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.Докозательство 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.

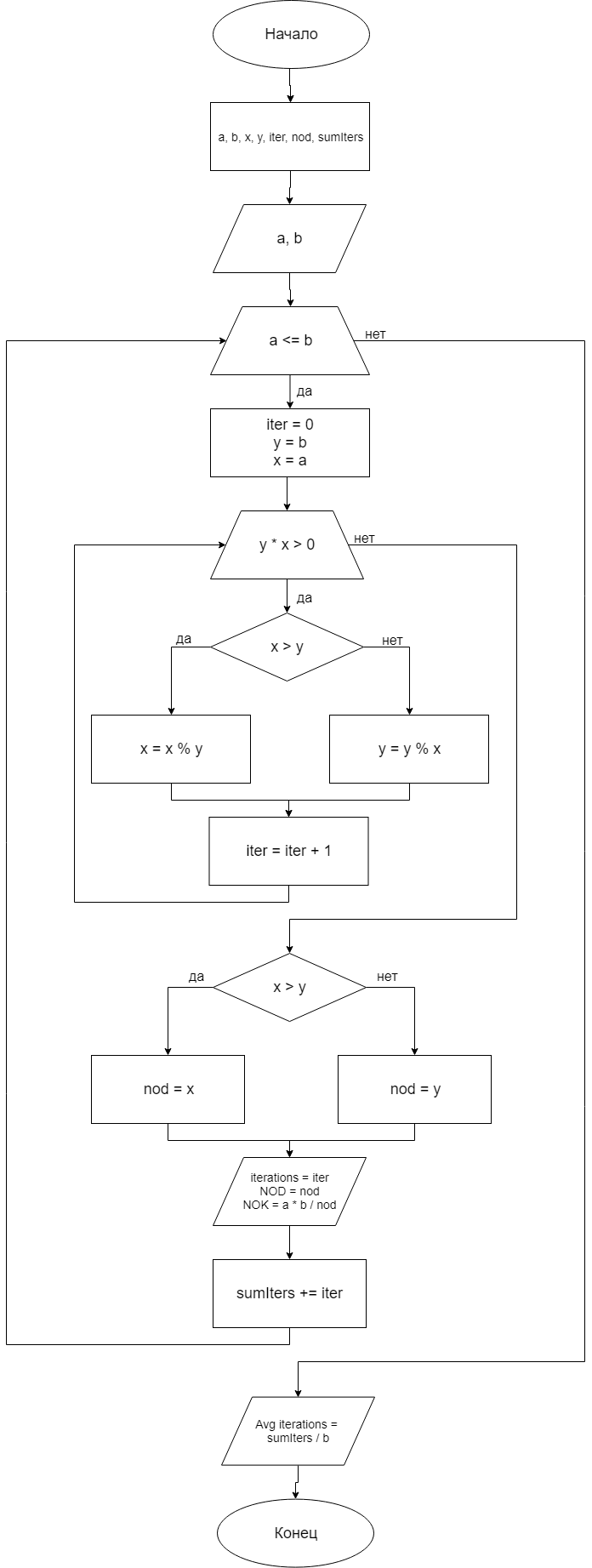
Вычислить НОД и НОК для чисел a и b своего варианта (в приложенном файле).

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 mod a  
b = aq + r  
  
2. m = a mod n  
a = nk + m  
b = (nk + m)q + r  
  
3. d = n mod e  
n = et + d  
b = ((et + d)k +m)q + r  
a = (et + d)k +m  
  
4. p = e mod 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;

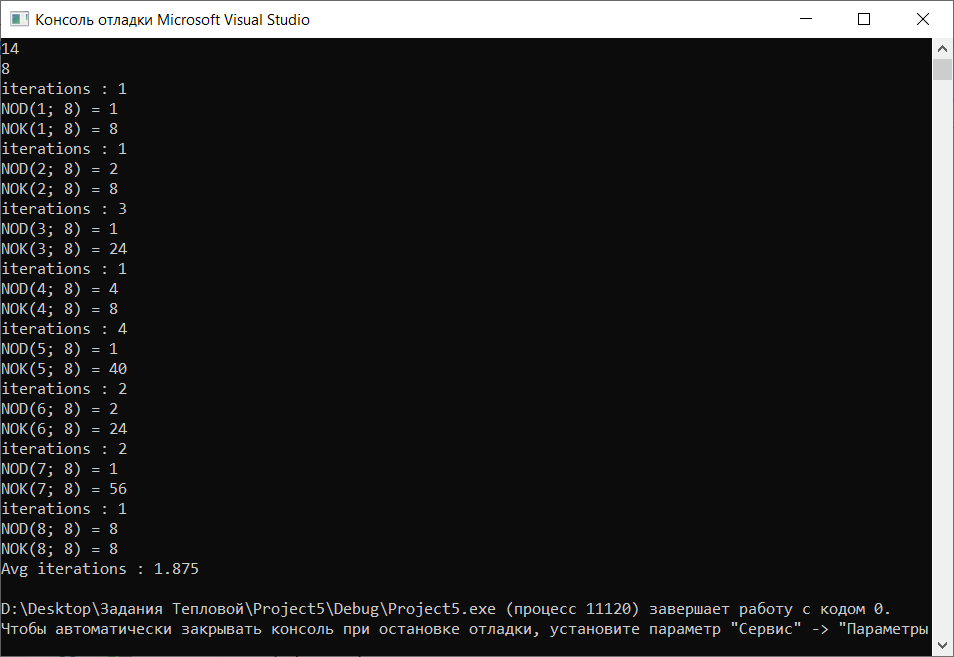
}

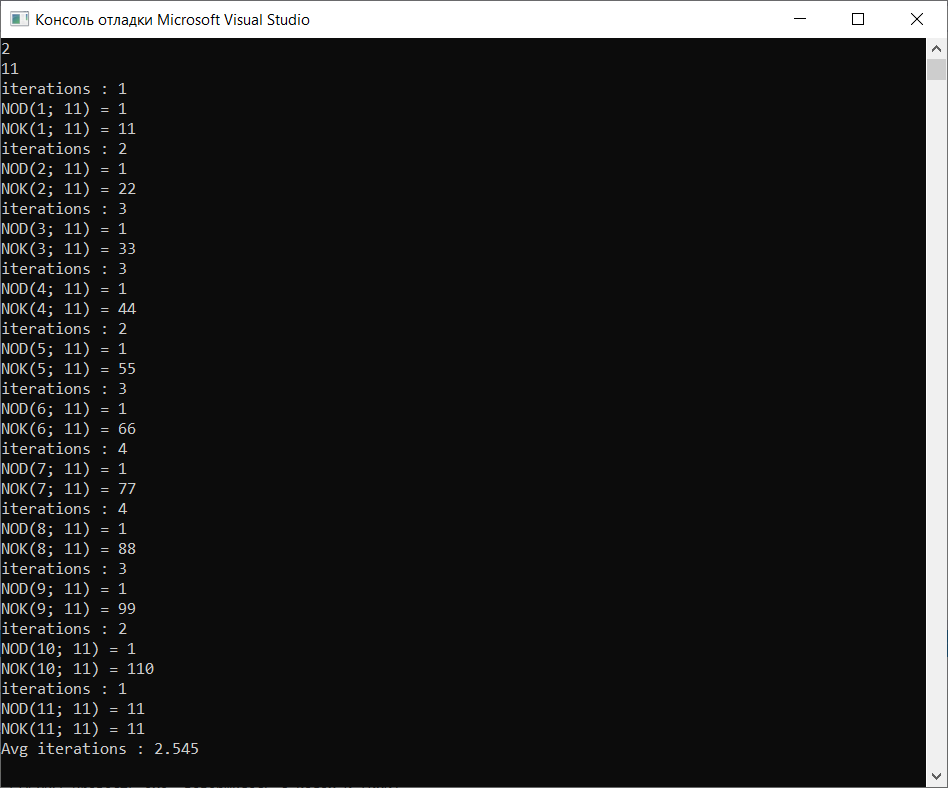
printf("Avg iterations : %5.3f\n", (float)sumIters / b);

return 0;

}

**5.Пример работы программы**





**6.Решение варианта № 2**

