МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Отчет № 10

по дисциплине «Информатика»

на тему: « Работа с массивами. Сортировки массивов: простыми включениями, простым обменом (метод «пузырька»).»

Выполнил:

студент группы 3530902/90001 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Непушкин Сергей Александрович

Проверил:

Доцент ВШКФСиУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Теплова Наталья Витальевна

Санкт-Петербург

2019 г.

Оглавление

1.Задание………………………………………………………………………………………….3

2.Блок-схема алгоритма…………………………………………………………………………4

3.Текст кода………………………………………………………………………………………5

4.Пример работы программы……………………………………………………………………8

5.Вывод…………………………………………………………………………………………...11

**1. Задание.**

Решить задачу Леонардо Фибоначчи: "Сколько пар кроликов можно получить от одной пары кроликов в год, если каждая пара ежемесячно дает еще одну пару приплода, каждая новая пара становится способной к размножению в возрасте одного месяца и в течение года кролики не умирают." Использовать два алгоритма: 1) с рекурсией 2) без рекурсии

Пользователь вводит: n=количество месяцев

Вывести на экран для каждого шага:

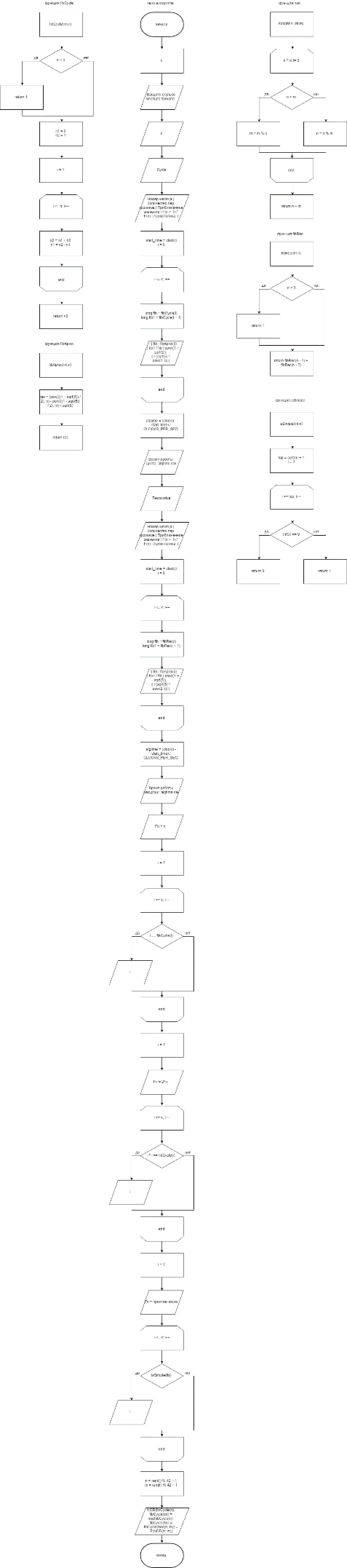
номер месяца, количество пар кроликов, приближенное значение, значение Fn+1/Fn, приближенное значение номера числа Фибоначчи

Сравнить с асимптотикой при больших n.

 Найти все n, для которых а) Fn=n б) Fn=n2 в) Fn - простое

Найти НОД для двух чисел Фибоначчи с номерами m и n и показать, что НОД (Fn, Fm) = НОД (n,m)

**2.Блок-схема алгоритма**

****

**3.Текст кода.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <math.h>

#include <time.h>

int nod(int n, int m)

{

while (n\*m != 0)

{

if (n < m) m = m % n;

else n = n % m;

}

return n + m;

}

int fibRec(int n)

{

if (n < 3) return 1;

return fibRec(n - 1) + fibRec(n - 2);

}

int fibCycle(int n)

{

if (n < 3) return 1;

int n1 = 0, n2 = 1;

for (int i = 1; i < n; i++)

{

n2 = n1 + n2;

n1 = n2 - n1;

}

return n2;

}

int isSimple(int n)

{

int top = sqrt(n) + 1;

for (int i = 2; i <= top; i++)

{

if ((n%i) == 0)

{

return 0;

}

}

return 1;

}

int fibAprox(int n)

{

double res = 0;

res = (pow(((1 + sqrt(5)) / 2), n) - pow(((1 - sqrt(5)) / 2), n)) / sqrt(5);

return (int)res;

}

double fibNumber(int n)

{

double res = 0;

return res;

}

//1 1 2 3 5 8 13 21

int main()

{

srand(time(0));

setlocale(LC\_ALL, "Russian");

int k;

printf("Введите сколько месяцев прошло: ");

scanf("%d", &k);

printf("Cycle\n");

printf("Номер месяца | Количество пар кроликов | Приближенное значение | F(n + 1) / F(n) | Асимптотика |\n");

unsigned int start\_time;

start\_time = clock();

for (int i = 1; i <= k; i++)

{

long fib = fibCycle(i);

long fib1 = fibCycle(i + 1);

printf(

"%13d| %24d| %22d| %16.6lf| %13.2lf|\n",

i,

fib,

fibAprox(i),

(double)fib1 / fib,

(double)pow((1 + sqrt(5)), i) / (sqrt(5) \* pow(2, i))

);

}

double algtime = (double)(clock() - start\_time) / CLOCKS\_PER\_SEC;

printf("Время работы цикла: %lf сек\n", algtime);

printf("\nRecoursive\n");

printf("Номер месяца | Количество пар кроликов | Приближенное значение | F(n + 1) / F(n) | Асимптотика |\n");

start\_time = clock();

for (int i = 1; i <= k; i++)

{

long fib = fibRec(i);

long fib1 = fibRec(i + 1);

printf(

"%13d| %24d| %22d| %16.6lf| %13.2lf|\n",

i,

fib,

fibAprox(i),

(double)fib1 / fib,

(double)pow((1 + sqrt(5)), i) / (sqrt(5) \* pow(2, i))

);

}

algtime = (double)(clock() - start\_time) / CLOCKS\_PER\_SEC;

printf("Время работы рекурсии: %lf сек\n", algtime);

printf("Fn = n: \n");

for (int i = 1; i <= k; i++)

{

if (i == fibCycle(i))

{

printf("%d \n", i);

}

}

printf("\n");

printf("Fn = n^2: \n");

for (int i = 1; i <= k; i++)

{

if (i \* i == fibCycle(i))

{

printf("%d \n", i);

}

}

printf("\n");

printf("Fn - простое число: \n");

for (int i = 1; i <= k; i++)

{

int fib = fibCycle(i);

if (isSimple(fib))

{

printf("%d \n", fib);

}

}

printf("\n");

int n = rand() % 42 + 1;

int m = rand() % 42 + 1;

printf(

"NOD(%d; %d) = %d = %d = F(NOD(%d; %d))\n",

fibCycle(n), fibCycle(m),

nod(fibCycle(n), fibCycle(m)),

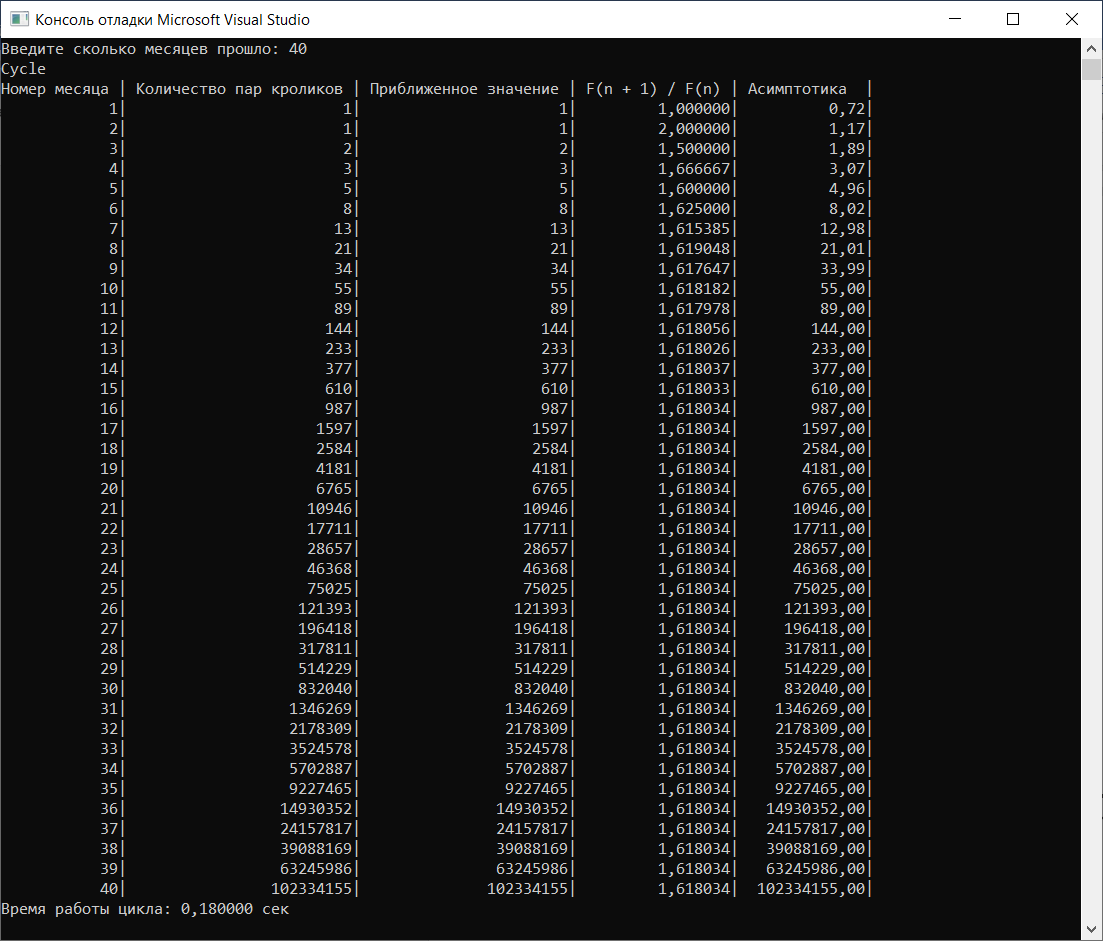
fibCycle(nod(n, m)),

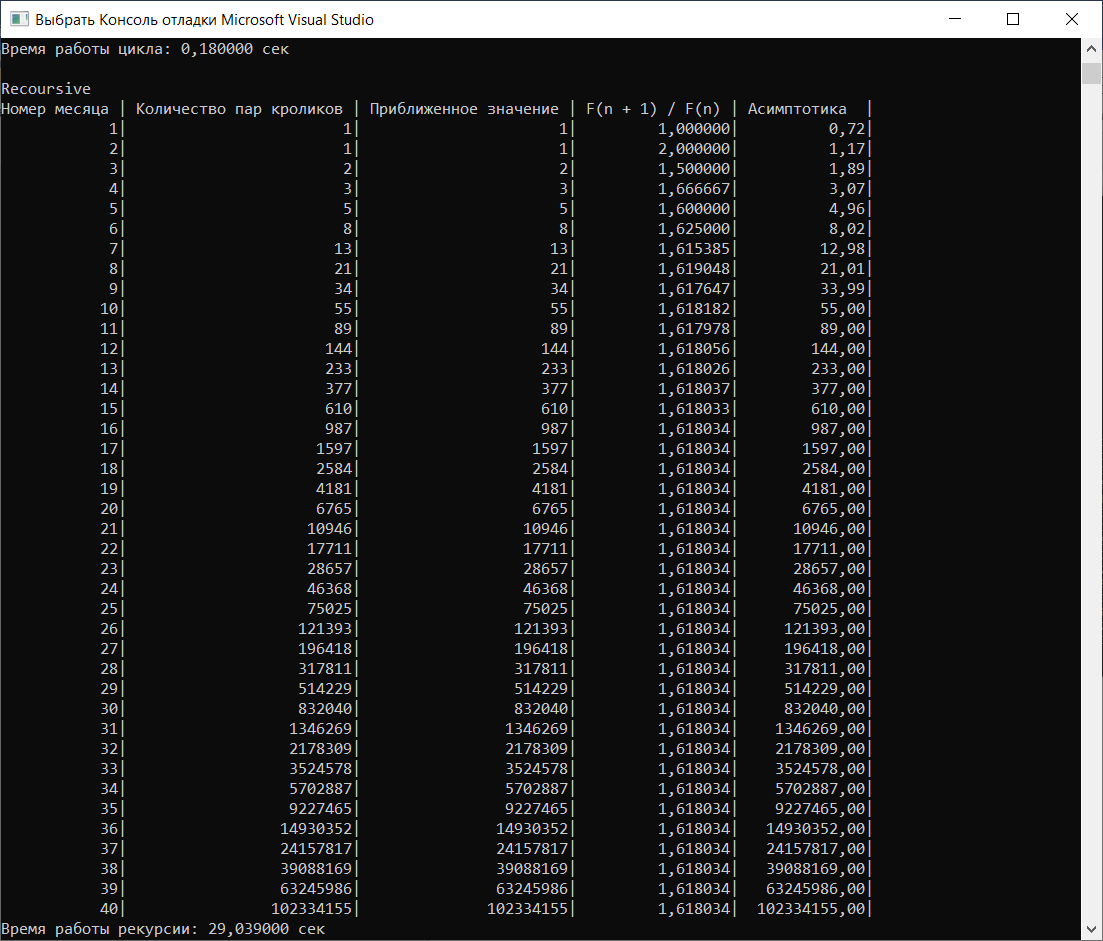
n, m);

return 0;

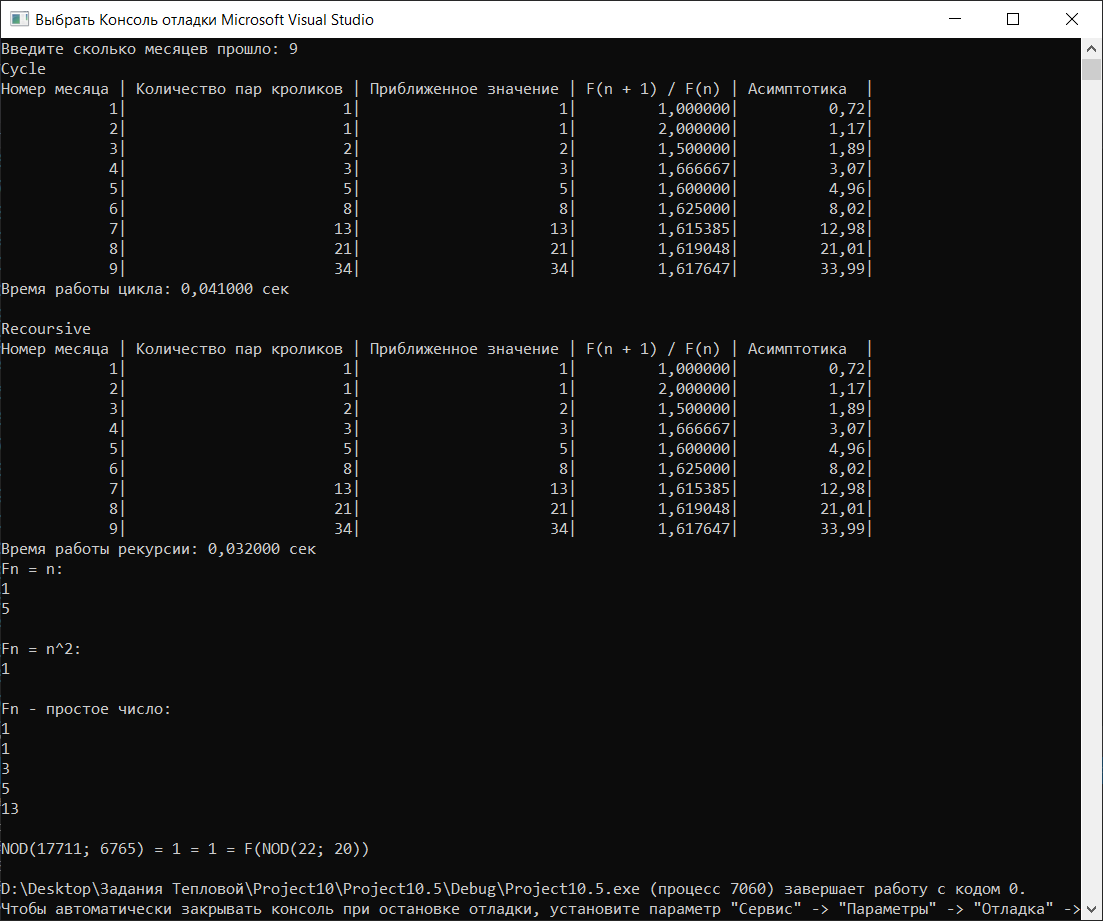
}

**4.Пример работы программы**









**5. Вывод**

Рекурсия при малом количестве итераций работает быстрее цикла, но при достаточно большом количестве итераций (даже при 40 и более) цикл считает числа Фибоначи гораздо быстрее рекурсии.