//somsc 200

//completed

//boli zhang

// week5 hw

// *fracCalculator.cpp*

#include <iostream>

#include <cstdlib> // labs()

using namespace std;

struct frac{

int num;

int den;

};

frac inputfrac() { //input

char dummychar;

frac f;

while(1) {

cin >> f.num; //numerator

cin >> dummychar; //'/' character

cin >> f.den; //denominator

if(f.den != 0) break; //no problem, so exit loop

cout << "Denominator cannot be zero. Try again: ";

}

return f;

}

frac lowterms(frac a) { //arg reduced to lowest terms

frac t; //temporary fraction

long tlong, gcd;

t.num = labs(a.num); //use non-negative copies

t.den = labs(a.den);

if( t.num!=0 && t.den==0 ) //check for n/0

{ cout << "Illegal fraction: division by 0"; exit(1); }

else if( t.num==0 ) //check for 0/n

{ t.num=0; t.den = 1; return(t); }

//this 'while' loop finds the gcd of t.num and t.den

while(t.num != 0) {

if(t.num < t.den) //ensure numerator larger

{ tlong=t.num; t.num=t.den; t.den=tlong; }

t.num = t.num - t.den; //subtract them

}

gcd = t.den;

t.num = a.num / gcd; //divide both num and den by gcd

t.den = a.den / gcd; //to reduce frac to lowest terms

return t;

}

int main()

{

// frac f1, f2, fans;

frac f1, f2, fans;

char op;

do {

cout << "\nEnter fraction, operator, fraction";

cout << "\nform 3/4 + 3/8 (or 0/1 + 0/1 to exit): ";

f1 = inputfrac();

cin >> op;

f2 = inputfrac();

cout << "You have entered: " << f1.num << "/" << f1.den

<< " " << op << " " << f2.num << "/" << f2.den << endl;

switch(op) {

case '+':

fans.num = f1.num \* f2.den + f2.num \* f1.den;

fans.den = f1.den \* f2.den;

break;

case '-':

fans.num = f1.num \* f2.den - f2.num \* f1.den;

fans.den = f1.den \* f2.den;

break;

case '\*':

fans.num = f1.num \* f2.num;

fans.den = f1.den \* f2.den;

break;

case '/':

fans.num = f1.num \* f2.den;

fans.den = f1.den \* f1.num;

break;

default:

cout << "No such operator";

} //end switch

fans = lowterms(fans);

cout << fans.num << "/" << fans.den << endl;

} while( f1.num != 0 || f2.num != 0 );

cout << endl;

return 0;

}

//somsc 200

//completed

//boli zhang

// week5 hw

// *frac.h*

*// comsc 200*

*// boli zhang*

*// completed*

*// Frac.h*

*// lab1*

*//*

*// Created by Jeff on 9/7/16.*

*// Copyright © 2016 Jeff zhang. All rights reserved.*

*//*

*#ifndef Frac\_h*

*#define Frac\_h*

*#include <string>*

*class Frac;*

*std::ostream &operator << (std::ostream &, const Frac &);*

*class Frac{*

*private:*

*long num;*

*long den;*

*long gcd(long a, long b){*

*long c = a%b;*

*while(c!=0){*

*a = b;*

*b = c;*

*c = a%b;*

*}*

*return b;*

*};*

*void lowTerm(){*

*long g = gcd(num,den);*

*num = num/g;*

*den = den/g;*

*};*

*public:*

*Frac(){*

*num=0;*

*den=1;*

*};*

*Frac(long n){*

*num=n;*

*den=1;*

*};*

*Frac(long n,long d){*

*num=n;*

*den=d;*

*lowTerm();*

*};*

*Frac(const Frac &x){ //copy constructor: Frac x(y) that accept a Frac object as an argument.*

*num = x.num;*

*den = x.den;*

*};*

*void set(long n, long d){*

*num=n;*

*den=d;*

*lowTerm();*

*};*

*//post pre increment*

*Frac operator + (const Frac &x){*

*Frac temp;*

*temp.num = num\*x.den + x.num\*den;*

*temp.den = den\*x.den;*

*return temp;*

*};*

*Frac& operator - (const Frac &right){*

*Frac left;*

*left.num = num\*right.den - right.num\*den;*

*left.den = den\*right.den;*

*return left;*

*};*

*Frac& operator \* (const Frac &right){*

*Frac left;*

*left.num = num\*right.num;*

*left.den = den\*right.den;*

*return left;*

*};*

*Frac& operator / (const Frac &right){*

*Frac left;*

*left.num = num\*right.den;*

*left.den = den\*right.num;*

*return left;*

*};*

*std::string returnObj(){*

*std::string s = std::to\_string(num) + "/" + std::to\_string(den);*

*return s;*

*}*

*void show(){*

*std::cout<<num << "/" <<den ;*

*}*

*std::string toString(){*

*return std::to\_string(num) + "/" + std::to\_string(den);*

*}*

*// friend ostream operator << function definition*

*friend std::ostream &operator << (std::ostream &strm, const Frac &right){*

*strm << right.num << "/" << right.den;*

*return strm;*

*}*

*};*

*#endif /\* Frac\_h \*/*

