

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий

Отчет № 5

по дисциплине «Информатика»

на тему: «Циклы while, do-while, for. Операторы перехода и выхода из цикла.
Классификация циклических процессов. Алгоритм Евклида. Наименьшее общее кратное.
Вычисление степеней двойки»

Выполнил:

студент группы 3530902/90001

_____ Непушкин Сергей Александрович

Проверил:

Доцент ВШКФСИУ

_____ Теплова Наталья Витальевна

Санкт-Петербург

2019 г.

Оглавление	
1.Задание.....	3
2.Доказательство алгоритма.....	3
3.Блок-схема алгоритма.....	4
4.Текст кода.....	5
5.Пример работы программы.....	6
6.Решение варианта.....	7

1. Задание.

Доказать применимость алгоритма Евклида.

Написать программу нахождения наибольшего общего делителя для двух положительных целых чисел. Написать алгоритм нахождения наименьшего общего кратного для двух положительных целых чисел.

Вычислить НОД и НОК для чисел a и b своего варианта (в приложенном файле).

Для своего варианта вычислить среднее число итераций алгоритма для $a=1..b$, $b=b$ (всего b внешних циклов в программе).

2. Доказательство алгоритма

$$1. r = b \bmod a$$

$$b = aq + r$$

$$2. m = a \bmod n$$

$$a = nk + m$$

$$b = (nk + m)q + r$$

$$3. d = n \bmod e$$

$$n = et + d$$

$$b = ((et + d)k + m)q + r$$

$$a = (et + d)k + m$$

$$4. p = e \bmod f$$

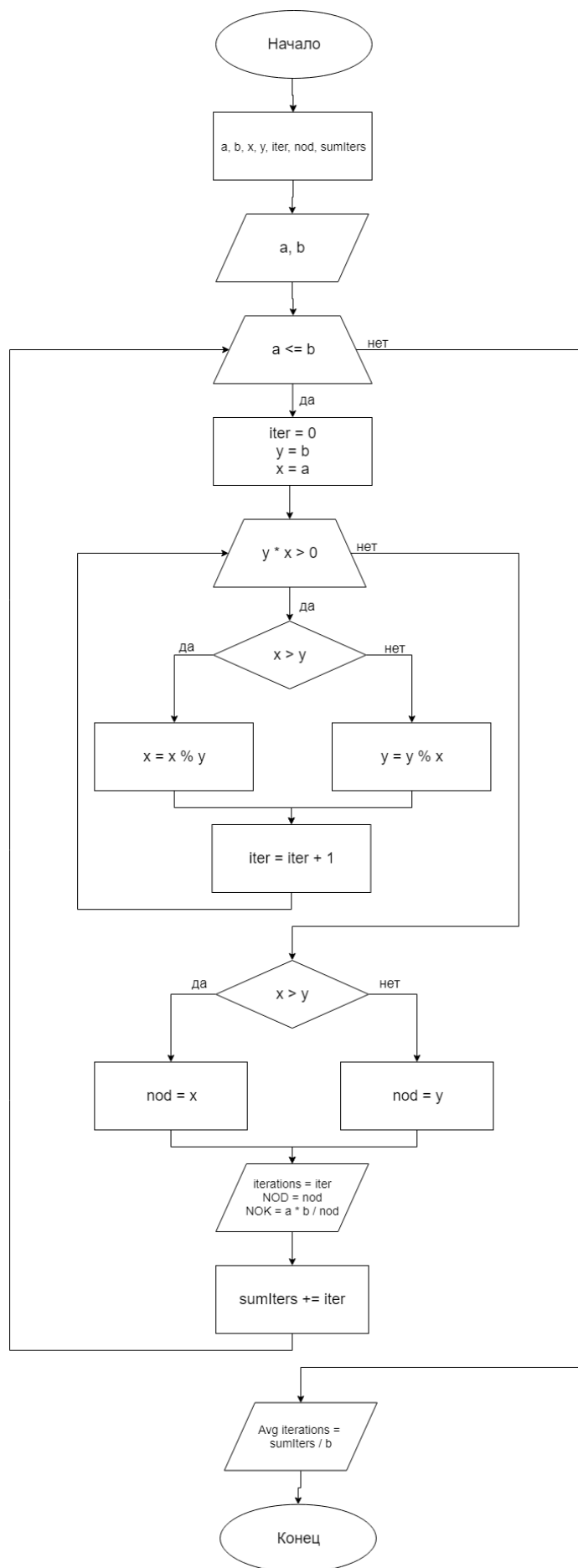
$$e = fc + p$$

$$b = (((fc + p)t + d)k + m)q + r$$

$$a = ((fc + p)t + d)k + m$$

Таким образом, мы доказали, что с помощью данного алгоритма изначальные числа можно разложить на простые множители и, как следствие, найти наибольший общий делитель.

3.Блок-схема алгоритма



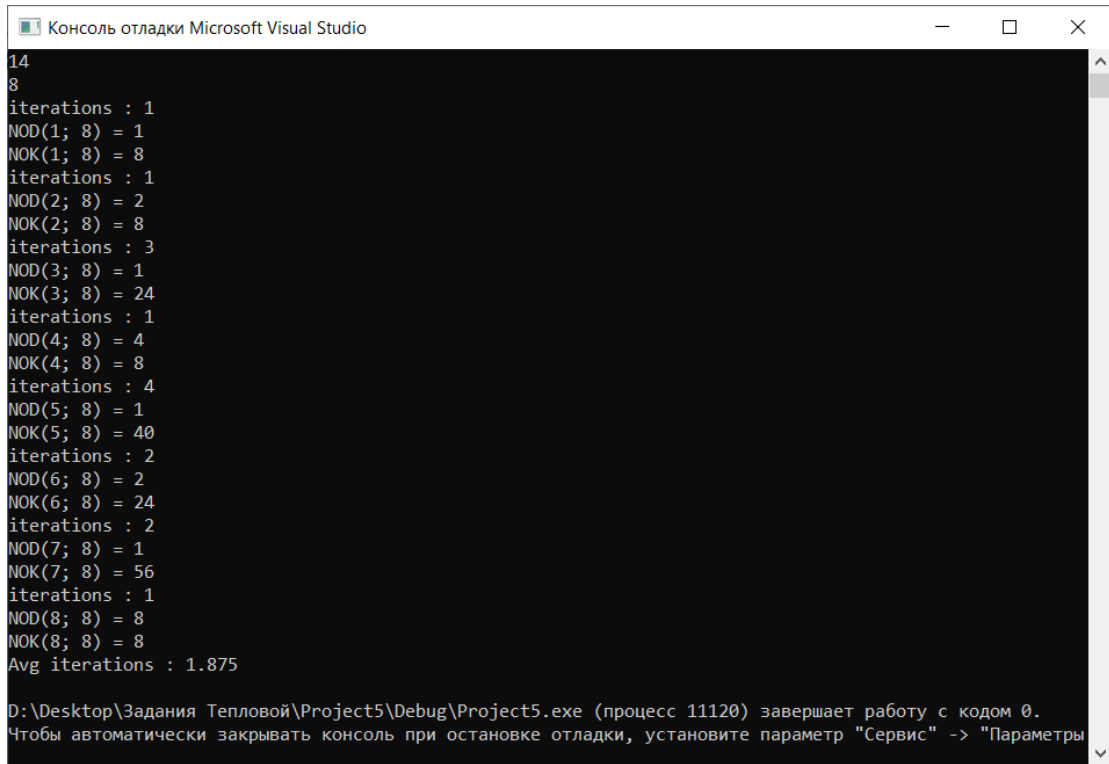
4.Текст кода.

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
using namespace std;
int main()
{
    int a;
    int b;
    int x = 0;
    int y = 0;
    int iter = 0;
    int nod = 0;
    int sumIters = 0;
    scanf("%d %d", &a, &b);

    for (a = 1; a <= b; a++)
    {
        iter = 0;
        y = b;
        x = a;
        while (x * y > 0)
        {
            if (x > y)
            {
                x = x % y;
            }
            else
            {
                y = y % x;
            }
            iter++;
        }
        nod = x > y ? x : y;
        printf("iterations : %d\n", iter);
        printf("NOD(%d; %d) = %d\n", a, b, nod);
        printf("NOK(%d; %d) = %d\n", a, b, a * b / nod);
        sumIters += iter;
    }
    printf("Avg iterations : %5.3f\n", (float)sumIters / b);

    return 0;
}
```

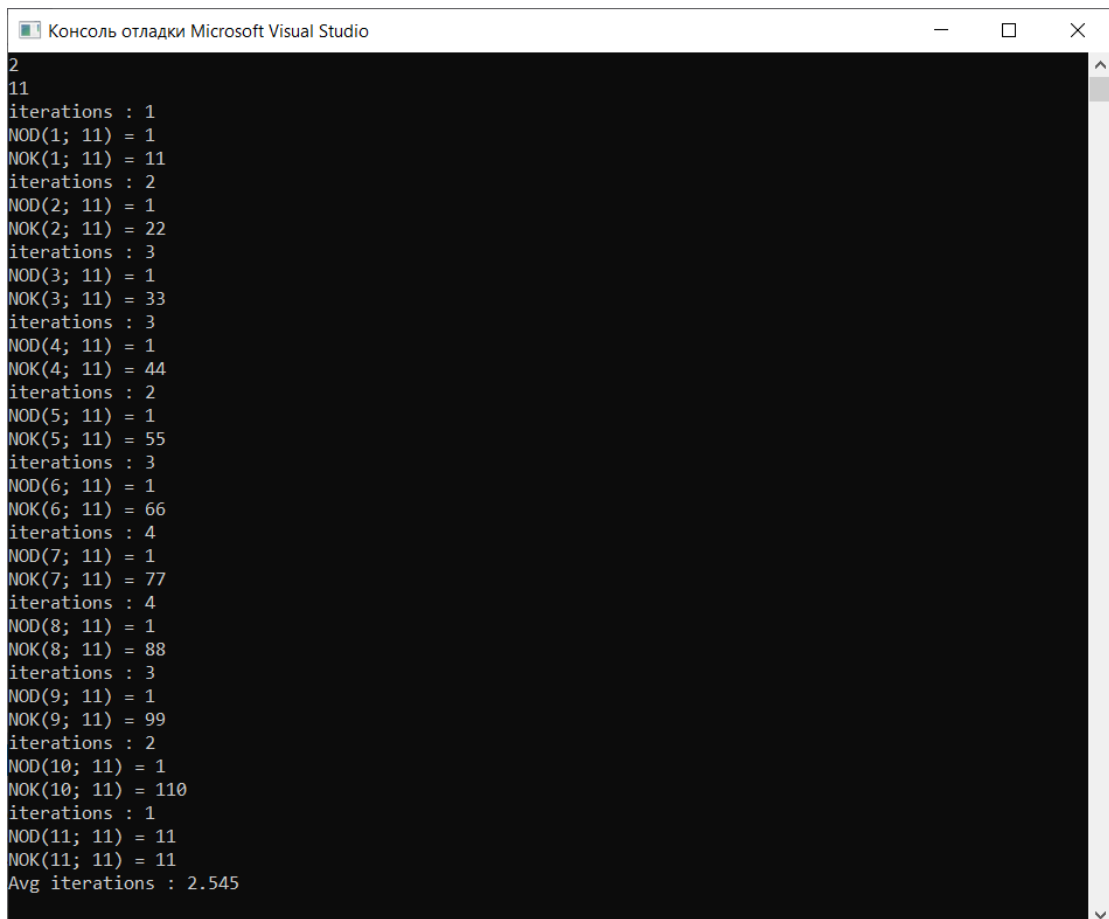
5.Пример работы программы



```
Консоль отладки Microsoft Visual Studio

14
8
iterations : 1
NOD(1; 8) = 1
NOK(1; 8) = 8
iterations : 1
NOD(2; 8) = 2
NOK(2; 8) = 8
iterations : 3
NOD(3; 8) = 1
NOK(3; 8) = 24
iterations : 1
NOD(4; 8) = 4
NOK(4; 8) = 8
iterations : 4
NOD(5; 8) = 1
NOK(5; 8) = 40
iterations : 2
NOD(6; 8) = 2
NOK(6; 8) = 24
iterations : 2
NOD(7; 8) = 1
NOK(7; 8) = 56
iterations : 1
NOD(8; 8) = 8
NOK(8; 8) = 8
Avg iterations : 1.875

D:\Desktop\Задания Тепловой\Project5\Debug\Project5.exe (процесс 11120) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр "Сервис" -> "Параметры"
```



```
Консоль отладки Microsoft Visual Studio

2
11
iterations : 1
NOD(1; 11) = 1
NOK(1; 11) = 11
iterations : 2
NOD(2; 11) = 1
NOK(2; 11) = 22
iterations : 3
NOD(3; 11) = 1
NOK(3; 11) = 33
iterations : 3
NOD(4; 11) = 1
NOK(4; 11) = 44
iterations : 2
NOD(5; 11) = 1
NOK(5; 11) = 55
iterations : 3
NOD(6; 11) = 1
NOK(6; 11) = 66
iterations : 4
NOD(7; 11) = 1
NOK(7; 11) = 77
iterations : 4
NOD(8; 11) = 1
NOK(8; 11) = 88
iterations : 3
NOD(9; 11) = 1
NOK(9; 11) = 99
iterations : 2
NOD(10; 11) = 1
NOK(10; 11) = 110
iterations : 1
NOD(11; 11) = 11
NOK(11; 11) = 11
Avg iterations : 2.545
```

6.Решение варианта № 2

```
Консоль отладки Microsoft Visual Studio
75
180
iterations : 1
NOD(1; 180) = 1
NOK(1; 180) = 180
iterations : 1
NOD(2; 180) = 2
NOK(2; 180) = 180
iterations : 1
NOD(3; 180) = 3
NOK(3; 180) = 180
iterations : 1
NOD(4; 180) = 4
NOK(4; 180) = 180
iterations : 1
NOD(5; 180) = 5
NOK(5; 180) = 180
iterations : 1
NOD(6; 180) = 6
NOK(6; 180) = 180
iterations : 4
NOD(7; 180) = 1
NOK(7; 180) = 1260
iterations : 2
NOD(8; 180) = 4
NOK(8; 180) = 360
iterations : 1
NOD(9; 180) = 9
NOK(9; 180) = 180
iterations : 1
NOD(10; 180) = 10
NOK(10; 180) = 180
iterations : 4
NOD(11; 180) = 1
NOK(11; 180) = 1980
iterations : 1
NOD(12; 180) = 12
NOK(12; 180) = 180
iterations : 4
NOD(13; 180) = 1
NOK(13; 180) = 2340
iterations : 3
NOD(14; 180) = 2
NOK(14; 180) = 1260
iterations : 1
```

```
Выбрать Консоль отладки Microsoft Visual Studio
NOD(74; 180) = 2
NOK(74; 180) = 6660
iterations : 3
NOD(75; 180) = 15
NOK(75; 180) = 900
iterations : 5
NOD(76; 180) = 4
NOK(76; 180) = 3420
iterations : 4
NOD(77; 180) = 1
NOK(77; 180) = 13860
iterations : 3
NOD(78; 180) = 6
NOK(78; 180) = 2340
iterations : 6
NOD(79; 180) = 1
NOK(79; 180) = 14220
iterations : 2
NOD(80; 180) = 20
NOK(80; 180) = 720
iterations : 3
NOD(81; 180) = 9
NOK(81; 180) = 1620
iterations : 3
NOD(82; 180) = 2
NOK(82; 180) = 7380
iterations : 4
NOD(83; 180) = 1
NOK(83; 180) = 14940
iterations : 2
NOD(84; 180) = 12
NOK(84; 180) = 1260
iterations : 3
NOD(85; 180) = 5
NOK(85; 180) = 3060
iterations : 4
```

```
Консоль отладки Microsoft Visual Studio
NOD(171; 180) = 9
NOK(171; 180) = 3420
iterations : 3
NOD(172; 180) = 4
NOK(172; 180) = 7740
iterations : 5
NOD(173; 180) = 1
NOK(173; 180) = 31140
iterations : 2
NOD(174; 180) = 6
NOK(174; 180) = 5220
iterations : 2
NOD(175; 180) = 5
NOK(175; 180) = 6300
iterations : 2
NOD(176; 180) = 4
NOK(176; 180) = 7920
iterations : 2
NOD(177; 180) = 3
NOK(177; 180) = 10620
iterations : 2
NOD(178; 180) = 2
NOK(178; 180) = 16020
iterations : 2
NOD(179; 180) = 1
NOK(179; 180) = 32220
iterations : 1
NOD(180; 180) = 180
NOK(180; 180) = 180
Avg iterations : 3.717
```