Esercizio – classe `Orario`

Implementare in C++ una classe di nome `Orario` che realizza il tipo di dato astratto *ora del giorno* così definito:

valori:
terne `<h,m,s>`, con `h`, `m`, `s` numeri interi e $0 \leq h \leq 23$ e $0 \leq m,s \leq 59$, che rappresentano, rispettivamente, le ore, i minuti e i secondi;

operazioni su orari:

- confronto tra due orari: operatore == (vero se i due orari sono uguali); operatore < (vero se il primo orario precede il secondo);
- lettura e scrittura (con overloading degli operatori << e >>) di un orario da/su stream nel formato `h:m:s`;
- "getter" per gli attributi `h`, `m`, `s`; funzioni `get_ora()`, `get_minuti()`, `get_secondi()`;
- funzione di conversione da orario `<h,m,s>` a equivalente numero di: funzione `to_second()`; ad esempio, se `a` contiene l'orario `<9:42:10>`, `a.to_seconds()` restituisce `34930`;
- somma tra un orario e un dato numero di secondi: operatore +; ad esempio, se `a` contiene l'orario `<9:42:10>`, `a + 5221` restituisce il nuovo orario `<11:9:11>`.

La classe fornisce anche un costruttore con tre parametri di tipo intero che rappresentano rispettivamente ore, minuti e secondi, e un costruttore con un parametro di tipo intero che rappresenta l'equivalente in secondi dell'orario.
N.B. La funzione di lettura e i costruttori devono (obbligatoriamente) controllare che l'orario fornito sia corretto; in caso contrario, viene stampato un opportuno messaggio di errore e l'orario viene forzato al valore `0:0:0`.

Note di implementazione.

- Definire una funzione propria privata `controlla_orario()` che controlla la correttezza dell'orario.

Main di prova.

La classe realizzata deve essere utilizzata dal main di prova messo a disposizione su 'elly' insieme al testo dell'esercizio (file `orario_main.cpp`). L'esecuzione del main, completato con la classe `Orario`, deve dare i risultati riportati di seguito al main stesso. Non è ammesso apportare modifiche al main.

N.B. E' vietato: usare dichiarazioni friend, usare goto, usare variabili globali non motivate, definire public tutti i campi di una classe, usare ereditarietà a sproposito.