## Практические домашние задания для Урока 10 Модуля 3

1. Создай класс Fraction (обыкновенная дробь). Добавь конструктор, который будет принимать два целых числа и добавлять объекту атрибуты - числитель и знаменатель. Добавь метод print который будет возвращать текстовое представление объекта в формате: «Дробь {числитель}/{знаменатель}», например «Дробь 1/2». Переопредели методы сравнения ==, !=, >, <, >=, <=. Методы должны возвращать True или False.

Считай с клавиатуры четыре целых чисел - числитель и знаменатель первой дроби и числитель и знаменатель второй дроби. Создай экземпляры класса Fraction с параметрами считанными с клавиатуры. Выведи результаты сравнения==, !=, >, <, >=, <= этих дробей.

Входные данные:

Вводится четыре строки, каждая строка содержит целое число - числитель и знаменатель первой дроби и числитель и знаменатель второй дроби.

**Выходные данные:**

Выводится 6 строк 1 или 0.

**Пример ввода:**

1

2

2

3

**Пример вывода:**

0

1

0

1

0

1

## Решение.



#include <iostream>

using namespace std;

class Fraction

{

public:

int num, den;

Fraction(int n, int d)

{

num=n/NOD(n, d);

den = d/NOD(n, d);

}

int NOD(int a, int b)

{

while(a > 0 && b > 0)

if(a > b)

a %= b;

else

b %= a;

return a + b;

}

int get\_common\_denominator(int den1,int den2)

{

int common\_den = den1 \* den2 / NOD(den1, den2);

return common\_den;

}

void print()

{

cout<<"Дробь "<< num<<"/"<<den<<endl;

}

bool operator ==(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num == common\_den / c2.den \* c2.num;

}

bool operator !=(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num != common\_den / c2.den \* c2.num;

}

bool operator >(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num > common\_den / c2.den \* c2.num;

}

bool operator <(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num < common\_den / c2.den \* c2.num;

}

bool operator >=(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num >= common\_den / c2.den \* c2.num;

}

bool operator <=(Fraction c2)

{

int common\_den = this->get\_common\_denominator(this->den, c2.den);

return common\_den / this->den \* this->num <= common\_den / c2.den \* c2.num;

}

};

int main()

{

int num, den;

cin>>num;

cin>>den;

Fraction fraction1(num, den);

cin>>num;

cin>>den;

Fraction fraction2(num, den);

bool b=fraction1 == fraction2;

cout<<b<<endl;

b=fraction1 != fraction2;

cout<<b<<endl;

b=fraction1 > fraction2;

cout<<b<<endl;

b=fraction1 < fraction2;

cout<<b<<endl;

b=fraction1 >= fraction2;

cout<<b<<endl;

b=fraction1 <= fraction2;

cout<<b<<endl;

return 0;

}