

Started on	Thursday, 9 March 2023, 4:34 PM
State	Finished
Completed on	Thursday, 9 March 2023, 4:43 PM
Time taken	8 mins 9 secs
Marks	379.55/600.00
Grade	63.26 out of 100.00

Question **1**
Partially correct
Mark 54.55 out of 150.00

Time limit	1 s
Memory limit	64 MB

Pecahan (4 Sekawan)

Diberikan file header [Pecahan.hpp](#)
Implementasikan kelas pecahan dengan mengimplementasikan header di atas!

Kumpulkan file dengan nama Pecahan.cpp

C++14

 [Pecahan.cpp](#)

Score: 40

Blackbox

Score: 40

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.98 MB
2	10	Accepted	0.01 sec, 2.99 MB
3	10	Accepted	0.00 sec, 2.89 MB
4	0	Wrong answer	0.00 sec, 3.04 MB
5	0	Wrong answer	0.00 sec, 3.02 MB
6	0	Wrong answer	0.00 sec, 3.05 MB
7	10	Accepted	0.00 sec, 2.89 MB

Time limit	1 s
Memory limit	64 MB

Kebun Kelompok Arsa & Akbar

Saat mengerjakan tugas besar yang bertemakan program kebun, terdapat sebuah bug yang membuat bingung satu kelompok. Arsa dan Akbar yang sudah *mengcarry* kelompok meminta anda membantu untuk mengimplementasikan fungsi **print** dan **print2** pada setiap kelas agar mereka berdua dapat melakukan debugging. Sebagai anggota kelompok yang baik (*jangan gak dikerjain >:()*), bantulah mereka membuat fungsi debugging.

Diberikan sekumpulan kelas Tumbuhan yang bisa dilihat pada [Tumbuhan.hpp](#).

Saat melihat kelas-kelas tersebut, anda memiliki firasat ada yang kurang... Apakah itu??? Silakan ditambahkan **keyword** yang kurang, dan sesuaikan implementasinya dengan keluaran berikut.

Kumpulkan **Tumbuhan.hpp** yang sudah disertai dengan implementasi dan perubahan terhadap keyword yang kurang.

Hint: letakkan **keyword** yang kurang pada tempat yang membutuhkan saja.

Constructor tidak perlu diubah/ditambah print.

Contoh Program:

```
int main() {
    Tumbuhan *array_tumbuhan[3];
    Tumbuhan a("a");
    TumbuhanMakan b("b", "6");
    TumbuhanObat c("c", "7");
    array_tumbuhan[0] = &a;
    array_tumbuhan[1] = &b;
    array_tumbuhan[2] = &c;

    for(int i=0;i<3;i++) {
        array_tumbuhan[i]->print();
        array_tumbuhan[i]->print2();
    }
    array_tumbuhan[1] = &c;
    array_tumbuhan[1]->print();

    TumbuhanMakan *p = new Padi("i", "8");
    p->print();
    p->print2();

    TumbuhanMakan *d = new Jagung("d","1");
    TumbuhanObat *e = new Jahe("e","2");;
    delete p;
    d->print();
    e->print();
    d->print2();
    e->print2();
    Jagung f("f","3");
    Jahe g("g","4");

    f.print();
    g.print();
    f.print2();
    g.print2();
    TumbuhanObat *array_obat[3];
    array_obat[0] = &c;
    array_obat[1] = new Kencur("h", "5");
    array_obat[2] = &g;

    for (int i=0; i < 3; i++) {
        array_obat[i]->print2();
        array_obat[i]->print();
    }
    delete array_obat[1];
    delete e;
    delete d;
    return 0;
}
```

Contoh Keluaran:

(1) Tumbuhan ini milik 'a'
(2) Tumbuhan ini milik 'a'
(1) Tumbuhan Makan ditanam di pot '6'
(2) Tumbuhan ini milik 'b'
(1) Tumbuhan Obat ditanam di pot '7'
(2) Tumbuhan ini milik 'c'
(1) Tumbuhan Obat ditanam di pot '7'
(1) Padi pot '8' tumbuh tinggi
(2) Tumbuhan Makan ditanam di pot '8'
Padi pot '8' mati
TumbuhanMakan pot '8' mati
Tumbuhan milik 'i' mati
(1) Jagung pot '1' tumbuh segar
(1) Jahe pot '2' terlihat aman
(2) Tumbuhan Makan ditanam di pot '1'
(2) Tumbuhan Obat ditanam di pot '2'
(1) Jagung pot '3' tumbuh segar
(1) Jahe pot '4' terlihat aman
(2) Jagung pot '3' tumbuh segar
(2) Jahe pot '4' terlihat aman
(2) Tumbuhan Obat ditanam di pot '7'
(1) Tumbuhan Obat ditanam di pot '7'
(2) Tumbuhan Obat ditanam di pot '5'
(1) Kencur pot '5' terlihat sehat
(2) Tumbuhan Obat ditanam di pot '4'
(1) Jahe pot '4' terlihat aman
Kencur pot '5' mati
TumbuhanObat pot '5' mati
Tumbuhan milik 'h' mati
Jahe pot '2' mati
TumbuhanObat pot '2' mati
Tumbuhan milik 'e' mati
Jagung pot '1' mati
TumbuhanMakan pot '1' mati
Tumbuhan milik 'd' mati
Jahe pot '4' mati
TumbuhanObat pot '4' mati
Tumbuhan milik 'g' mati
Jagung pot '3' mati
TumbuhanMakan pot '3' mati
Tumbuhan milik 'f' mati
TumbuhanObat pot '7' mati
Tumbuhan milik 'c' mati
TumbuhanMakan pot '6' mati
Tumbuhan milik 'b' mati
Tumbuhan milik 'a' mati

C++14

 [Tumbuhan.hpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	100	Accepted	0.00 sec, 2.87 MB

Time limit	1 s
Memory limit	64 MB

Pak William adalah seorang guru matematika SD yang juga mempunyai hobi membuat program. Pada suatu hari, Pak William tidak sengaja merusak kalkulator miliknya. Untuk menanggulangi hal tersebut sembari mengasah kemampuan koding, Pak William berniat untuk membuat program kalkulator sederhana demi menggantikan beberapa fungsionalitas dari kalkulator yang rusak.

Program kalkulator yang ingin dibuat dapat menerima dan mengevaluasi hasil dari ekspresi aritmetika yang mengandung operator "`(`", "`)`", "`+`", "`-`", "`*`", dan "`/`". Program hanya dapat menerima dan memproses integer (**bukan `float`/`double`**). Berikut adalah contoh evaluasi ekspresi aritmetika

Contoh Ekspresi Aritmetika Pada Kalkulator

- 1. $2 * (1 + 3) = 2 * (4) = 8$
- 2. $(2 - 3) * (5 / 2) = (-1) * (2) = -2$
- 3. $4 * (5 - 1) / 3 = 4 * 4 / 3 = 16 / 3 = 5$

Kalkulator akan memanfaatkan salah satu STL yaitu stack untuk membuat 2 stack, satu stack berisikan operator dan satu stack berisikan operand. **Penjelasan dari algoritma kalkulator dapat dibaca pada header yang disediakan.**

Berikut adalah beberapa batasan atau catatan dari program yang akan diimplementasikan

- 1. Simbol yang valid pada ekspresi aritmetika adalah **angka**, "`(`", "`)`", "`+`", "`-`", "`*`", dan "`/`".
- 2. Operator "`+`" **tidak** akan digunakan untuk operasi unary (simbol "`+1`" atau "`+(2+1)`" **bukanlah** ekspresi yang **valid**).
- 3. Operator "`-`" tidak akan digunakan untuk operasi unary (simbol "`-1`" atau "`-(2+1)`" **bukanlah** ekspresi yang **valid**).
- 4. Tidak akan ada dua operator yang muncul berurutan ("`(`" dan "`)`" tidak termasuk) dan tidak akan ada dua operand yang muncul berurutan.
- 5. Operand yang muncul pada ekspresi aritmetika dipastikan selalu satuan (di rentang "`0-9`", tidak mungkin puluhan keatas)
- 6. Setiap "`(`" mempunyai "`)`" yang berkorespondensi.
- 7. Setiap angka yang masuk adalah integer dan kalkulasi dilakukan secara integer (tidak perlu menambahkan floor atau ceiling saat melakukan pembagian, ikuti saja behaviour `c++` saat terjadi pembagian integer/integer).

Disediakan header [CException.hpp](#), [Calculator.hpp](#), dan [Calculator.cpp](#). Kumpulkan dan lengkapi implementasi method-method pada kelas Calculator di [Calculator.cpp](#).

Perhatikan bahwa terdapat **exception** yang harus ditangani yaitu

- 1. Saat ditemukan simbol yang tidak valid, semisal "`^`" (pangkat) atau " " (spasi).
- 2. Terjadi pembagian oleh angka 0.

Sefruit Tips: Buatlah main program untuk menguji kebenaran kode yang sudah kalian buat

Penjelasan STL Stack

Stack adalah kontainer data dengan mekanisme LIFO (Last In First Out). Elemen paling puncak pada stack disebut sebagai TOP. Penyisipan pada stack selalu dilakukan "di atas" TOP dan penghapusan selalu dilakukan pada TOP. TOP adalah satu-satunya alamat tempat terjadi operasi.

Stack mempunyai beberapa method dasar seperti

- 1. **`bool empty()`** – Mengembalikan apakah stack sedang kosong.
- 2. **`size_type size()`** – Mengembalikan ukuran stack.
- 3. **`reference top()`** – Mengembalikan reference kepada element TOP pada stack.
- 4. **`void push(value_type g)`** – Menambahkan elemen 'g' diatas TOP pada stack.
- 5. **`void pop()`** – Menghapus element TOP pada stack.

Berikut adalah contoh penggunaan STL stack

```
1 #include <iostream>
2
3 #include <stack>
4
5 using namespace std;
6
7 int main() {
8
9     stack<int> stack;
10
11     stack.push(21);
12
13     stack.push(22);
14
15     stack.push(24);
16
17     stack.push(25);
18
19     int num=0;
20
21     stack.push(num);
22
23     stack.pop();
24
25     stack.pop();
26
27     stack.pop();
28
29
30
31     while (!stack.empty()) {
32
33         cout << stack.top() <<" ";
34
35         stack.pop();
36
37     }
38
39 }
```

C++14

 [Calculator.cpp](#)

Score: 90

Blackbox

Score: 90

Verdict: Runtime error

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.93 MB
2	10	Accepted	0.00 sec, 2.91 MB
3	10	Accepted	0.04 sec, 2.91 MB
4	10	Accepted	0.02 sec, 2.93 MB
5	10	Accepted	0.00 sec, 2.98 MB
6	10	Accepted	0.00 sec, 3.01 MB
7	10	Accepted	0.00 sec, 2.99 MB
8	10	Accepted	0.02 sec, 2.98 MB
9	0	Wrong answer	0.00 sec, 3.01 MB
10	0	Wrong answer	0.00 sec, 2.89 MB
11	0	Wrong answer	0.01 sec, 2.94 MB
12	0	Wrong answer	0.00 sec, 2.96 MB
13	0	Wrong answer	0.01 sec, 2.91 MB

No	Score	Verdict	Description
14	0	Wrong answer	0.00 sec, 2.94 MB
15	10	Accepted	0.00 sec, 2.90 MB
16	0	Wrong answer	0.00 sec, 3.01 MB
17	0	Wrong answer	0.00 sec, 3.07 MB
18	0	Runtime error	0.19 sec, 2.94 MB

◀ Slide Responsi 3

Jump to...

Pra Responsi 4 (Mystery Box) ▶