

Un Anfiteatro è formato da colonne di marmo **numerate a partire da 1**. Ogni colonna è formata da mattonelle, che ne determinano l'altezza. Le colonne possono avere altezza diversa. Implementare le seguenti operazioni che possono essere effettuate su un Anfiteatro:

--- **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**)

✓

Anfiteatro a(n)

Costruttore che inizializza un Anfiteatro a formato da n colonne. Inizialmente, tali colonne non contengono mattonelle.

✓

a.aggiungiMattonelle(k)

Operazione che aggiunge k mattonelle alla colonna con nessuna mattonella dell' Anfiteatro a. Qualora ce ne sia più d'una, le aggiunge alla prima di esse.

✓

a.aggiungiColonna(k)

Operazione che aggiunge una colonna in fondo all' Anfiteatro a. Tale colonna contiene k mattonelle.

✓

cout << a

Operatore di uscita per il tipo Anfiteatro. L'uscita ha la forma: < [7], [12], [0], [0]>. Ciascuna coppia di parentesi quadre racchiude il numero di mattonelle di una colonna. In questo esempio, l'anfiteatro ha 4 colonne, la prima colonna ha 7 mattonelle, la seconda colonna ne ha 12 e le restanti due nessuna mattonella.

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

✓

a.togliColonna(j)

Operazione che toglie la colonna j dell' Anfiteatro a.

✓

a1 = a

Operatore di assegnamento, che sostituisce il valore dell' Anfiteatro risultato a1 con quello del Anfiteatro a.

✓

~Anfiteatro()

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dati astratti Anfiteatro, definito dalle precedenti specifiche. **Individuare le eventuali situazioni di errore, e metterne in opera un corretto trattamento.**

USCITA PRODOTTA DAL PROGRAMMA COL MAIN SEGUENTE

--- PRIMA PARTE ---

```
Chiamata al costruttore
<[0],[0],[0],[0]>
Chiamata alla funzione aggiungiMattonelle()
<[7],[0],[0],[0]>
Altra chiamata alla funzione aggiungiMattonelle()
<[7],[12],[0],[0]>
```

```
Chiamata alla funzione aggiungiColonna()
<[7],[12],[0],[0],[8]>
```

--- SECONDA PARTE ---

```
Chiamata alla funzione togliaColonna()
<[12],[0],[0],[8]>
```

```
Test del costruttore di copia
<[12],[0],[0],[8]>
```

```
Test dell'operatore assegnamento
a1 ( *prima* dell'assegnamento ) <[0],[0]>
a1 ( *dopo* l'assegnamento ) <[12],[0],[0],[8]>
```

```
Test del distruttore (a1 sta per essere distrutto)
```

```
-----MAIN-----
#include "compito.h"
#include <iostream>
using namespace std;
int main()
{
    cout << endl << "--- PRIMA PARTE ---" << endl;
    cout<<"\nChiamata al costruttore"<<endl;
    Anfiteatro a(4);
    cout<<a<<endl;
    cout << "Chiamata alla funzione aggiungiMattonelle()" << endl;
    a.aggiungiMattonelle(7);
    cout << a << endl;
    cout << "Altra chiamata alla funzione aggiungiMattonelle()" << endl;
    a.aggiungiMattonelle(12);
    cout << a << endl;
    cout << "\nChiamata alla funzione aggiungiColonna()" << endl;
    a.aggiungiColonna(8);
    cout << a << endl;
    cout << endl << "--- SECONDA PARTE ---" << endl;
    cout << endl << "Chiamata alla funzione togliaColonna()" << endl;
    a.togliaColonna(1);
    cout << a << endl;
    cout<<endl<<"Test del costruttore di copia"<<endl;
    Anfiteatro aCopy(a);
    cout<<aCopy<<endl;
    {
        cout << endl << "Test dell'operatore assegnamento " << endl;
        Anfiteatro a1(2);
        cout << "a1 ( *prima* dell'assegnamento ) " << a1 << endl;
        a1 = a;
        cout << "a1 ( *dopo* l'assegnamento ) " << a1 << endl;
        cout << endl;
        cout << "Test del distruttore (a1 sta per essere distrutto)" << endl;
    }
    return 0;
}
```