

C++プログラミング III 実験／（後期）2022 年

出題日 2022年10月6日（木）

履修者は自分の担当者に CoursePower で提出. 提出〆切 10 月 12 日（水） **24:00**

課題 1 : CoursePower から prac02_skel.cpp をダウンロードし使用すること.

prac02_skel.cpp では分数のクラス Fraction が部分的に作成されている. main 関数が動作するように必要なプログラムを作成しなさい.

```
#実行例 1 (数字の間にはスペースを入れること)
$./a.exe
分数を入力(g1)-->1 6
分数を入力(g2)-->1 3
g1+g2 = (1/6)+(1/3) = (1/2)
g1-g2 = (1/6)-(1/3) = (-1/6)
g1*g2 = (1/6)*(1/3) = (1/18)
g1/g2 = (1/6)/(1/3) = (1/2)
g1*定数 = (1/6)*2 = (1/3)
定数*g1 = 2*(1/6) = (1/3)
```

```
#実行例 2 (数字の間にはスペースを入れること)
$./a.exe
分数を入力(g1)-->1 2
分数を入力(g2)-->1 -3
g1+g2 = (1/2)+(-1/3) = (1/6)
g1-g2 = (1/2)-(-1/3) = (5/6)
g1*g2 = (1/2)*(-1/3) = (-1/6)
g1/g2 = (1/2)/(-1/3) = (-3/2)
g1*定数 = (1/2)*2 = 1
定数*g1 = 2*(1/2) = 1
```

```
#include <iostream>
#include <cstdlib>
using std::cin, std::cout, std::endl, std::cerr;

class Fraction
{
private:
    int numerator; // 分子
    int denominator; // 分母
    int gcd(int x, int y) const
    {
        //最大公約数を求める関数(ユークリッドの互除法)。外部関数でも良い
        // xをyで割った余りをzとすると、xとyの最大公約数はyとzの最大公約数に等しい
        int z{x % y};
        while (z != 0)
        {
            x = y;
            y = z;
            z = x % y;
        }
        return y;
    }
}
```

```

public:
    Fraction(int n = 0, int d = 1)
    { // コンストラクタ(デフォルト引数の指定)
        numerator = n;
        denominator = d;
        if (denominator < 0)
        { //分母が負の場合、代わりに分子の符号を変更
            numerator = -numerator;
            denominator = -denominator;
        }
    }
    void reduce()
    {
        if (denominator == 0)
        {
            cerr << "??分母が0!!" << endl; //分母が0のとき警告を出して止める
            exit(EXIT_FAILURE);
        }
        else if (numerator == 0)
        { // 何もしない。出力の際に0が表示される
        }
        else
        {
            int bunshi = numerator;
            if (bunshi < 0)
            { // 分子が負のとき
                bunshi = -bunshi;
            }
            int bunbo = denominator;
            if (bunbo < 0)
            { // 分母が負のとき
                bunbo = -bunbo;
            }
            int saidaikouyakusuu = gcd(bunshi, bunbo);
            numerator /= saidaikouyakusuu;
            denominator /= saidaikouyakusuu;
        }
    }
}

```

/*追加する関数

- >>演算子のオーバーロード:分母が0の場合、「標準入力で分母に0が設定されています」とエラー文も出す。
分母がマイナスの場合、分子にマイナスが付くようにする
- <<演算子のオーバーロード:分子が0の場合、0を出力。分母が1の場合、分母が出力されないようにする
- +演算子のオーバーロード:分数の足し算をする
- 演算子のオーバーロード:分数の引き算をする
- *演算子のオーバーロード:「定数*分数」と「分数*定数」にも対応できるようにする
- /演算子のオーバーロード:分数の割り算をする

```
*/  
};
```

```
int main()  
{  
    Fraction g1, g2, ans;  
    int n{2};  
  
    cout << "分数を入力(g1)-->"; // 分子、分母の順で入力 3 4 など  
    cin >> g1;  
    cout << "分数を入力(g2)-->";  
    cin >> g2;  
    ans = g1 + g2;  
    cout << "g1+g2 = " << g1 << "+" << g2 << " = " << ans << "\n";  
    ans = g1 - g2;  
    cout << "g1-g2 = " << g1 << "-" << g2 << " = " << ans << "\n";  
    ans = g1 * g2;  
    cout << "g1*g2 = " << g1 << "*" << g2 << " = " << ans << "\n";  
    ans = g1 / g2;  
    cout << "g1/g2 = " << g1 << "/" << g2 << " = " << ans << "\n";  
    ans = g1 * n;  
    cout << "g1*定数 = " << g1 << "*" << n << " = " << ans << "\n";  
    ans = n * g1;  
    cout << "定数*g1 = " << n << "*" << g1 << " = " << ans << "\n";  
  
    return 0;  
}
```