Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

Кафедра прикладной математики и кибернетики

**Лабораторная работа №1**  
 **по дисциплине**  
 **Алгоритмы и вычислительные методы оптимизации**

Бригада 2

Выполнил:

студент гр.ИП-017 Костин А.В.

«14» февраля 2023 г.

Новосибирск 2023 г.

**Задание**

Написать программу, находящую решение системы линейных уравнений методом Жордана-Гаусса с выбором главного элемента в столбце.

**Входные данные**

На вход программе подаются коэффициенты системы линейных уравнений (считываются из файла в виде матрицы размера m×(n+1)):

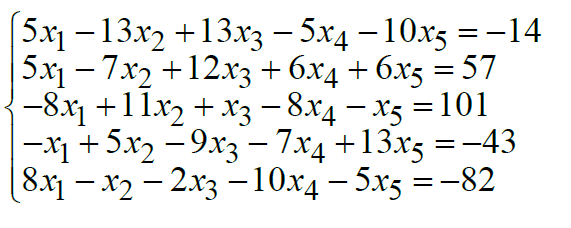
**Требования к программе**

Программа должна:

* выводить промежуточные матрицы после каждого шага исключений и решение системы;
* работать для различных случаев решений системы: система имеет единственное решение, система имеет бесконечно много решений, система не имеет решения.

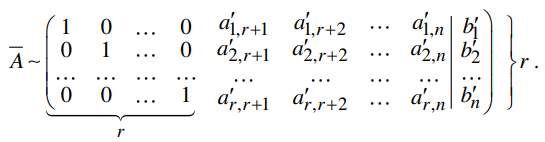
Начальную работу программу необходимо продемонстрировать на предложенной ниже системе (система выбирается по номеру бригады).

Для получения максимальной оценки необходимо, чтобы все вычисления выполнялись в простых дробях. Для этого реализовать класс простых дробей. Реализованный класс можно будет использовать в лабораторной №2 и курсовой работе.



**Теоретические сведения**

Суть метода Жордана-Гаусса состоит в том, чтобы, выполнив элементарные преобразования над строками матрицы , аналогично алгоритму нахождения ранга матрицы A, получить матрицу вида:



Преобразования матрицы называются жордановыми исключениями. В процессе жордановых исключений возможны следующие случаи:

1. Получена строка, состоящая из нулей, кроме последнего коэффициента (правой части уравнения). В этом случае система не имеет решения.

2. Ранг матрицы А равен количеству неизвестных n (r = n). Тогда система имеет единственное решение.

3. Ранг матрицы А меньше количества неизвестных r < n. Тогда система имеет бесконечно много решений.

Если выразить в системе, приведенной к единичному базису, базисные переменные через свободные, то получим общее решение системы.

**Описание используемых алгоритмов**

Реализация класса простых дробей.

Задаем в классе две переменные для числителя и знаменателя. Перегружаем все операторы для этого класса. Реализуем метод Жордана-Гауса с использованием этих дробей.

**Результат работы**

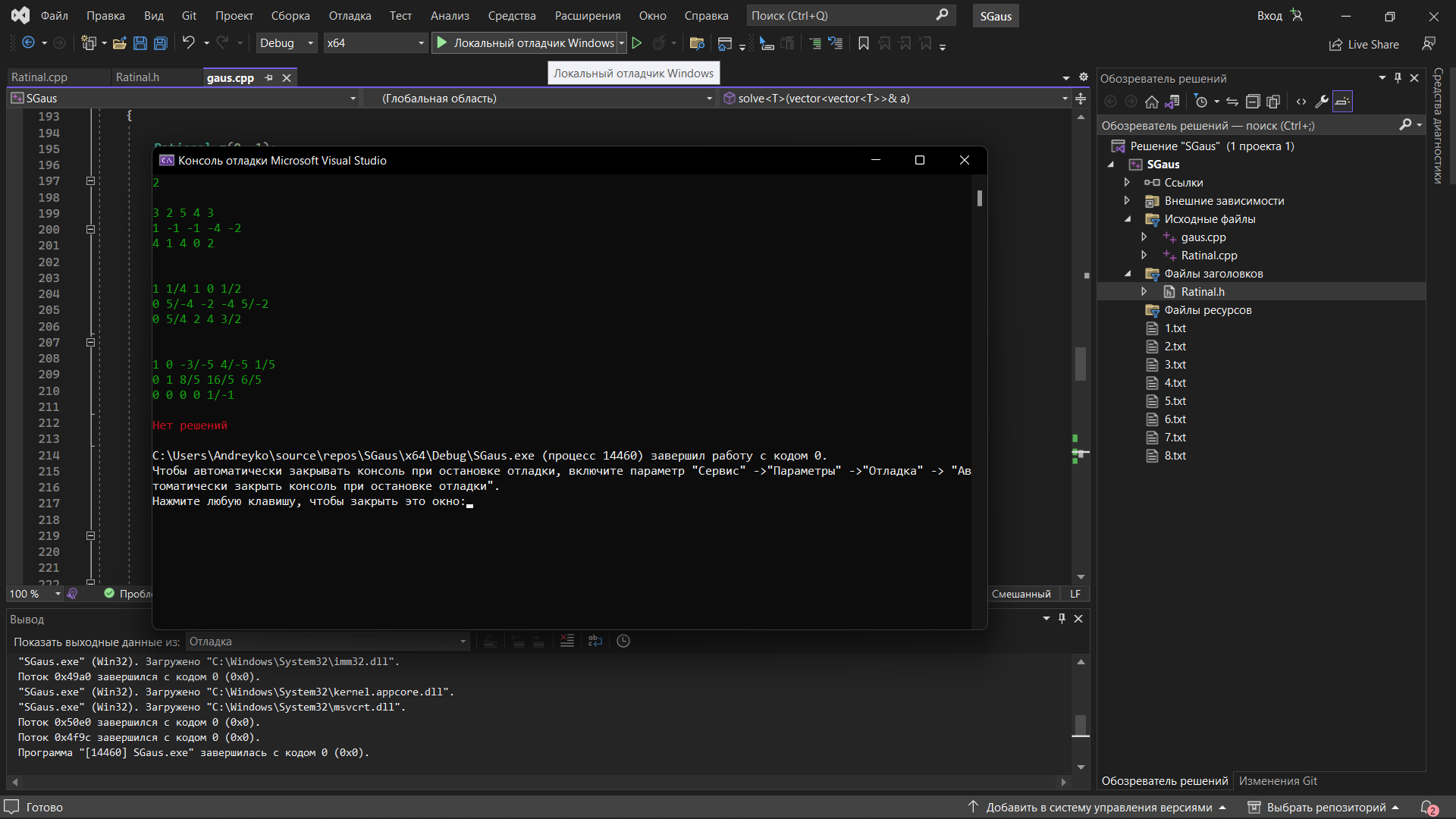


Рисунок 1. Результат работы программы с матрицей, которая не имеет решения

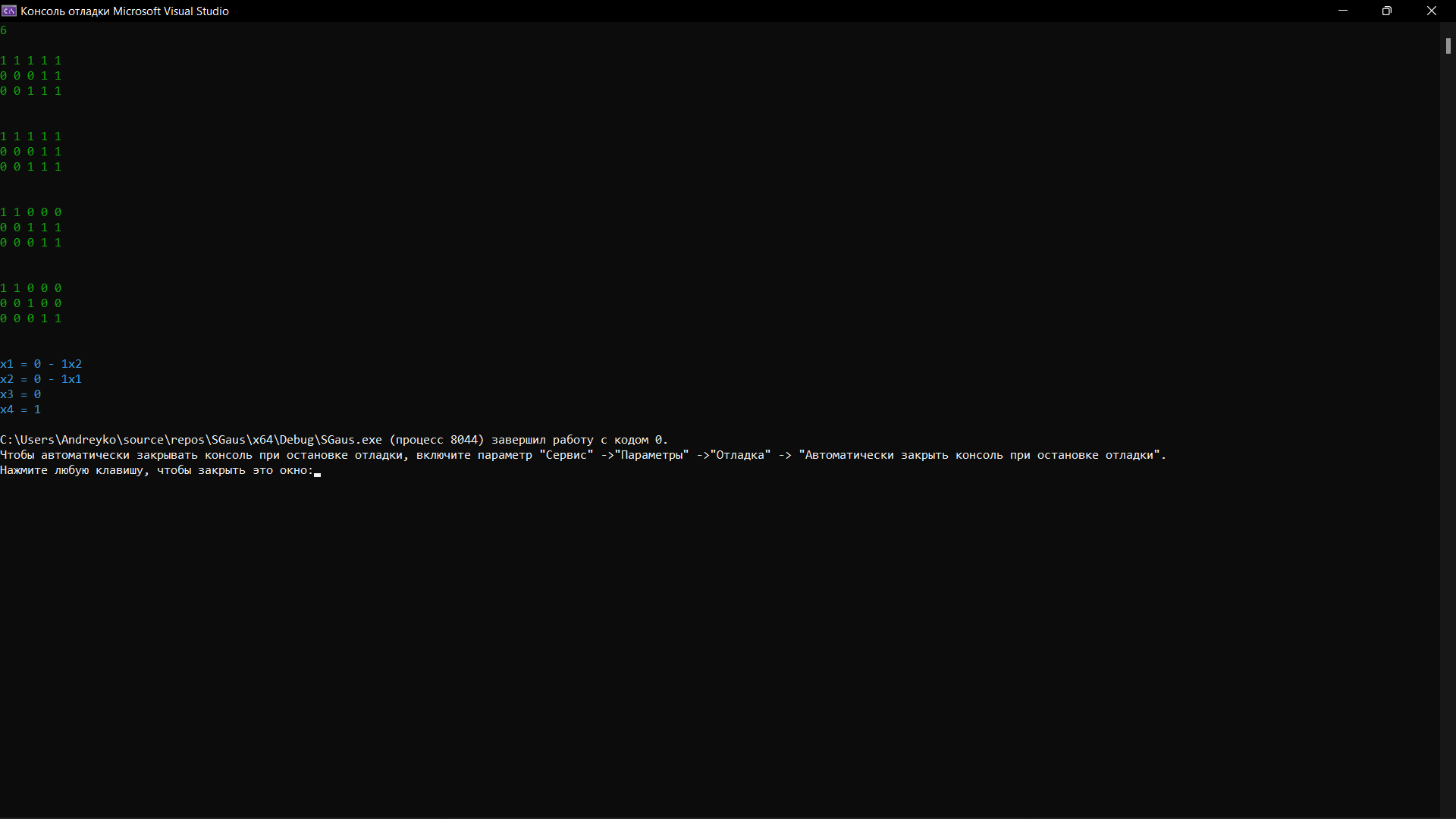


Рисунок 2. Результат работы программы с матрицей, которая имеет множество решений



Рисунок 3. Результат работы программы с примером 2

**Исходный код программы**

***gaus.cpp***

#include <string>

#include <fstream>

#include <iostream>

#include <vector>

#include <iomanip>

#include "Ratinal.h"

#include <cassert>

#include <conio.h>

#include <stdio.h>

#include <Windows.h>

using namespace std;

//abobaboba

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE); //получаем наше окно, в котором нужно изменить цвет

bool operator!= (Rational f, int s)

{

double c = double(double(f.a) / double(f.b));

//cout << "a/b = " << c << " s = " << s;

if (c != s)

return true;

return false;

}

bool operator== (Rational f, int s)

{

double c = double(double(f.a) / double(f.b));

if (c == s)

return true;

return false;

}

bool operator< (Rational f, int s)

{

double c = double(double(f.a) / double(f.b));

if (c < s)

return true;

return false;

}

bool operator> (Rational f, int s)

{

double c = double(double(f.a) / double(f.b));

if (c > s)

return true;

return false;

}

bool operator>= (Rational f, Rational s)

{

double c1 = double(double(f.a) / double(f.b));

double c2 = double(double(s.a) / double(s.b));

if (c1 >= c2)

return true;

return false;

}

ostream& operator<< (ostream& out, Rational s)

{

if (s.b == 1 || s.a == 0)

return out << s.a;

return out << s.a << "/" << s.b;

}

istream& operator>> (istream& out, Rational& s)

{

double number;

out >> number;

std::stringstream ss;

ss << std::setprecision(15) << number;

std::string strNum = ss.str();

size\_t pos = strNum.find('.');

int len = 0;

s.b = 1;

if (pos != strNum.npos)

{

len = strNum.size() - 1 - pos;

s.a = number;

for (int i = 0; i < len; ++i)

{

s.a \*= 10;

s.b \*= 10;

}

//cout << " (" << s.a << " " << s.b<<") ";

}

else

{

s.a = number;

//cout << " (" << s.a << " " << s.b << ") ";

}

return out;

}

int size;

vector<string> vars;

int whitespace = 0;

template<class T>

void read(vector<vector <T> >& matrix, string fileName)

{

ifstream file;

file.open(fileName);

string str;

vector<T> m;

getline(file, str);

whitespace = 0;

string s = "";

for (int i = 0; i <= str.size(); i++) {

if (str[i] == ' ' || i == str.size())

{

whitespace++;

vars.push\_back(s);

s = "";

}

else

{

s += str[i];

}

}

for (int i = 0; true; i++)

{

vector<T> newLine;

for (int j = 0; j < whitespace + 1; j++)

{

T a;

file >> a;

newLine.push\_back(a);

if (file.eof())

break;

}

matrix.push\_back(newLine);

if (file.eof())

break;

}

}

template<class T>

void printMatrix(vector<vector <T> > matrix)

{

cout << "\n";

for (int i = 0; i < matrix.size(); ++i)

{

for (int j = 0; j < matrix[i].size(); ++j)

cout << matrix[i][j] << " ";

cout << '\n';

}

cout << "\n";

}

template<class T>

void composition(vector<T>& i, vector<T>& j, T k) // i = j \*k +i

{

for (int d = 0; d < i.size(); d++) {

T c = j[d] \* k;

//cout << "|| " << i[d] << " + " << c << " ||";

i[d] = i[d] + c;

}

}

template<class T>

bool divStr(vector<T>& d, T k)

{

if (k == 0) {

SetConsoleTextAttribute(h, 4);

cout << "Нет решения!";

return false;

}

for (int i = 0; i < d.size(); i++) {

d[i] = d[i] / k;

//cout << "| " << d[i] << " |";

}

return true;

}

template<class T>

bool solve(vector<vector<T>>& a)

{

int size = min(a.size(), whitespace);

for (int k = 0; k < size; k++)

{

Rational m(0, 1);

int id = -1;

for (int j = k; j < a.size(); ++j)

{

//cout<<"\n-"<<j<<"-\n";

if (rabs(a[j][k]) >= m)

{

//cout << "\n" << rabs(a[j][k]) << " > " << m << "\n";

m = rabs(a[j][k]);

id = j;

}

else

{

//cout << "\n" << rabs(a[j][k]) << " < " << m << "\n";

}

//cout << "\n-\n";

}

//cout << "\n--\n";

if (id != k)

swap(a[id], a[k]);

int step = 0;

//cout << "\n--\n";

if (a[k][k] == 0)

{

bool flag = false;

while (flag == false)

{

m.a = -1;

m.b = 1;

id = -1;

for (int i = k; i < size; i++)

{

if (a[i][k + step] != 0 && rabs(a[i][k + step]) >= m)

{

m = rabs(a[i][k + step]);

id = i;

}

}

if (id != -1)

{

flag = true;

//cout << "\*";

swap(a[id], a[k]);

//cout << "\*";

break;

}

//cout << "\n( " << k << " " << step << ")\n";

if (k + step == size && a[k][k + step + 1] != 0)

{

SetConsoleTextAttribute(h, 4);

cout << "Нет решений\n";

return false;

}

//cout << "\n-\n";

if (k + step == size && a[k][k + step + 1] == 0)

{

//cout << "\n--\n";

a.erase(a.begin() + k);

size = min(a.size(), whitespace);

//cout << "\n---\n";

break;

}

//cout << "\n----\n";

if (flag == false)

{

step++;

}

}

}

if (k >= size)

break;

if (a[k][k + step] != 1)

{

if (divStr(a[k], a[k][k + step]) == false)

return false;

}

/\*else

cout << a[k][k] << " != 1";\*/

for (int i = 0; i < k; i++)

composition(a[i], a[k], a[i][k + step] \* -1);

for (int i = k + 1; i < a.size(); i++)

composition(a[i], a[k], a[i][k + step] \* -1);

printMatrix(a);

}

cout << "\n";

return true;

}

int main()

{

setlocale(0, "");

SetConsoleTextAttribute(h, 2);

vector<vector <Rational> > matrix;

string fileName;

cin >> fileName;

read(matrix, fileName + ".txt");

printMatrix(matrix);

//solve(matrix);

if (solve(matrix) == true)

{

SetConsoleTextAttribute(h, 3);

for (int i = 0; i < vars.size(); ++i)

{

int line = -1;

for (int j = 0; j < matrix.size(); ++j)

{

if (matrix[j][i] == 1)

{

line = j;

break;

}

}

if (line == -1)

continue;

cout << vars[i] << " = " << matrix[line][matrix[line].size() - 1];

for (int j = matrix[line].size() - 2; j >= 0; --j)

{

if (j == i)

continue;

if (matrix[line][j] != 0)

{

if (matrix[line][j] < 0)

cout << " + ";

else

cout << " - ";

cout << matrix[line][j] << vars[j];

}

}

cout << "\n";

}

}

//Rational a(1,2), b(1,4);

//Rational c = a / b;

//cout << c;

SetConsoleTextAttribute(h, 15);

}

***Rational.h***

#pragma once

#include <string>

#include <fstream>

#include <iostream>

#include <vector>

#include <iomanip>

#include <sstream>

using namespace std;

class Rational

{

public:

long long a, b; //a-числитель b-знаменатель//

Rational(long long chislit = 0, long long znamenat = 1) :a(chislit), b(znamenat)

{

a = chislit;

if (znamenat != 0)

b = znamenat;

else

{

cout << "Ne mogu podelit na ZERO\n";

exit(1);

}

}

Rational operator+ (Rational);

Rational operator- (Rational);

Rational operator\* (Rational);

Rational operator\* (int);

Rational operator/ (Rational);

Rational operator= (Rational);

long long Nod(long long, long long);

};

Rational rabs(Rational a);

***Rational.cpp***

#include "Ratinal.h"

Rational rabs(Rational a)

{

Rational r;

r.a = abs(a.a);

r.b = abs(a.b);

return r;

}

Rational Rational::operator- (Rational s)

{

Rational r;

r.a = a \* s.b - b \* s.a;

r.b = b \* s.b;

long long c = Nod(r.b, r.a);

r.a = r.a / c;

r.b = r.b / c;

return r;

}

Rational Rational::operator+ (Rational s)

{

Rational r;

r.a = a \* s.b + b \* s.a;

r.b = b \* s.b;

int c = Nod(r.b, r.a);

r.a = r.a / c;

r.b = r.b / c;

return r;

}

Rational Rational::operator/ (Rational s)

{

Rational r;

r.a = a \* s.b;

r.b = b \* s.a;

if (r.b == 0)

{

cout << "nelzia delit na Zero" << endl;

exit(1);

}

long long c = Nod(r.b, r.a);

r.a = r.a / c;

r.b = r.b / c;

return r;

}

Rational Rational::operator\* (int s)

{

Rational r;

r.a = a \* s;

r.b = b;

long long c = Nod(r.b, r.a);

r.a = r.a / c;

r.b = r.b / c;

return r;

}

Rational Rational::operator\* (Rational s)

{

Rational r;

r.a = a \* s.a;

r.b = b \* s.b;

long long c = Nod(r.b, r.a);

r.a = r.a / c;

r.b = r.b / c;

return r;

}

Rational Rational::operator= (Rational s)

{

a = s.a;

if (s.b)

b = s.b;

else

{

cout << "ne mogu delit na zero" << endl;

exit(1);

};

return \*this;

}

long long Rational::Nod(long long a, long long b)

{

if (a == 0 && b == 0)

throw "----";

if (b == 0)

return a;

if (a == 0)

return b;

if (a % b == 0)

return b;

if (b % a == 0)

return a;

if (a > b)

return Nod(abs(a % b), abs(b));

else

return Nod(abs(a), abs(b % a));

}