Un `Museo` è formato da `n` sale di quadri. Ogni sala può mantenere un numero massimo `q` di quadri. Le sale sono numerate a partire da 1. Un quadro è identificato univocamente dall'autore e dal titolo (non ci possono nel museo due quadri dello stesso autore con lo stesso titolo). L'autore e il titolo sono stringhe di al più 50 caratteri. Le operazioni che possono essere effettuate su un `Museo` sono le seguenti:

- **`InizializzaMuseo(m, n, q);`**

Costruttore che inizializza un museo **`m`** con **`n`** sale di massimo **`q`** quadri. Tutte le sale sono vuote.

- **`aggiungiOpera (m, autore, titolo)`**

Funzione che aggiunge il quadro con autore **`autore`** e titolo **`titolo`** alla sala con minor numero di quadri del museo **`m`**. Se ci sono più sale con lo stesso numero minimo di quadri, la sala scelta è quella di numero minore. In caso di errore, la funzione lascia il museo inalterato. La funzione restituisce `true` se l'operazione ha successo, `false` altrimenti.

- **`eliminaOpera(m, autore, titolo)`**

Funzione che elimina l'opera con autore **`autore`** e titolo **`titolo`** dal museo **`m`**. La funzione restituisce `true` se l'operazione ha successo, `false` altrimenti.

- **`stampaMuseo(m)`**

Funzione che stampa a video le opere del museo **`m`**. L'uscita ha il seguente formato: numero della sala racchiuso fra parentesi angolate seguito, per ogni opera della sala, dalle informazioni `autore` e `titolo` racchiuse fra parentesi tonde. Se non ci sono quadri nella sala viene stampato il carattere '-'. Le sale vengono stampate su righe diverse. L'uscita seguente corrisponde ad un museo con 3 sale, dove la prima sala contiene due quadri, la seconda sala e la terza sala non contengono quadri.

<1> (Sandro Botticelli, Nascita di Venere) (Leonardo da Vinci, Annunciazione)
<2> -
<3> -

- **`opereAutore(m, autore, s)`**

Funzione che restituisce il numero di quadri dell'autore **`autore`** presenti nella sala **`s`** del museo **`m`**.

Mediante il Linguaggio C++, realizzare il tipo `Museo` definito dalle precedenti specifiche utilizzando le strutture. Implementare il tipo di dato con una matrice. Gestire le eventuali situazioni di errore.

Continua sul retro ……

Esempio di funzione main()

```
// Il contenuto dello stream d'output standard è rappresentato racchiuso tra i simboli di
// commento /**/

int main(){
  Museo M;
  inizializzaMuseo (M, 3, 5);
  stampaMuseo(M);
                          /*
                             <1> -
                             <2> -
                             <3> -
                          */

aggiungiOpera (M, "Sandro Botticelli", "Nascita di Venere");
aggiungiOpera (M, "Leonardo da Vinci", "Annunciazione");
stampaMuseo(M);

/*
   <1> (Sandro Botticelli, Nascita di Venere)
   <2> (Leonardo da Vinci, Annunciazione)
   <3> -
*/

aggiungiOpera (M, "Michelangelo Merisi", "Bacco");
aggiungiOpera (M, "Sandro Botticelli", "La Primavera");
stampaMuseo(M);

/*
<1> (Sandro Botticelli, Nascita di Venere) (Sandro Botticelli, La Primavera)
<2> (Leonardo da Vinci, Annunciazione)
<3> (Michelangelo Merisi, Bacco)
 */

cout << opereAutore(M, "Sandro Botticelli", 1) << endl ;
                                                          /*  2 */

eliminaOpera (M, "Sandro Botticelli", "Nascita di Venere");
eliminaOpera (M, "Leonardo da Vinci", "Annunciazione");

cout << opereAutore(M, "Sandro Botticelli", 1) << endl;
                                                          /*  1 */
stampaMuseo (M);
/*
<1> (Sandro Botticelli, La Primavera)
<2> -
<3> (Michelangelo Merisi, Bacco)
*/
return 0;
}
```