<u>Dashboard</u> / My courses / <u>ITB IF2210 2 2223</u> / <u>Praktikum 1</u> / <u>Latihan Praktikum 1</u>

Question **1**Tries remaining: 10
Marked 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 "krincing" sejumlah **n_keys** kali *(setiap "krincing" 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.

C + + 14

Maximum size for new files: 512MB, maximum attachments: 1

```
Files

You can drag and drop files here to add them.
```

Run

Check

Question **2**Tries remaining: 10
Marked out of

100.00

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
```

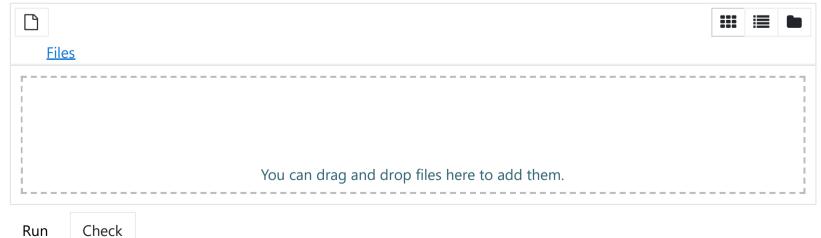
Sebagai seorang mahasiswa yang telah memahami konsep dasar OOP dengan baik, anda ditugaskan untuk membuat **program utama** dengan memanfaatkan kelas **Paper** yang mengeluarkan output sebagai berikut:

```
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

Maximum size for new files: 512MB, maximum attachments: 1



Question **3**Tries remaining: 10
Marked 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 a disebut bagian imajiner. Misalkan sebuah bilangan kompleks a + a tersebut memiliki bagian riil yang bernilai 3 dan bagian imaginer yang bernilai 5.

Diberikan sebuah header kelas Kompleks sebagai berikut

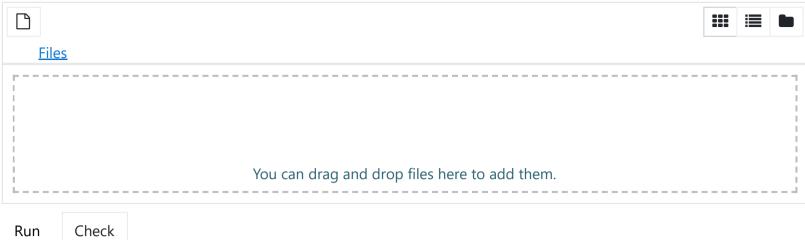
```
#ifndef KOMPLEKS_H
#define KOMPLEKS_H
class Kompleks {
 public:
   // ctor tanpa parameter
   // inisialisasi seluruh koefisien dengan nilai 0
   Kompleks();
   // ctor dengan parameter
   Kompleks(int real, int imaginer);
   //mengembalikan bagian riil
   int GetReal() const;
   // mengembalikan bagian imaginer
   int GetImaginer() const;
   // mengisi bagian riil
   void SetReal(int);
   // mengisi bagian imaginer
   void SetImaginer(int);
   // operator overloading
   // operator+ untuk melakukan penjumlahan dengan rumus berikut
   // (a + bi) + (c + di) = (a+c) + (b+d)i
   friend Kompleks operator+ (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&);
   // 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 imaginer;
#endif
```

2/10/23, 3:27 PM Latihan Praktikum 1

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

C++14

Maximum size for new files: 512MB, maximum attachments: 1



Question **4**Tries remaining:
10
Marked out of

115.00

Time limit	1 s
Memory limit	64 MB

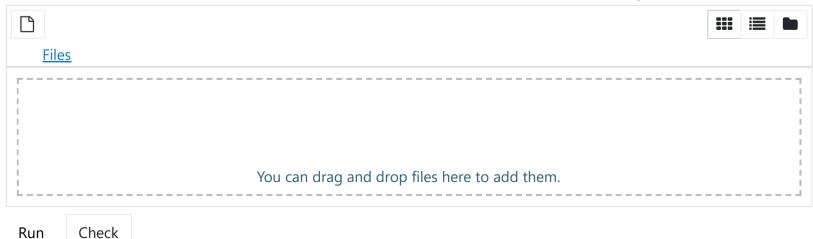
Kali ini Anda diminta untuk mengimplementasikan kelas **Polinom** yang memanfaatkan struktur data *array* yang ukurannya dialokasikan secara dinamis.

```
// Polinom.hpp
#ifndef POLINOM_HPP
#define POLINOM_HPP
class Polinom {
 public:
    // ctor, cctor, dtor, op=
    // untuk konstruktor, inisialisasi seluruh nilai koefisien dengan 0.
   Polinom(); // ctor Polinom dengan orde = 0
    Polinom(int); // ctor Polinom dengan orde = n (sesuai parameter)
    Polinom(const Polinom&);
   ~Polinom();
    Polinom& operator=(const Polinom&);
    // getter, setter
    int getKoefAt(int idx) const;
    int getDerajat() const;
    void setKoefAt(int idx, int val);
    void setDerajat(int);
    // member function
    // Melakukan pembacaan koefisien sejumlah derajat Polinom, dimulai dari x^0 (konstanta)
    void input();
    // Mencetak seluruh koefisien polinom. Untuk setiap koefisien akhiri dengan end-of-line
    // Cetaklah apa adanya dari koefisien ke-0 hingga derajat tertinggi (termasuk apabila koefisien = 0)
    void printKoef();
    // Menghitung hasil substitusi x dengan sebuah bilangan ke dalam polinom
    int substitute(int);
    // Mencetak polinom dengan format: A+Bx^1+Cx^2+Dx^3...dst (diakhiri dengan end-of-line)
    // Apabila suatu koefisien bernilai < 0, gunakan tanda "-" untuk menggantikan tanda "+"
   // Apabila suatu koefisien bernilai 0, lewati koefisien tersebut dan lanjutkan ke koefisien selanjutnya
   // Jika seluruh koefisien bernilai 0, keluarkan "0"
    void print();
 private:
   int * koef;
    int derajat; // derajat tertinggi
#endif
```

Anda hanya perlu mengumpulkan file Polinom.cpp nya saja.

C++14

Maximum size for new files: 512MB, maximum attachments: 1



2/10/23, 3:27 PM Latihan Praktikum 1

→ Praktikum 1

Jump to...