*Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет***  ***имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: Информатики и систем управления

КАФЕДРА: Компьютерные системы и сети

**Отчет**

**по усложнённой лабораторной работе №** 3

**Название лабораторной работы:** Программирование функций**.**

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент гр. ИУ6-22 |  |  | | Боярских Никита |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |
| Преподаватель |  | |  | Черноусова Т. Г. |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |

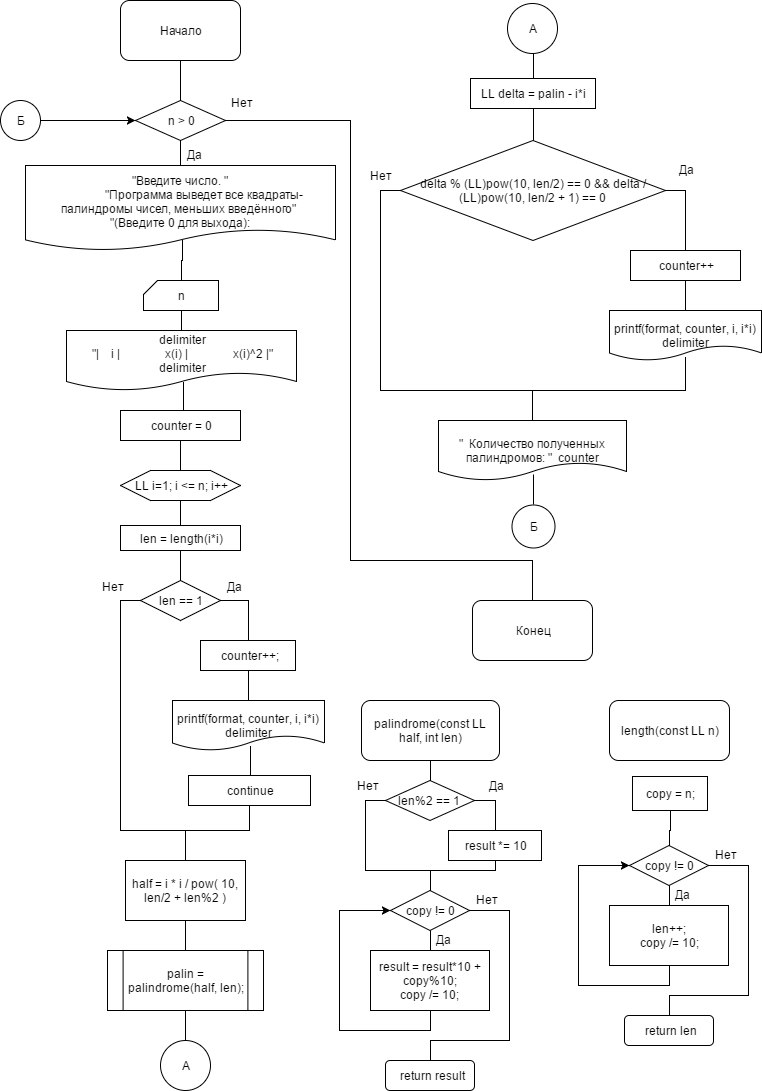
**Вариант 1**

Москва, 2017

**Задание:**

Написать программу, определяющую количество целых чисел ≤n (массив чисел n вводится с клавиатуры), квадрат которых является палиндромом. Например: 262 – 676. Вывести на печать числа, квадраты и их количество.

Для поставленной задачи написан алгоритм (схема алгоритма представлена на рис. 1)



*Рисунок 1 - схема алгоритма программы, находящей палиндромы*

**Код программы, находящей палиндромы**

#include <iostream>

#include <vector>

#include <cmath>

#include <cstdio>

typedef long long LL;

int length(const LL n)

{

int len = 0;

LL copy = n;

while( copy != 0 ) {

len++;

copy /= 10;

}

return len;

}

LL palindrome(const LL half, int len)

{

LL copy = half;

LL result = half;

if( len%2 == 1 ) result \*= 10;

while( copy != 0 ) {

result = result\*10 + copy%10;

copy /= 10;

}

return result;

}

int main(int argc, char \*\*argv)

{

using std::cout;

using std::cin;

using std::endl;

LL n;

while( cout <<"Введите число. "<<

"Программа выведет все квадраты-палиндромы чисел, меньших введённого"<< endl <<

"(Введите 0 для выхода): ",

cin >> n, n > 0 ) {

const char delimiter[] = "|------+--------------------+----------------------|";

const char format[] = "| %4d | %18lld | %20lld |";

cout << delimiter << endl <<"| i | x(i) | x(i)^2 |"<< endl << delimiter << endl;

int counter = 0;

for( LL i=1; i <= n; i++ ) {

int len = length(i\*i);

if( len == 1 ) {

counter++;

printf(format, counter, i, i\*i);

cout << endl << delimiter << endl;

continue;

}

LL half = i \* i / pow( 10, len/2 + len%2 );

LL palin = palindrome(half, len);

const LL delta = palin - i\*i;

(Продолжение кода на следующей странице)

(Продолжение кода)

if( delta % (LL)pow(10, len/2) == 0 && delta / (LL)pow(10, len/2 + 1) == 0 ) {

counter++;

printf(format, counter, i, i\*i);

cout<<endl<<delimiter<<endl;

}

}

cout <<"Количество полученных палиндромов: "<< counter << endl;

}

return 0;

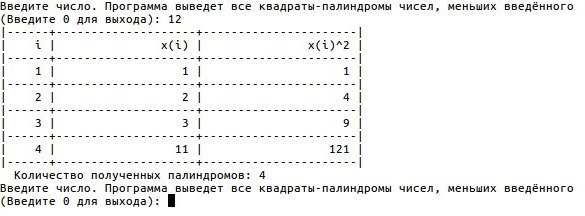
}

Проведено тестирование программы (см. табл. 1)

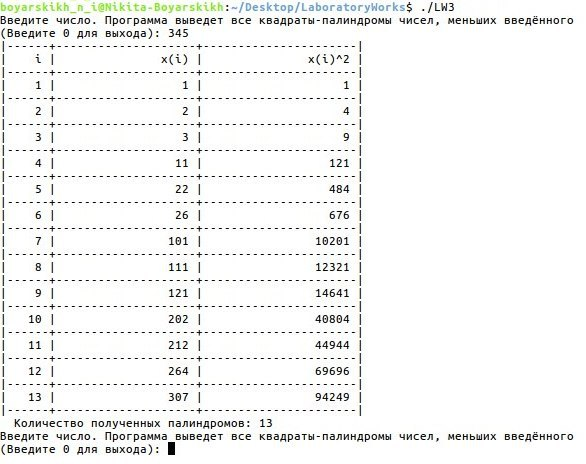
Таблица 1 – результаты тестирования программы, находящей палиндромы:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Исходные данные | Ожидаемый результат | Полученный результат |
| 1 | 12 | 4 | 4 |
| 2 | 345 | 13 | 13 |
| 3 | 666 | 13 | 13 |

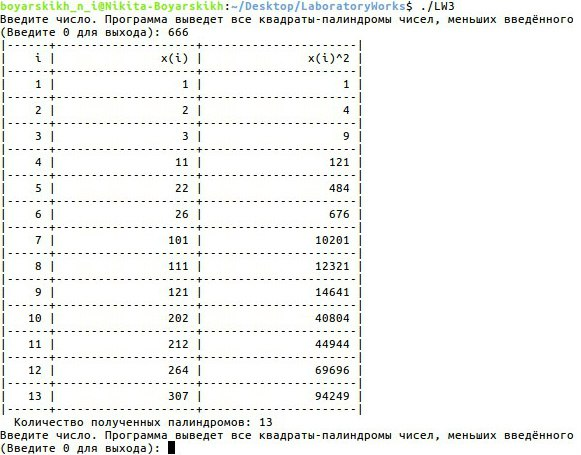
**Скриншоты, демонстрирующие работу программы**



*Рисунок 2 – тестирование программы при исходных данных №1 (см. табл. 1)*

**

*Рисунок 3 – тестирование программы при исходных данных №2 (см. табл. 1)*

*Рисунок 4 – Тестирование программы при исходных данных №3 (см. табл. 1)*

**Вывод:**

1) Разработан алгоритм для решения поставленной задачи. Составлена его схема в среде [draw.io](http://draw.io)(рис. 1)

2) Создан код программы по алгоритму в среде Vim

3) Проведено тестирование (см. табл. 1)

4) Тестирование показало корректность работы при заданных исходных данных (см. табл. 1 и рис. 2 - 4)