FONDAMENTI DI PROGRAMMAZIONE
CORSO DI LAUREA IN INGEGNERIA INFORMATICA

PROVA PRATICA 05 GIUGNO 2019

Una `SongPlaylist` rappresenta una lista di canzoni in riproduzione su un lettore multimediale. Ogni canzone è rappresentata dal titolo, l'album, l'artista, e la durata. Titolo, album e artista sono stringhe di al più 50 caratteri. La durata è misurata in secondi. Una delle canzoni è attualmente in esecuzione. Implementare le seguenti operazioni che possono essere effettuate su una `SongPlaylist`:

--- Metodi invocati nella PRIMA PARTE di main.cpp: ---

- ✓ **`SongPlaylist sp;`**

  Costruttore di default che inizializza una `SongPlaylist` vuota.
- ✓ **`sp.aggiungi(titolo, album, artista, sec);`**

  Funzione che aggiunge una canzone in fondo alla `SongPlaylist`, dove le stringhe `titolo`, `album` e `artista` rappresentano, rispettivamente, il titolo, l'album e l'artista. `sec` rappresenta la durata della canzone, in secondi. Se la canzone è la prima ad essere aggiunta alla playlist, entra subito in riproduzione dal secondo 0. Una stessa canzone può essere presente più volte nella playlist. Se titolo, album, o artista sono lunghi più di 50 caratteri, vengono inseriti troncati nella playlist. Se uno degli input non è valido, la funzione non fa niente.
- ✓ **`sp.play(n);`**

  Funzione che fa avanzare la riproduzione corrente di `n` secondi, dove `n` è un numero non negativo. Se la canzone corrente finisce, inizia la successiva. La playlist è da considerarsi ciclica, ovvero la canzone che va in esecuzione dopo l'ultima della playlist, è la prima della playlist. Se la playlist è vuota, la funzione non fa niente. Se l'input non è valido, la funzione non fa niente.
- ✓ **`cout << sp;`**

  Operatore di uscita per il tipo `SongPlaylist`. L'output è nel formato seguente:

```
Symphony of Destruction-Countdown to Extinction-Megadeth-320
Enter Sandman-Black Album-Metallica-337
>[2:10]Walk-Vulgar Display of Power-Pantera-315
Raining Blood-Reign in Blood-Slayer-257
```

  Dove titolo, album e artista sono separati dal carattere `'-'`. La durata è espressa in secondi. La canzone preceduta dal carattere `'>'` è quella attualmente in esecuzione, e tra parentesi quadre è espresso il secondo attualmente in riproduzione, sempre in formato *minuti*:*secondi*.

--- Metodi invocati nella SECONDA PARTE di main.cpp: ---

- ✓ **`sp.elimina(titolo, album, artista);`**

  Funzione che elimina dalla `SongPlaylist` la canzone avente titolo `titolo`, album `album` e artista `artista`. Se la canzone non è presente nella playlist, la funzione non fa niente. Se la canzone è presente più volte nella playlist, la funzione elimina solo la prima occorrenza. Se la canzone da eliminare è attualmente in riproduzione, entra in riproduzione la canzone successiva (se esiste) dal secondo 0. Se l'input non è valido, la funzione non fa niente.
- ✓ **`sp += sp2;`**

  Operatore di somma e assegnamento tra due `SongPlaylist`, che concatena tutte le canzoni di `sp2` alla fine di `sp1`.
- ✓ **`int(sp);`**

  Operatore di conversione a `int` su tipo `SongPlaylist`, che restituisce il numero di secondi riprodotti dall'inizio della playlist fino al secondo corrente di riproduzione. Per esempio, nella playlist mostrata sopra, l'operatore dovrà restituire `5:20+5:37+2:10 = 320 + 337 + 130 = 787`.
- ✓ **`~SongPlaylist();`**

  Distruttore.

Mediante il linguaggio C++, realizzare il tipo di dato astratto **`SongPlaylist`**, definito dalle precedenti specifiche. **Gestire le eventuali situazioni di errore**.

USCITA CHE DEVE PRODURRE IL PROGRAMMA

```
---PRIMA PARTE---
Test costruttore e operatore di uscita:

Test aggiungi():
>[0:0]titolo1-album1-artista1-300
titolo2-album2-artista2-240
titolo3-album3-artista3-301

Test play():
titolo1-album1-artista1-300
>[1:5]titolo2-album2-artista2-240
titolo3-album3-artista3-301

>[0:24]titolo1-album1-artista1-300
titolo2-album2-artista2-240
titolo3-album3-artista3-301

---SECONDA PARTE---
Test somma e assegnamento:
sp2=
>[0:0]titolo4-album4-artista4-400
titolo5-album5-artista5-200
sp=
>[0:24]titolo1-album1-artista1-300
titolo2-album2-artista2-240
titolo3-album3-artista3-301
titolo4-album4-artista4-400
titolo5-album5-artista5-200

Test int():
Sono stati riprodotti 24 secondi
Sono stati riprodotti 424 secondi
Sono stati riprodotti 483 secondi

Test elimina():
sp=
>[3:3]titolo2-album2-artista2-240
titolo3-album3-artista3-301
titolo5-album5-artista5-200

Test distruttore:
(distruttore chiamato)
```

Note per la consegna:

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In questo caso, i docenti procederanno alla valutazione dell'elaborato **solo se** lo studente avrà completato l'autocorrezione del proprio elaborato. In **tutti** gli altri casi (per esempio, il programma non compila, non collega, non esegue o la prima parte dell'output non coincide con quella attesa), l'elaborato è considerato **insufficiente** e, pertanto, **non verrà corretto**.