

实验一：实数的运算

由于学过 C++ 所有使用面向对象的思想来写。

截图

1. 输入为分数

```
C:\Windows\System32\cmd.exe - PAUSE
请输入两个有理数：输入分数请输f, 其他输入认为是输入小数：f
注意：分数输入格式为 分子/分母 ！（分子分母用/隔开）
现在输入第一个数：1/2
现在输入第二个数：1/3
请输入运算方式（+,-,*,/）：+
答案是（分数形式）：5/6
答案是（小数形式）：0.833333
Press any key to continue . . .
```

2. 输入为小数

```
C:\Windows\System32\cmd.exe - PAUSE
请输入两个有理数：输入分数请输f, 其他输入认为是输入小数：x
现在输入第一个数：1.23
现在输入第二个数：3.21
请输入运算方式（+,-,*,/）：+
答案是（分数形式）：111/25
答案是（小数形式）：4.44
Press any key to continue . . .
```

Code

```
#include <bits/stdc++.h>

using namespace std;

inline int lcm(int a, int b) {
    return a / __gcd(a, b) * b;
}

namespace sux {
    class real {
        int deno = 1, mole = 0; // mu zi
    public:
        real operator+(real &r) const {
            int nd = lcm(this->deno, r.deno); // mu
            int nm = this->mole * (nd / this->deno) + r.mole * (nd / r.deno);
            int sign = nm > 0 ? 1 : -1;
```

```

        int sign = nm > 0 ? 1 : -1;
        nm *= sign;
        int div = __gcd(nm, nd);
        return {sign * nm / div, nd / div};
    }

    real operator-(real &r) const {
        int nd = lcm(this->deno, r.deno); //mu
        int nm = this->mole * (nd / this->deno) - r.mole * (nd / r.deno);
        int sign = nm > 0 ? 1 : -1;
        nm *= sign;
        int div = __gcd(nm, nd);
        return {sign * nm / div, nd / div};
    }

    real operator*(real &r) const {
        int nm = this->mole * r.mole;
        int nd = this->deno * r.deno;
        int sign = nd * nm > 0 ? 1 : -1;
        nm = abs(nm), nd = abs(nd);
        int div = __gcd(nm, nd);
        return {sign * nm / div, nd / div};
    }

    real operator/(real &r) const {
        int nm = this->mole * r.deno;
        int nd = this->deno * r.mole;
        int sign = nd * nm > 0 ? 1 : -1;
        nm = abs(nm), nd = abs(nd);
        int div = __gcd(nm, nd);
        return {sign * nm / div, nd / div};
    }

    double toDouble() const {
        return (double) this->mole / this->deno;
    }

    friend ostream &operator<<(ostream &o, real &r) {
        o << r.mole << '/' << r.deno;
        return o;
    }

    friend istream &operator>>(istream &i, real &r) {
        char ch;
        int m, d;
        i >> m >> ch >> d;
        r = sux::real(m, d);
        return i;
    }

    real(int m, int d) {
        int gcd = __gcd(m, d);
        this->mole = m / gcd;
        this->deno = d / gcd;
    }

```

```

        real() : deno(1), mole(0) {}
    };
}

int main() {
    sux::real a, b, ans;
    double x, y;

    function<void(sux::real &)> calc = [](sux::real &r) {
        double x;
        cin >> x;
        int a = (int) x, t = 1;
        while (a != x) {
            x *= 10;
            a = (int) x;
            t *= 10;
        }
        r = sux::real(a, t);
    };

    cout << "请输入两个有理数： ";
    cout << "输入分数请输f,其他输入认为是输入小数： ";
    string str;
    cin >> str;
    if (str[0] == 'f') cout << "注意：分数输入格式为 分子/分母 ! (分子分母用/隔开) \n";
    cout << "现在输入第一个数： ";
    if (str[0] != 'f') calc(a);
    else cin >> a;

    cout << "现在输入第二个数： ";
    if (str[0] != 'f') calc(b);
    else cin >> b;
    cout << "请输入运算方式 (+,-,*,/): ";
    cin >> str;
    if (str[0] == '+') ans = a + b;
    else if (str[0] == '-') ans = a - b;
    else if (str[0] == '*') ans = a * b;
    else if (str[0] == '/') ans = a / b;
    else {
        cout << "非法输入! ";
        return 0;
    }
    cout << "答案是 (分数形式) : " << ans << endl;
    cout << "答案是 (小数形式) : " << ans.toDouble() << endl;
}

```