

Esercizio 3)

```
-----  
Lista dei brani usciti tra il 2000 e il 2018  
-----  
BRANO (0)      Pollini Chopin_Nocturnes (2005 - c).   Downloads = 20.  
BRANO (1)      Perahia Bach_Suites_Inglesesi (2012 - c).   Downloads = 5.  
...  

```

Esercizio 4)

```
-----  
SERVER 0  
-----  
Name: server.rock.  
Downloads totali: 108.  
BRANO (0)      Queen   Bohemian_Rhapsody      (1975 - r).   Downloads = 60.  
...  
  
-----  
SERVER 1  
-----  
Name: server.pop.  
Downloads totali: 64.  
BRANO (4)      Jackson The_Girl_Is_Mine (1983 - p).   Downloads = 43.  
...  

```