Лаботаторная работа 2

Код Юнит-тестов:

#include "pch.h"

#include "CppUnitTest.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Проверка работы на целых числах\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Проверка уравнения вида ax+by=e и cx+dy=f где a!=c и b!=d

TEST\_METHOD(iTestMethod0)

{

double a = 5, b = 0, c = 3, d = 0, e = 10, f = 18;

int q = 0;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (!((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (!((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

q = 1;

}

}

}

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и cx+dy=f => 1 пара решений, т.к. х=0

TEST\_METHOD(iTestMethod1\_10)

{

double a = 5, b = 0, c = 8, d = 10, e =0, f = 1;

int q = 0;

double y=0, x=0;

if (b == 0 && e == 0)

{

double k, n;

k = -c / d;

n = f / d;

}

if (x == 0 && y == 0.1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и cx+0y=0 => 1 пара решений, т.к. х=0

TEST\_METHOD(iTestMethod1\_11)

{

double a = 8, b = 10, c = 8, d = 0, e = 1, f = 0;

int q = 0;

double y=0, x=0;

if (d == 0 && f == 0)

{

double k, n;

k = -a / b;

n = e / b;

}

if (x == 0 && y == 0.1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=0 и cx+dy=f => 1 пара решений, т.к.y=0

TEST\_METHOD(iTestMethod1\_21)

{

double a = 0, b = 10, c = 10, d = 8, e = 0, f = 1;

int q = 0;

double y=0, x=0;

if (a == 0 && e == 0)

{

double k, n;

k = -d / c;

n = f / c;

}

if (x == 0.1 && y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+by=e и 0x+dy=0 => 1 пара решений, т.к.y=0

TEST\_METHOD(iTestMethod1\_22)

{

double a = 10, b = 8, c = 0, d = 8, e = 1, f = 0;

int q = 0;

double y=0, x=0;

if (c == 0 && f == 0)

{

double k, n;

k = -b / a;

n = e / a;

}

if (x == 0.1 && y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+by=e и cx+dy=f где a / b == c / d => решений нет

TEST\_METHOD(iTestMethod1\_30)

{

double a = 10, b = 5, c = 6, d = 3, e = 2, f = 2;

int q = 0;

double y, x;

if ((a / b == c / d))

{

double k, n;

k = -c / d;

n = f / d;

}

Assert::IsTrue(q == 1);

}

//Проверка нахождения пары решений

TEST\_METHOD(iTestMethod2)

{

double a = 1, b = -1, c = 3, d = 2, e = 7, f = 16;

int q = 0;

double y, x;

if ((a \* d - c \* b != 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0)))

{

y = (a \* f - c \* e) / (a \* d - c \* b);

x = (d \* e - b \* f) / (d \* a - b \* c);

}

if (x == 6 && y == -1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и cx+0y=f

TEST\_METHOD(iTestMethod3\_00)

{

double a = 0, b = 0, c = 3, d = 0, e = 0, f = 6;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и cx+0y=0

TEST\_METHOD(iTestMethod3\_01)

{

double a = 0, b = 0, c = 3, d = 0, e = 0, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=e и 0x+0y=0

TEST\_METHOD(iTestMethod3\_10)

{

double a = 3, b = 0, c = 0, d = 0, e = 6, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и 0x+0y=0

TEST\_METHOD(iTestMethod3\_11)

{

double a = 3, b = 0, c = 0, d = 0, e = 0, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=e и cx+0y=f где a=c, e=f

TEST\_METHOD(iTestMethod3\_2)

{

double a = 3, b = 0, c = 3, d = 0, e = 6, f = 6;

int q = 0;

double x;

if (b == 0 && d == 0)

{

if (e == 0)

x = f / c;

else if (f == 0)

x = e / a;

else

x = e / a;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и 0x+dy=f

TEST\_METHOD(iTestMethod4\_00)

{

double a = 0, b = 0, c = 0, d = 3, e = 0, f = 6;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y ==2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и 0x+dy=0

TEST\_METHOD(iTestMethod4\_01)

{

double a = 0, b = 0, c = 0, d = 3, e = 0, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y ==0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=e и 0x+0y=0

TEST\_METHOD(iTestMethod4\_10)

{

double a = 0, b = 3, c = 0, d = 0, e = 6, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=0 и 0x+0y=0

TEST\_METHOD(iTestMethod4\_11)

{

double a = 0, b = 3, c = 0, d = 0, e = 0, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=e и 0x+dy=f где b=d и e=f

TEST\_METHOD(iTestMethod4\_2)

{

double a = 0, b = 3, c = 0, d = 3, e = 6, f = 6;

int q = 0;

double y;

if (a == 0 && c == 0)

{

if (e == 0)

y = f / d;

else if (f == 0)

y = e / b;

else

y = e / b;

}

if (y == 2) q = 1;

Assert::IsTrue(q == 1);

}

TEST\_METHOD(iTestMethod5)

{

double a=0, b = 0, c = 0, d = 0, e = 0, f = 0;

int q = 0;

if ((a == 0) && (b == 0) && (c == 0) && (d == 0) && (e == 0) && (f == 0))

{

q = 1;

}

Assert::IsTrue(q == 1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Проверка работы на дробных числах числах\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Проверка уравнения вида ax+by=e и cx+dy=f где a!=c и b!=d

TEST\_METHOD(dTestMethod0)

{

double a = 0.5, b = 0, c = 0.3, d = 0, e = 0.10, f = 0.18;

int q = 0;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (!((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (!((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

q = 1;

}

}

}

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и cx+dy=f => 1 пара решений, т.к. х=0

TEST\_METHOD(dTestMethod1\_10)

{

double a =0.5, b = 0, c =0.8, d = 0.10, e = 0, f = 0.1;

int q = 0;

double y = 0, x = 0;

if (b == 0 && e == 0)

{

double k, n;

k = -c / d;

n = f / d;

}

if (x == 0 && y == 1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и cx+0y=0 => 1 пара решений, т.к. х=0

TEST\_METHOD(dTestMethod1\_11)

{

double a = 0.8, b =0.10, c = 0.8, d = 0, e = 0.1, f = 0;

int q = 0;

double y = 0, x = 0;

if (d == 0 && f == 0)

{

double k, n;

k = -a / b;

n = e / b;

}

if (x == 0 && y == 1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=0 и cx+dy=f => 1 пара решений, т.к.y=0

TEST\_METHOD(dTestMethod1\_21)

{

double a = 0, b = 0.10, c = 0.10, d = 0.8, e = 0, f = 0.1;

int q = 0;

double y = 0, x = 0;

if (a == 0 && e == 0)

{

double k, n;

k = -d / c;

n = f / c;

}

if (x == 1 && y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+by=e и 0x+dy=0 => 1 пара решений, т.к.y=0

TEST\_METHOD(dTestMethod1\_22)

{

double a = 0.10, b = 0.8, c = 0, d = 0.8, e = 0.1, f = 0;

int q = 0;

double y = 0, x = 0;

if (c == 0 && f == 0)

{

double k, n;

k = -b / a;

n = e / a;

}

if (x == 1 && y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+by=e и cx+dy=f где a / b == c / d => решений нет

TEST\_METHOD(dTestMethod1\_30)

{

double a = 10, b = 5, c = 6, d = 3, e = 2, f = 2;

int q = 0;

double y, x;

if ((a / b == c / d))

{

double k, n;

k = -c / d;

n = f / d;

}

Assert::IsTrue(q == 1);

}

//Проверка нахождения пары решений

TEST\_METHOD(dTestMethod2)

{

double a = 0.1, b = -0.1, c = 0.3, d = 0.2, e = 0.7, f = 0.16;

int q = 0;

double y, x;

if ((a \* d - c \* b != 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0)))

{

y = (a \* f - c \* e) / (a \* d - c \* b);

x = (d \* e - b \* f) / (d \* a - b \* c);

}

if (x == 0 && y == 1) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и cx+0y=f

TEST\_METHOD(dTestMethod3\_00)

{

double a = 0, b = 0, c = 0.3, d = 0, e = 0, f = 0.6;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и cx+0y=0

TEST\_METHOD(dTestMethod3\_01)

{

double a = 0, b = 0, c = 0.3, d = 0, e = 0, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=e и 0x+0y=0

TEST\_METHOD(dTestMethod3\_10)

{

double a = 0.3, b = 0, c = 0, d = 0, e = 0.6, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=0 и 0x+0y=0

TEST\_METHOD(dTestMethod3\_11)

{

double a = 0.3, b = 0, c = 0, d = 0, e = 0, f = 0;

int q = 0;

double x;

if (((a == 0 && b == 0 && e == 0 && c != 0 && d == 0) ||

(c == 0 && d == 0 && f == 0 && a != 0 && b == 0)))

{

if (a == 0)

x = f / c;

else if (c == 0)

x = e / a;

else if (e == 0 || f == 0)

x = 0;

}

if (x == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида ax+0y=e и cx+0y=f где a=c, e=f

TEST\_METHOD(dTestMethod3\_2)

{

double a = 0.3, b = 0, c = 0.3, d = 0, e = 0.6, f = 0.6;

int q = 0;

double x;

if (b == 0 && d == 0)

{

if (e == 0)

x = f / c;

else if (f == 0)

x = e / a;

else

x = e / a;

}

if (x == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и 0x+dy=f

TEST\_METHOD(dTestMethod4\_00)

{

double a = 0, b = 0, c = 0, d = 0.3, e = 0, f = 0.6;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+0y=0 и 0x+dy=0

TEST\_METHOD(dTestMethod4\_01)

{

double a = 0, b = 0, c = 0, d = 0.3, e = 0, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=e и 0x+0y=0

TEST\_METHOD(dTestMethod4\_10)

{

double a = 0, b = 0.3, c = 0, d = 0, e = 0.6, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 2) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=0 и 0x+0y=0

TEST\_METHOD(dTestMethod4\_11)

{

double a = 0, b = 0.3, c = 0, d = 0, e = 0, f = 0;

int q = 0;

double y;

if (((a \* d - c \* b == 0) && ((e \* d - b \* f != 0) || (a \* f - c \* e != 0))) ||

(a == 0 && c == 0 && e / b != f / d) ||

(b == 0 && d == 0 && e / a != f / c) ||

(a == 0 && b == 0 && c == 0 && d == 0 && (e / f > 0)))

{

if (((a == 0 && b == 0 && e == 0 && d != 0 && c == 0) ||

(c == 0 && d == 0 && f == 0 && b != 0 && a == 0)))

{

if (b == 0)

y = f / d;

else if (d == 0)

y = e / b;

else if (e == 0 || f == 0)

y = 0;

}

}

if (y == 0) q = 1;

Assert::IsTrue(q == 1);

}

//Проверка уравнения вида 0x+by=e и 0x+dy=f где b=d и e=f

TEST\_METHOD(dTestMethod4\_2)

{

double a = 0, b = 3, c = 0, d = 0.3, e = 6, f = 0.6;

int q = 0;

double y;

if (a == 0 && c == 0)

{

if (e == 0)

y = f / d;

else if (f == 0)

y = e / b;

else

y = e / b;

}

if (y == 2) q = 1;

Assert::IsTrue(q == 1);

}

TEST\_METHOD(dTestMethod5)

{

double a = 0, b = 0, c = 0, d = 0, e = 0, f = 0;

int q = 0;

if ((a == 0) && (b == 0) && (c == 0) && (d == 0) && (e == 0) && (f == 0))

{

q = 1;

}

Assert::IsTrue(q == 1);

}

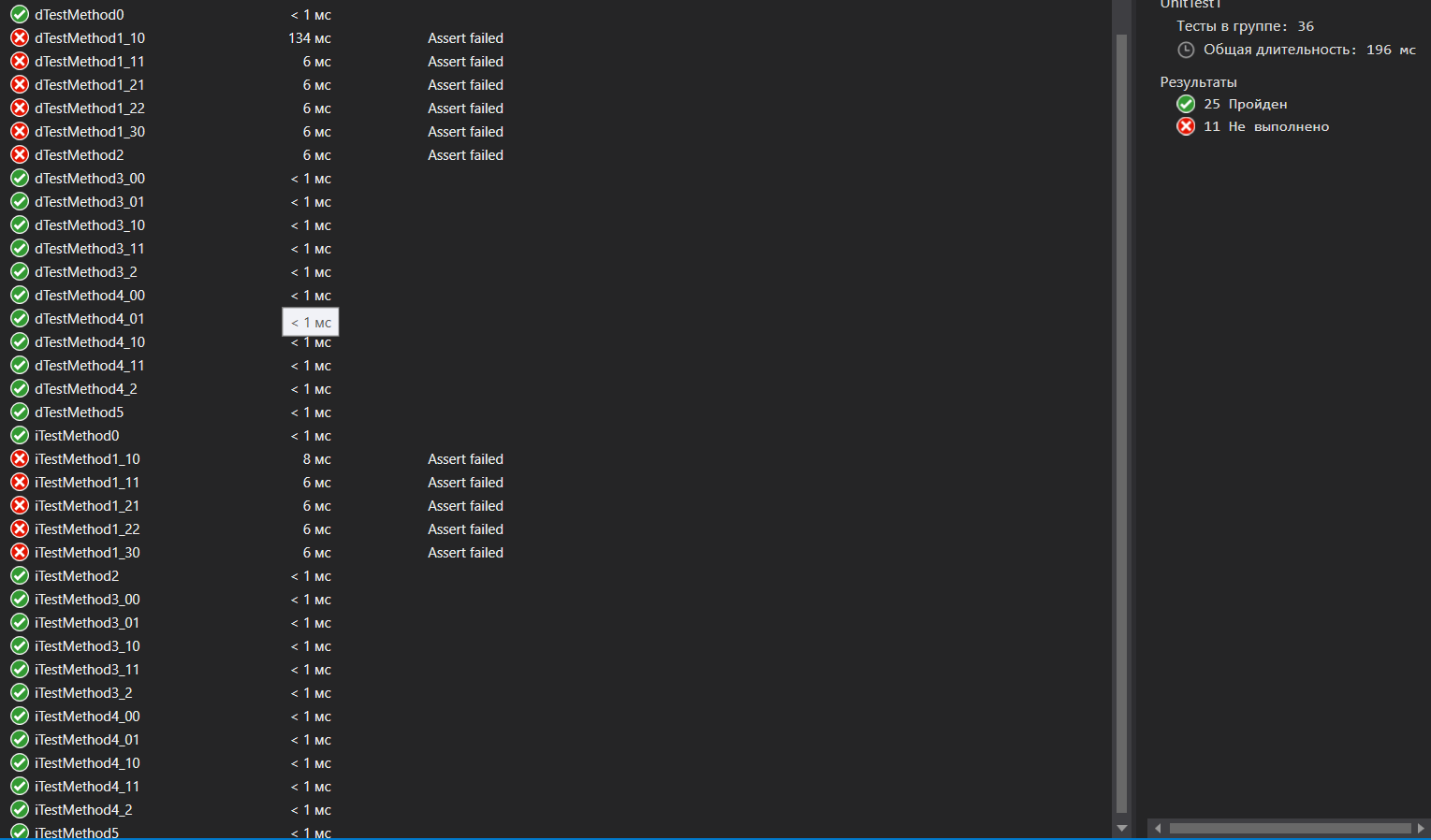
};

}

**Результат работы:**

**Имя тестов, начинающиеся с I – работа с целыми числами**

**D – работа с дробными числами**



Проверку не прошел вывод на экран 1 и коэффициентов прямой. Это связано с неправильной логикой кода. Система будет иметь решение вида y=kx+n при условии: a/c==b/d==e/f.

Так же проверку не прошел тест, который ищет пару решений уравнения при работе с дробными числами. Это связано с особенностями машинной арифметики.