Министерство образования и науки РФ

СевГУ

Кафедра ИС

Отчёт

по лабораторной работе №7

«**Изучение алгоритмов для нахождения НОД целых чисел**»

Выполнил:

ст.гр. ИТ-12д

Воронин И.Ю.

Проверил:

Заикина Е.Н.

Севастополь

2015

**1. ЦЕЛЬ РАБОТЫ**

1. Познакомиться с алгоритмами для вычисления НОД;

2. Изучить основные правила построения алгоритмов вычисления НОД чисел;

3. Рассмотреть алгоритм Эвклида для нахождения НОД чисел.

**2. ВАРИАНТЫ ЗАДАНИЙ**

Вариант №4

Даны натуральные а и b, не равные 0. Найти d = НОД (a,b) и такие целые x и y, что d = a\*x + b\*y.

**3.ТЕКСТ ПРОГРАММЫ**

#include <iostream>

using namespace std;

int main(){

setlocale(LC\_ALL,"Russian");

int a,b,d,x,y;

// натуральные числа не равные нулю

do{

cout<<"Введите a: ",cin>>a;

cout<<"Введите b: ",cin>>b;

}

while(!(a>0 && b>0));

//НОД

for(d = (a>b) ? b: a; a%d || b%d; d--);

cout<<'\n'<<"НОД чисел "<< a <<" и "<< b <<" равен "<<d<<'.'<<'\n';

//второе задание

for (x = 0; x>=-32000; x--)

for (y = 0; y<=32000; y++)

if(a\*x+b\*y==d) {

cout<<'\n'<<"Соотношение чисел: "<<'\n'

<< a <<" \* ("<< x <<") + "<< b <<" \* ("<<y<<") = "<<d<<"."<<'\n';

return 0;

}

for (x = 0; x<=32000; x++){

for (y = 0; y>=-32000; y--)

if(a\*x+b\*y==d) {

cout<<'\n'<<"Соотношение чисел: "<<'\n'

<< a <<" \* ("<< x <<") + "<< b <<" \* ("<<y<<") = "<<d<<"."<<'\n';

return 0;

} }}

**4.РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

Стандартный запуск программы выглядит следующим образом (рис.4.1):

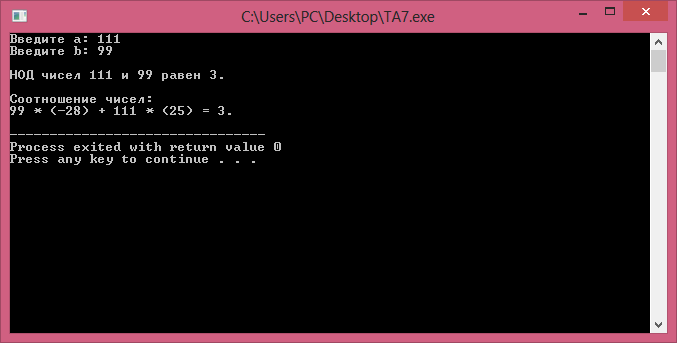


Рисунок 4.1-Тест программы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | НОД | X | Y |
| 99 | 102 | 3 | -1 | 1 |
| 7 | 11 | 1 | -3 | 2 |
| 33 | 33 | 33 | 0 | 1 |
| 1000 | 750 | 250 | 1 | -1 |
| 1234 | 4321 | 1 | -1082 | 309 |
| 777 | 616 | 7 | 23 | -29 |
| 1 | 300 | 1 | -299 | 1 |

**ВЫВОДЫ**

В данной работе были изучены алгоритмы поиска НОД и НОК на примере алгоритма Эвклида. Была написана программа на языке С++, которая выполняла поиск НОД и разбиение его на сумму произведение исходных чисел на целые коэффициенты. Поиск НОД занимает одну строчку кода, однако разбиение представлено в виде двух циклов проверки всех вариантов. Программа была протестирована на различных значениях.