

Un `Raccoglitore` è formato da righe di blocchi colorati, ed i possibili colori sono BIANCO, ROSSO, GIALLO e VERDE. Le righe possono avere larghezza diversa e debbono essere molto flessibili, nel senso che debbono permettere l'aggiunta o la rimozione di un blocco colorato, in qualunque momento. In particolare si deve poter aggiungere un numero arbitrario di nuovi blocchi a ciascuna riga.

Le operazioni che possono essere effettuate su un oggetto `Raccoglitore` sono le seguenti:

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

✓ **`Raccoglitore r;`**

Costruttore senza argomenti, che inizializza il `Raccoglitore r` dotandolo di una sola riga. Inizialmente tale riga non contiene blocchi.

✓ **`Raccoglitore r(n);`**

Costruttore con argomento intero, che inizializza la `Raccoglitore r` dotandolo di `n` righe. Inizialmente nessuna delle righe contiene blocchi.

✓ **`r.aggiungiBlocco(COLORE);`**

Operazione che aggiunge un nuovo blocco, di colore `COLORE`, **alla riga con meno blocchi al momento**. In caso ce ne siano più di una col minor numero di blocchi, lo aggiunge alla riga più in alto tra quelle a larghezza minima.

`COLORE` è una costante enumerata che può valere solamente uno dei valori sotto riportati:

```
enum Colore{BIANCO, ROSSO, GIALLO, VERDE};
```

Esempio: Qualora il `raccoglitore` si trovi in questo stato (si veda in proposito l'operatore di uscita),

`VR`

`VG`

`BB`

la chiamata alla `aggiungiBlocco(GIALLO)` dovrà portarlo in questo stato:

`VRG`

`VG`

`BB`

✓ **`cout << r;`**

Operatore di uscita per il tipo `Raccoglitore`. L'output deve avere la seguente formattazione:

`BVR`

`RG`

Nell'esempio di sopra, il `raccoglitore` è composto da due righe non vuote, di cui la prima composta da tre blocchi (di colore BIANCO, VERDE e ROSSO, rispettivamente) e la seconda da due blocchi, di colore ROSSO e GIALLO.

✓ `r.togliBlocco()` ;

Funzione che modifica il raccoglitore, togliendo il blocco più a destra presente **nella riga più lunga**. Nel caso in cui più righe abbiano la stessa larghezza massima, rimuove l'ultimo blocco della riga più in alto tra quelle aventi larghezza massima.

Esempio:

Dato il seguente raccoglitore,

```
V
VG
BB
```

La chiamata alla `togliBlocco` dovrà fornire il seguente raccoglitore:

```
V
V
BB
```

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

✓ `~Raccoglitore()` ;

Distruttore, *qualora necessario*.

✓ `Raccoglitore r2(r)` ;

Costruttore di copia.

✓ `r3 = r;`

Operatore di assegnamento.

Mediante il linguaggio C++ realizzare il tipo di dato astratto **Raccoglitore** definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL (Standard Template Library) come il tipo `std::string`, il tipo `std::vector`, il tipo `std::list`, ecc. **Gestire le eventuali situazioni di errore.**

USCITA CHE DEVE PRODURRE IL PROGRAMMA

---PRIMA PARTE---

Test del costruttore e della aggiungiBlocco()

VR
VG
B

Altri test della aggiungiBlocco() e op. di uscita

VRV
VG
BB

Primo test della toglBlocco()

VR
VG
BB

Secondo test della toglBlocco()

V
VG
BB

Terzo test della toglBlocco()

V
V
BB

---SECONDA PARTE---

Test del costruttore di copia

V
V
BB

Test dell'operatore di assegnamento

V
V
BB

Test del distruttore (se ridefinito)

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**.