

|              |                                     |
|--------------|-------------------------------------|
| Started on   | Thursday, 9 February 2023, 11:01 AM |
| State        | Finished                            |
| Completed on | Thursday, 9 February 2023, 12:01 PM |
| Time taken   | 59 mins 59 secs                     |
| Marks        | 350.00/350.00                       |
| Grade        | 100.00 out of 100.00                |

Question **1**  
Correct  
Mark 100.00 out of 100.00

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |



Buatlah implementasi kelas **BunchOfKeys** dalam bahasa pemrograman C++ yang memiliki sebuah atribut **n\_keys** bertipe integer yang menyimpan jumlah kunci yang ada. Kelas juga **hanya** memiliki 2 buah *member function*, yaitu:

- 1. **add**: menambah kunci (hanya melakukan increment pada atribut **n\_keys**)
- 2. **shake**: mengeluarkan bunyi "kringing" sejumlah **n\_keys** kali (*setiap "kringing" diakhiri end-of-line*) apabila jumlah kunci > 1, atau mengeluarkan pesan "Tidak terjadi apa-apa" (*diakhiri end-of-line*) apabila jumlah kunci ≤ 1.

Pada saat konstruksi obyek, jumlah kunci semula adalah 0.

Diberikan header C++ sebagai berikut, Anda hanya diminta mengumpulkan file implementasi (**BunchOfKeys.cpp**) nya saja.

```
// BunchOfKeys.hpp
#ifndef __BUNCH_OF_KEYS_HPP__
#define __BUNCH_OF_KEYS_HPP__

#include <iostream>
using namespace std;

class BunchOfKeys {
public:
    // ctor
    BunchOfKeys();

    // member function
    void add();
    void shake();

private:
    int n_keys; // jumlah kunci yg ada
};

#endif
```

C++14

 [BunchOfKeys.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description       |
|----|-------|----------|-------------------|
| 1  | 25    | Accepted | 0.00 sec, 2.85 MB |
| 2  | 25    | Accepted | 0.00 sec, 2.85 MB |
| 3  | 25    | Accepted | 0.00 sec, 2.88 MB |
| 4  | 25    | Accepted | 0.00 sec, 2.88 MB |

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

Diberikan file berikut yang berisi sebuah kelas bernama **Paper**. Kelas **Paper** menyimpan dua buah atribut integer dan menghasilkan keluaran khusus setiap dilakukan pemanggilan *constructor*, *copy constructor*, *copy assignment operator*, dan *destructor*.

1. [Paper.hpp](#)
2. [Paper.cpp](#)

Perhatikan contoh program berikut:

```
#include "Paper.hpp"
int main() {
    Paper a('A');
    a.fold();
    a.fold();
    return 0;
}
```

Program di atas jika dijalankan dengan kelas Paper yang disediakan, akan mengeluarkan output:

```
ctor A
fold A(1)
fold A(2)
dtor A
```

Sebagai seorang mahasiswa yang telah memahami konsep dasar OOP dengan baik, anda ditugaskan untuk membuat **program utama** dengan memanfaatkan kelas **Paper** yang mengeluarkan output sebagai berikut:

```
ctor A
ctor B
ctor C
cctor C
fold A(1)
fold B(1)
fold C(1)
glue C
fold C(2)
setName C => X
fold C(1)
glue C
fold C(2)
dtor C
dtor X
dtor B
dtor A
```

Unggahlah program utama tersebut dengan nama file **main.cpp**

C++14

 [main.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description       |
|----|-------|----------|-------------------|
| 1  | 100   | Accepted | 0.03 sec, 2.78 MB |

Question **3**

Correct

Mark 150.00 out of 150.00

Time limit

1 s

Memory limit

64 MB

**Bilangan kompleks** dalam matematika, adalah bilangan yang dinotasikan oleh  $a + bi$ , di mana  $a$  dan  $b$  adalah bilangan riil, dan  $i$  adalah suatu bilangan imajiner di mana  $i^2 = -1$ . Bilangan riil  $a$  disebut juga *bagian riil* dari bilangan kompleks, dan bilangan real  $b$  disebut *bagian imajiner*. Misalkan sebuah bilangan kompleks  $3 + 5i$ , maka bilangan kompleks tersebut memiliki bagian riil yang bernilai 3 dan bagian imajiner yang bernilai 5.

Diberikan sebuah header kelas Kompleks sebagai berikut

```
#ifndef KOMPLEKS_H
#define KOMPLEKS_H

class Kompleks {
public:
    // ctor tanpa parameter
    // inisialisasi seluruh koefisien dengan nilai 0
    Kompleks();

    // ctor dengan parameter
    Kompleks(int real, int imajiner);

    //mengembalikan bagian riil
    int GetReal() const;

    // mengembalikan bagian imajiner
    int GetImajiner() const;

    // mengisi bagian riil
    void SetReal(int);

    // mengisi bagian imajiner
    void SetImajiner(int);

    // operator overloading

    // operator+ untuk melakukan penjumlahan dengan rumus berikut
    // (a + bi) + (c + di) = (a+c) + (b+d)i
    friend Kompleks operator+ (const Kompleks&, const Kompleks&);

    // operator- untuk melakukan pengurangan dengan rumus berikut
    // (a + bi) - (c + di) = (a-c) + (b-d)i
    friend Kompleks operator- (const Kompleks&, const Kompleks&);

    // operator* untuk melakukan perkalian dengan rumus berikut
    // (a + bi)(c + di) = ac + bci + adi + bd i^2 = (ac-bd) + (bc+ad)i
    friend Kompleks operator* (const Kompleks&, const Kompleks&);

    // operator* untuk mengkalikan bilangan kompleks dengan konstanta
    // (a + bi)(c) = (ac) + (bc)i
    friend Kompleks operator* (const Kompleks&, const int);

    // operator* untuk mengkalikan bilangan kompleks dengan konstanta (sifat komutatif)
    friend Kompleks operator* (const int, const Kompleks&);

    // mengembalikan jumlah instance yang pernah dibuat
    static int CountKompleksInstance();

    // mencetak bilangan kompleks ke layar, diakhiri dengan end-of-line
    // contoh:
    // 3+5i
    // 0+0i
    // -5-4i
    void Print();

private:
    static int n_kompleks;
    int real;
    int imajiner;
};

#endif
```

Implementasikan kelas tersebut, lalu kumpulkan **Kompleks.cpp** nya saja.

C++14

 [Kompleks.cpp](#)

Score: 150

Blackbox

Score: 150

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description       |
|----|-------|----------|-------------------|
| 1  | 10    | Accepted | 0.00 sec, 2.98 MB |
| 2  | 10    | Accepted | 0.00 sec, 2.87 MB |
| 3  | 10    | Accepted | 0.00 sec, 2.87 MB |
| 4  | 10    | Accepted | 0.00 sec, 2.85 MB |
| 5  | 10    | Accepted | 0.00 sec, 2.84 MB |
| 6  | 10    | Accepted | 0.00 sec, 2.91 MB |
| 7  | 10    | Accepted | 0.00 sec, 2.89 MB |
| 8  | 10    | Accepted | 0.00 sec, 2.96 MB |
| 9  | 10    | Accepted | 0.00 sec, 2.89 MB |
| 10 | 10    | Accepted | 0.00 sec, 2.87 MB |
| 11 | 10    | Accepted | 0.00 sec, 2.91 MB |
| 12 | 10    | Accepted | 0.01 sec, 2.93 MB |
| 13 | 10    | Accepted | 0.00 sec, 2.89 MB |
| 14 | 10    | Accepted | 0.00 sec, 2.98 MB |
| 15 | 10    | Accepted | 0.00 sec, 2.93 MB |