

Una `MatQuad` è una matrice quadrata di interi, di dimensioni qualsiasi.

Implementare le seguenti operazioni che possono essere compiute su di una `MatQuad`:

--- **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**)

✓ **`MatQuad m(size) ;`**

Costruttore che crea una `MatQuad` di dimensioni `size` per `size`, con tutti gli elementi inizialmente uguali a zero.

✓ **`m.aggiorna(vett, dim) ;`**

Operazione che aggiorna il contenuto di `m`, con i valori contenuti nel vettore `vett` (di dimensioni `dim`). `m` viene aggiornata per righe, finché ci sono elementi in `vett`. Se `vett` non ha elementi sufficienti ad aggiornare tutti gli elementi di `m`, gli elementi di `m` che non possono essere aggiornati conservano il loro precedente valore. Se `vett` ha più elementi del necessario, quelli in eccesso vengono ignorati.

Esempio: Nel caso in cui `m` abbia ancora tutti gli elementi uguali a zero, dopo la aggiorna con il seguente vettore di sette elementi `{5, 5, 5, 6, 4, 4, 3}`, `m` varrà:

```
5 5 5 6
4 4 3 0
0 0 0 0
0 0 0 0
```

✓ **`cout << m;`**

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

```
5 5 5 6
4 4 3 0
0 0 0 0
0 0 0 0
```

L'output mostrato corrisponde a una `MatQuad` 4x4.

✓ **`m.trova() ;`**

Operazione che restituisce `true` se `m` contiene **almeno una riga con almeno tre elementi consecutivi tutti uguali fra loro** e restituisce `false` in caso contrario.

Esempio: Nel caso in cui `m` sia la seguente, la `m.trova()` restituirà `true`:

```
5 5 5 6
4 4 3 0
0 0 0 0
0 0 0 0
```

--- **SECONDA PARTE** ---

✓ **`m.raddoppia() ;`**

Operazione che raddoppia le dimensioni della matrice. Inoltre, il vecchio contenuto viene copiato nel quadrante Q2 e Q3 della nuova matrice, mentre nel quadrante Q1 e Q4 finiscono tutti 8 e 9, rispettivamente.

Esempio: Nel caso in cui m sia la seguente:

```
5 5 5 6
4 4 3 0
0 0 0 0
0 0 0 0
```

dopo l'operazione $m.raddoppia()$ l'uscita a video di m produrrà (a destra vengono mostrati Q1-Q4):

```
8 8 8 8 5 5 5 6
8 8 8 8 4 4 3 0
8 8 8 8 0 0 0 0
8 8 8 8 0 0 0 0
5 5 5 6 9 9 9 9
4 4 3 0 9 9 9 9
0 0 0 0 9 9 9 9
0 0 0 0 9 9 9 9
```

Q1	Q2
Q3	Q4

✓ `~MatQuad()` ;

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto **MatQuad**, 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

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

Test della 'aggiorna'

```
5 5 5 6
4 4 3 0
0 0 0 0
0 0 0 0
```

Test della 'trova':

trovata una riga con almeno tre valori consecutivi uguali fra loro

--- SECONDA PARTE ---

Test della 'raddoppia'

```
8 8 8 8 5 5 5 6
8 8 8 8 4 4 3 0
8 8 8 8 0 0 0 0
8 8 8 8 0 0 0 0
5 5 5 6 9 9 9 9
4 4 3 0 9 9 9 9
0 0 0 0 9 9 9 9
0 0 0 0 9 9 9 9
```

Test del distruttore ($m2$ e' stata appena distrutta)