```
//日期: 2018/ 时间:
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
using namespace std;
//制定以下总则
//down为非负数，如果分数为负，令分子up为负即可;
//如果该分数为0，那么规定其分子为0，分母为1
//分子和分母没有除了1以外的公约数
struct Fraction{
    int up,down;
};

int gcd(int a,int b){
    return !b ? a : gcd(b,a%b);
}

//分数化简，化简步骤有三:
//如果分母down为负数，那么分子分母取相反数
//如果分子up为0，那么令分母down为1
//找出分子分母绝对值的最大公约数，分子分母分别除以这个数
Fraction reduction(Fraction result){
    if(result.down < 0){
        result.up = -result.up;
        result.down = -result.down;
    }
    if(result.up == 0)
        result.down = 1;
    else{
        int d = gcd(abs(result.down),abs(result.up));
        result.down /= d;
        result.up /= d;
    }

    return result;
}

//分数的输出
void showResult(Fraction r){
    r = reduction(r);
    if(r.down == 1) printf("%d",r.up);
    else if(abs(r.up) > r.down){
        printf("%d %d/%d",r.up/r.down,abs(r.up)%r.down,r.down);
    }else{
        printf("%d/%d",r.up,r.down);
    }
}

//以下的加减乘除都是一样的。
Fraction add(Fraction f1,Fraction f2){
    Fraction result;
    result.up = f1.up * f2.down + f2.up * f1.down;
    result.down = f1.down * f2.down;
    return reduction(result);
}

```

```
Fraction minu(Fraction f1,Fraction f2){
    Fraction result;
    result.up = f1.up * f2.down - f2.up * f1.down;
    result.down = f1.down * f2.down;
    return reduction(result);
}

Fraction multi(Fraction f1,Fraction f2){
    Fraction result;
    result.up = f1.up * f2.up;
    result.down = f1.down * f2.down;
    return reduction(result);
}

Fraction divide(Fraction f1,Fraction f2){
    Fraction result;
    result.up = f1.up * f2.down;
    result.down = f1.down * f2.up;
    return reduction(result);
}

int main(){


    return 0;
}


```