

`VoliDelGiorno` rappresenta un elenco di voli in partenza da un dato aeroporto nell'arco di una giornata. Questo elenco può contenere un numero illimitato di voli. Ogni volo ha le seguenti caratteristiche:

- Un orario di partenza, che è una stringa nel formato `AB.CD` dove `A` può essere un numero da 0 a 2; `B` e `D` possono essere ciascuno un numero tra 0 e 9; infine `C` può essere un numero tra 0 e 5. Ad esempio, "23.59" è un orario valido, mentre "83.00" non lo è. (Nota: per ragioni di semplicità implementative, "29.59" viene considerato un orario valido, anche se non ha alcun senso nella realtà. L'utilizzatore della classe sarà invitato a non inserire orari non significativi).
- Una destinazione, che è una stringa non vuota di al più 20 caratteri. Questa stringa può essere composta esclusivamente dai 26 caratteri minuscoli dell'alfabeto inglese e da spazi.
- Un'indicazione se quel volo è stato annullato o meno.

Implementare le seguenti operazioni che possono essere effettuate su `VoliDelGiorno`:

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

✓ `VoliDelGiorno vg`;

Costruttore di default che inizializza un `VoliDelGiorno`, inizialmente vuoto.

✓ `cout << vg`;

Operatore di uscita per il tipo `VoliDelGiorno`, che stampa a schermo i voli nel seguente formato:

```
- Orario: 07.30
  Destinazione: milano malpensa
  Annullato: No
- Orario: 08.05
  Destinazione: roma ciampino
  Annullato: Si
```

Come mostrato nell'esempio di cui sopra, i voli devono essere stampati in ordine crescente di orario di partenza. In altre parole, il primo volo ad essere stampato è il primo volo programmato per il giorno; il secondo volo ad essere stampato è il secondo volo programmato per il giorno, e così via. Vengono stampati tutti i voli presenti, sia quelli confermati che quelli annullati, come si vede dall'esempio.

✓ `vg.aggiungi(orario, destinazione)`;

Metodo che aggiunge al `VoliDelGiorno vg` un nuovo viaggio caratterizzato da `orario` e `destinazione`, entrambe C-stringhe. Il formato dell'orario è il consueto `AB.CD`. Il viaggio è inizialmente confermato. Se un volo con stesso orario di partenza è già presente, o se `orario` o `destinazione` non hanno il formato corretto, il volo da aggiungere non viene aggiunto e `VoliDelGiorno vg` rimane invariato.

✓ `vg annulla(orario)`;

Metodo che annulla, se non era già stato annullato, il volo che parte a `orario`. Se il volo non è presente, `VoliDelGiorno vg` rimane invariato. Il volo annullato deve comunque rimanere in lista.

✓ `VoliDelGiorno vg2 = vg`;

Costruttore di copia che inizializza `VoliDelGiorno vg2` uguale a `vg`.

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

✓ `~VoliDelGiorno()` ;

Qualora sia necessario, implementare il distruttore.

✓ `vg.nonAnnullati()` ;

Metodo che restituisce un nuovo `VoliDelGiorno`, contenente solo i voli confermati presenti in `vg`.

✓ `~vg` ;

Operatore di complemento bit a bit su `VoliDelGiorno`, che modifica `vg` invertendo lo stato confermato/annullato per ciascun volo, e ne restituisce un riferimento. In altre parole, un volo confermato viene annullato e viceversa.

✓ `vg+vg2` ;

Operatore di somma tra due `VoliDelGiorno`. L'operatore somma deve restituire un nuovo `VoliDelGiorno` che contenga i voli presenti in entrambi `vg` e `vg2`. Nel nuovo `VoliDelGiorno`, i voli devono essere presenti in ordine crescente di orario di partenza. Se uno stesso orario di partenza è presente sia in `vg` che in `vg2`, il nuovo `VoliDelGiorno` restituito dovrà contenere solo il volo presente in `vg`.

Mediante il linguaggio C++, realizzare il tipo di dato astratto VoliDelGiorno , definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo <code>string</code> , il tipo <code>vector</code> , il tipo <code>list</code> , ecc. Gestire le eventuali situazioni di errore.
--

Uscita che deve produrre il programma

--- PRIMA PARTE ---

Test costruttore e aggiungi

- Orario: 07.30
Destinazione: milano malpensa
Annullato: No
- Orario: 08.05
Destinazione: roma ciampino
Annullato: No
- Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: No

Test annulla

- Orario: 07.30
Destinazione: milano malpensa
Annullato: Si
- Orario: 08.05
Destinazione: roma ciampino
Annullato: Si
- Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: No

Test costruttore di copia

- Orario: 07.30
Destinazione: milano malpensa
Annullato: Si
- Orario: 08.05
Destinazione: roma ciampino
Annullato: Si
- Orario: 09.45
Destinazione: torino caselle
Annullato: No
- Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: No

--- SECONDA PARTE ---

Test eventuale distruttore

- Distruttore chiamato
- Test nonAnnullati
- Orario: 09.45
Destinazione: torino caselle
Annullato: No
 - Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: No

Test operatore di complemento

- Orario: 07.30
Destinazione: milano malpensa
Annullato: No
- Orario: 08.05
Destinazione: roma ciampino
Annullato: No
- Orario: 09.45
Destinazione: torino caselle
Annullato: Si
- Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: Si

Test operatore di somma

- Orario: 07.30
Destinazione: milano malpensa
Annullato: No
- Orario: 08.05
Destinazione: roma ciampino
Annullato: No
- Orario: 09.45
Destinazione: torino caselle
Annullato: Si
- Orario: 10.20
Destinazione: firenze peretola
Annullato: No
- Orario: 11.35
Destinazione: pisa galileo galilei
Annullato: Si
- Orario: 13.10
Destinazione: catania fontanarossa
Annullato: No

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**.