

Started on Thursday, 10 February 2022, 11:00 AM

State Finished

Completed on Thursday, 10 February 2022, 12:16 PM

Time taken 1 hour 16 mins

Marks 400.00/400.00

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

Diberikan header sebuah kelas **Weed**, yaitu semak-semak yang memiliki jumlah kawan semak (**many**).

[Weed.hpp](#)

Buatlah implementasi header tersebut dan kumpulkan sebagai **Weed.cpp**.

C++14

 [Weed.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 2.87 MB
2	20	Accepted	0.00 sec, 2.85 MB
3	20	Accepted	0.00 sec, 2.93 MB
4	20	Accepted	0.00 sec, 2.86 MB
5	20	Accepted	0.00 sec, 2.83 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Berikut adalah deklarasi dan definisi dari kelas **Card**. Kelas **Card** menghasilkan keluaran khusus jika constructor, copy constructor, copy assignment operator, destructor, dan fungsi gesek dipanggil.

[Card.hpp](#)

[Card.cpp](#)

Berikut adalah contoh program yang menggunakan kelas **Card** dan keluarannya:

<pre>int main() { Card bca (1); bca.gesek(); return 0; }</pre>	<pre>new empty card 1 chaching 1 destroy card 1</pre>
--	---

Tugas anda: Unggah file **main.cpp** yang berisi main program, yang menggunakan kelas Card untuk menghasilkan keluaran berikut:

```
new empty card 1  
new empty card with default number 0  
new empty card 2  
assign card 0 <- 1  
chaching 1  
copy card 2  
chaching 2  
destroy card 1  
destroy card 2  
destroy card 2  
destroy card 1
```

C++14

 [main.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	100	Accepted	0.00 sec, 2.95 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Kelas **SpacingGuildShip** merupakan pesawat luar angkasa dengan kapasitas penumpang **maxPassengerCap** dan navigator bernama Guild Navigator sebanyak **guildNavigatorCount**. Kelas **SpacingGuildShip** menyimpan berapa banyak pesawat yang telah diproduksi (**producedShips**) dan catatan total jarak yang ditempuh oleh semua pesawat (**totalGuildTravel**). Pesawat menampung penumpang sebanyak **passengerCount** dan memiliki nomor seri yang unik **serialNum**.

SerialNum didapat dari jumlah pesawat yang diproduksi (**producedShips** + 1) Implementasi kelas **SpacingGuildShip** dalam bentuk **SpacingGuildShip.cpp** dari definisinya di **SpacingGuildShip.hpp**

[SpacingGuildShip.hpp](#)

Kumpulkan **SpacingGuildShip.cpp**

C++14

 [SpacingGuildShip.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	16	Accepted	0.00 sec, 2.93 MB
2	16	Accepted	0.00 sec, 2.97 MB
3	16	Accepted	0.00 sec, 2.87 MB
4	16	Accepted	0.00 sec, 3.10 MB
5	16	Accepted	0.00 sec, 2.93 MB
6	20	Accepted	0.00 sec, 2.74 MB

Question 4

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Kelas **BigNumber** merupakan kelas yang digunakan untuk melakukan komputasi pada angka yang sangat besar, biasanya digunakan ketika tidak muat dalam tipe data int atau long int (misal 10^{10000}). Cara kerja kelas ini adalah menyimpan setiap digit angka dalam sebuah array. Dalam contoh kelas ini, banyak digit maksimal yang dapat disimpan adalah 20.

Berikut adalah contoh penyimpanan bilangan desimal dalam BigNumber.

Angka	BigNumber
123	[3, 2, 1, 0, 0, 0, 0, ...]
2210	[0, 1, 2, 2, 0, 0, 0, ...]

Operator yang perlu diimplementasikan adalah `==`, `<`, `>`, dan `+`.

- Operasi perbandingan (`==`, `<`, `>`) diimplementasikan dengan membandingkan digit di posisi yang sama:

Contoh perbandingan `123 < 2210`:

Angka	BigNumber
123	[3, 2, 1, 0 , 0, 0, 0, ...]
2210	[0, 1, 2, 2 , 0, 0, 0, ...]

Lakukan perulangan dari digit ke-20, bandingkan nilainya apakah lebih besar, lebih kecil, atau sama. Jika sama, maka lanjutkan ke digit selanjutnya (digit 19), dan seterusnya. Pada tabel diatas, kita sedang membandingkan digit 0 dan 2, karena `0 < 2`, maka `123 < 2210`.

- Operasi penambahan (`+`), yaitu penjumlahan setiap digit dari 2 BigNumber

Contoh penjumlahan `123 + 2210`:

Angka	BigNumber
123	[3, 2, 1, 0, 0, 0, 0, ...]
2210	[0, 1, 2, 2, 0, 0, 0, ...]
<code>123 + 2210 = 2333</code>	[3, 3, 3, 2, 0, 0, 0, ...]

BigNumber juga punya aksesori `[]` untuk mengakses digit ke `i` sesuai indeks array BigNumber. Operator `[]` ini kalian gunakan sebagai getter dan setter.

Berikut adalah contoh penggunaannya

```
BigNumber bn(123);
cout << bn[0] << endl; // 3
cout << bn[1] << endl; // 2
cout << bn[2] << endl; // 1
cout << bn[3] << endl; // 0
bn[0] = 5;
cout << bn[0] << endl; // 5
```

Silahkan gunakan main sederhana berikut untuk membantu menguji kode kalian.

```
#include <iostream>
#include "BigNumber.hpp"
using namespace std;

ostream& operator<<(ostream& os, const BigNumber& bn) {
    bool foundNonZero = false;
    for (int i = BigNumber::max_digit - 1; i >= 0; i--) {
        if (bn[i] > 0) {
            foundNonZero = true;
        }
        if (foundNonZero) {
            os << bn[i];
        }
    }
    if (!foundNonZero) {
        os << "0";
    }
    return os;
}

int main() {
    BigNumber a(123), b("2210");
    cout << a << endl; // 123
    cout << b << endl; // 2210
    cout << (a < b) << endl; // 1
    cout << (a > b) << endl; // 0
    cout << (a + b) << endl; // 2333
    return 0;
}
```

[BigNumber.hpp](#)

[BigNumber.cpp](#)

Diberikan definisi dan sebagian realisasi BigNumber, lengkapi bagian BigNumber.cpp yang masih kosong (operator ==, <, >, dan +)! Perhatikan hint yang sudah ada juga.

Kumpulkan **BigNumber.cpp**

C++14

 [BigNumber.cpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	16	Accepted	0.00 sec, 2.87 MB
2	16	Accepted	0.00 sec, 2.88 MB
3	16	Accepted	0.00 sec, 2.87 MB
4	16	Accepted	0.00 sec, 2.87 MB
5	16	Accepted	0.01 sec, 2.91 MB
6	20	Accepted	0.00 sec, 2.96 MB