

Una *Rastrelliera* raccoglie i dischi da caricare sui bilancieri di una palestra. Ci sono quattro tipi di dischi: i dischi gialli da un chilo, i dischi verdi da due chili, i dischi rossi da tre chili ed i dischi neri da quattro chili.

Implementare le seguenti operazioni che possono essere effettuate su una *Rastrelliera*:

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

✓ **`Rastrelliera r(ng, nv, nr, nn);`**

Costruttore che inizializza una *Rastrelliera* con *ng* dischi gialli, *nv* dischi verdi, *nr* dischi rossi e *nn* dischi neri. Per default, il numero di ciascun tipo di disco è 10. Se uno dei parametri non è valido, il corrispondente numero di dischi è 10.

✓ **`b = r.carica(ng, nv, nr, nn);`**

Crea un bilanciere *b* e lo carica con *ng* dischi gialli, *nv* dischi verdi, *nr* dischi rossi e *nn* dischi neri prelevati dalla *Rastrelliera* *r*. La funzione ritorna NULL se uno dei parametri non è valido, o se nella *Rastrelliera* non ci sono dischi sufficienti. In tali casi la *Rastrelliera* rimane invariata.

NOTA: Il formato del bilanciere *b* può essere deciso dal candidato, a patto che sia di tipo `int*`.

✓ **`r.scarica(b);`**

Scarica e distrugge il bilanciere *b*, riponendo i suoi dischi nella *Rastrelliera* *r*.

✓ **`cout << r;`**

Operatore di uscita. L'uscita ha il seguente formato:

```
GIALLO <6>      [] [] [] [] [] []
VERDE   <6>      [] [] [] [] [] []
ROSSO   <8>      [] [] [] [] [] [] [] []
NERO    <10>     [] [] [] [] [] [] [] [] [] []
```

L'output mostrato corrisponde a una *Rastrelliera* avente 6 dischi gialli, 6 verdi, 8 rossi e 10 neri.

✓ **`Rastrelliera::calcolaPeso(b);`**

Calcola il peso caricato sul bilanciere *b*.

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

✓ **`r2 = r1;`**

Operatore di assegnamento che assegna il valore della *Rastrelliera* *r1* alla *Rastrelliera* *r2*.

✓ **`b = Rastrelliera::unisci(b1, b2);`**

Unisce il bilanciere *b1* con il bilanciere *b2* e li distrugge. Crea il bilanciere *b* risultante e lo restituisce.

Mediante il linguaggio C++, realizzare il tipo di dato ***Rastrelliera***, definito dalle precedenti specifiche.

Gestire le eventuali situazioni di errore.

OUTPUT ATTESO DAL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore:

```
GIALLO <10>  [] [] [] [] [] [] [] [] [] []
VERDE  <10>  [] [] [] [] [] [] [] [] [] []
ROSSO  <10>  [] [] [] [] [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

Test di carica e di calcolaPeso:

il peso del bilanciere b1 e': 21

```
GIALLO <6>   [] [] [] [] [] []
VERDE  <6>   [] [] [] [] [] []
ROSSO  <7>   [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

Altro test di carica e di calcolaPeso:

il bilanciere b2 e' scarico

```
GIALLO <6>   [] [] [] [] [] []
VERDE  <6>   [] [] [] [] [] []
ROSSO  <7>   [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

Test di scarica:

```
GIALLO <10>  [] [] [] [] [] [] [] [] [] []
VERDE  <10>  [] [] [] [] [] [] [] [] [] []
ROSSO  <10>  [] [] [] [] [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

--- SECONDA PARTE ---

Test di unisci:

il peso del bilanciere b5 e': 18

```
GIALLO <5>   [] [] [] [] []
VERDE  <5>   [] [] [] [] []
ROSSO  <9>   [] [] [] [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

Test dell'op. di assegnamento:

```
GIALLO <5>   [] [] [] [] []
VERDE  <5>   [] [] [] [] []
ROSSO  <9>   [] [] [] [] [] [] [] [] []
NERO   <10>  [] [] [] [] [] [] [] [] [] []
```

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**.