

Uno `Schedario` può contenere documenti. Ogni documento è provvisto di un intero che ne definisce la tipologia. Nello `Schedario` i documenti sono organizzati su tre livelli di priorità, numerati da uno (livello più basso) a tre (livello più alto). Pertanto quando si vuole inserire un nuovo documento nello `Schedario`, oltre alla tipologia, occorre specificare anche il livello di priorità con cui si vuole inserire nello `schedario`.

Implementare le seguenti operazioni che possono essere compiute su di uno `Schedario`:

--- **PRIMA PARTE** --- *(qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova sarà considerata insufficiente e pertanto non sarà corretta)*

✓ **`Schedario s;`**

Costruttore che crea uno `Schedario` vuoto (non vi sono documenti, per nessun livello di priorità). Uno `schedario` vuoto viene mostrato a video nel seguente formato:

```
L1 () L2 () L3 ()
```

✓ **`s.aggiungi(liv,tip);`**

Operazione che aggiunge un documento di tipologia `tip` al livello di priorità `liv` nello `schedario s`, dove `liv` è compreso tra 1 e 3 e `tip` è un intero qualsiasi. L'inserimento avviene in modo tale che, qualora `s` venga stampato a video, il documento **inserito per ultimo venga mostrato a video per primo** tra quelli del livello di priorità `liv`.

✓ **`cout << s;`**

Operatore di uscita per il tipo `Schedario`. L'uscita ha il seguente formato:

```
L1 (6) L2 (35, 48, 21) L3 (10, 12)
```

L'output mostrato corrisponde a uno `Schedario` con un documento di tipologia 6 al livello di priorità più basso, tre documenti (di tipologia 35, 48 e 21) a livello 2 e due documenti (di tipologia 10 e 12) a livello di priorità massimo (l'ultimo documento inserito a livello 2 è di tipologia 35, l'ultimo a livello 3 è di tipologia 10).

✓ **`s-=liv;`**

Operazione che modifica lo `schedario s` eliminando tutti i documenti aventi priorità `liv`.

Esempio: Nel caso in cui `s` sia il seguente:

```
L1 (6) L2 (35, 48, 21) L3 (10, 12)
```

dopo l'esecuzione di `s-=2` l'uscita a video di `s` produce:

```
L1 (6) L2 () L3 (10, 12)
```

--- SECONDA PARTE ---

✓ `s.promuovi(liv,tip);`

Operazione di che sposta tutti i documenti di tipologia `tip` che si trovano al livello di priorità `liv` al livello immediatamente più alto (ovviamente questa operazione avrà un qualche effetto su `s` se e solo se `liv` è uno oppure due, altrimenti lo schedario viene lasciato inalterato).

Esempio: Nel caso in cui `s` si trovi in questo stato:

```
L1(6)L2(5,8,5,5,3,5)L3(10,12)
```

dopo l'operazione `s.promuovi(2,5)` l'uscita a video di `s` produce:

```
L1(6)L2(8,3)L3(5,5,5,5,10,12)
```

NB: nell'implementare questa funzione *si possono* utilizzare funzioni ausiliarie, ma si deve contestualmente cercare di **scorrere la struttura dati una sola volta**.

✓ `~Schedario();`

Distruttore.

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

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore e dell'operatore di uscita
L1()L2()L3()

Test della aggiungi
L1(6)L2(35,48,21)L3(10,12)

Test dell'operatore -=
L1(6)L2()L3(10,12)

--- SECONDA PARTE ---

Test della promuovi
[s prima della promuovi]
L1(6)L2(5,8,5,5,3,5)L3(10,12)
[s dopo la promuovi]
L1(6)L2(8,3)L3(5,5,5,5,10,12)

Test del distruttore (s e' stato appena distrutto)