МЭСИ

Информатика и программирование, лабораторная работа №1

Выполнил Вишняков Федор Николаевич, ДКО-102

Дата сдачи работы: \_\_.10.2012

Проверяющий Переверзев Илья Юрьевич

Москва, 2012

Оглавление

[Операторы и конструкции языка, используемые в программах 3](#_Toc337204621)

[Задачи 4](#_Toc337204622)

[Задача 1 4](#_Toc337204623)

[Постановка 4](#_Toc337204624)

[Текст программы 4](#_Toc337204625)

[Описание программы 5](#_Toc337204626)

[Результат работы программы 5](#_Toc337204627)

[Задача 2 6](#_Toc337204628)

[Постановка 6](#_Toc337204629)

[Текст программы 6](#_Toc337204630)

[Описание программы 7](#_Toc337204631)

[Результат работы программы 7](#_Toc337204632)

[Задача 3 8](#_Toc337204633)

[Постановка 8](#_Toc337204634)

[Текст программы 8](#_Toc337204635)

[Описание программы 9](#_Toc337204636)

[Результат работы программы 10](#_Toc337204637)

[Задача 4 10](#_Toc337204638)

[Постановка 10](#_Toc337204639)

[Текст программы 10](#_Toc337204640)

[Описание программы 11](#_Toc337204641)

[Результат работы программы 12](#_Toc337204642)

[Задачи 5, 6 12](#_Toc337204643)

[Постановка 12](#_Toc337204644)

[Тексты программ 13](#_Toc337204645)

[Результаты работы программ 14](#_Toc337204646)

[Используемая литература и Internet-ресурсы 15](#_Toc337204647)

# Структура программы на языке C++

Программа составляется из заголовочных файлов (.h) и файлов с исходным кодом (.cpp). Заголовочные файлы подключаются для возможности использования реализованных кем-либо функций, заданных констант, описанных типов и так далее. Заголовочные файлы обычно содержат прототипы функций и задание констант. Файлы с тем же названием, но расширением cpp содержат реализации функций, которые при создании превращаются компилятором в объектные файлы, позволяющие применять описанные в заголовке функции, но не дающие доступ к исходному коду реализаций.

Начинается программа обычно с препроцессорных команд. Программист сообщает компилятору, какие заголовочные файлы нужно подключить к программе, какие константы определить. Далее содержание кода варьируется, но неизменно одно – для того, чтобы программа запускалась с данного cpp-файла, нужно чтобы он содержал функцию main. Она является точкой входа в программу. Из неё вызываются другие функции. По завершении функции main программа также завершается.

# Операторы и конструкции языка, используемые в программах

#include

Директива препроцессора, подключающая библиотеку к программе

using namespace

Использование переменных и операторов пространства имени без указания этого имени

setlocale

Определение локализации для определенной категории символов

int, double

Типы числовых переменных (целых и вещественных)

cout

Оператор вывода библиотеки iostream

cin

Оператор считывания данных с клавиатуры

endl

Перенос строки для оператора cout

precision

Установка выводимого числа знаков (не включая «.»). Требуется библиотека iomanip

do {} while()

Цикл с постусловием (гарантированно выполнится один раз)

for (;;)

Итерационный цикл

system()

Функция, выполняющая процессорную команду

# 

# Задачи

## Задача 1

### Постановка

Числа Фибоначчи — числовая последовательность, в которой каждое последующее число равно сумме двух предыдущих чисел (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …). Число K – индекс элемента последовательности, начиная с нуля. Например, у числа Фибоначчи 8 будет индекс 6, а у 34 будет индекс 9. Даны числа M (M>0) и N (N>=M), вывести все элементы последовательности, удовлетворяющие условию M<=K<=N.

Пример:

M=3

N=8

Результат:

2 3 5 8 13 21

### Текст программы

#include <iostream>

using namespace std;

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int i = 0, a = 0, b = 1, c, m, n;

// ввод данных с "защитой от дурака"

do {

m = n = 0;

cout << "Введите M и N> ";

cin >> m >> n;

if (m <= 0 || n < m)

cout << "Неверные исходные данные\n";

} while (m <= 0 || n < m);

// основной цикл вывода чисел

while (i <= n) {

if (i >= m)

cout << a << " ";

c = a + b;

a = b;

b = c;

i++;

}

// просмотр результата программы до нажатия любой клавиши

cout << endl;

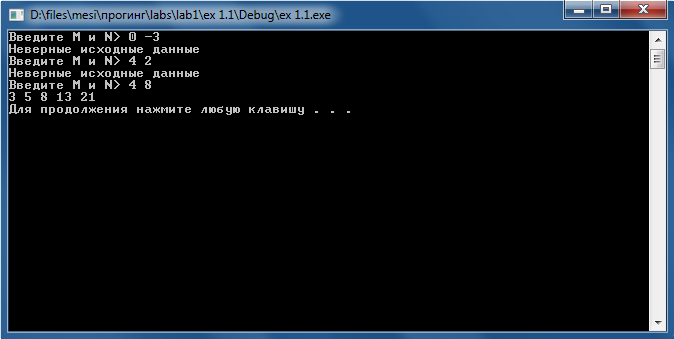
system("pause");

}

### Описание программы

Тривиальные действия пояснены в комментариях. В основном цикле чисел в переменные a и b записываются значения чисел из ряда Фибоначчи. Т.к. нам даны только границы, нам нужно считать числа с самого начала. Выводить же числа мы будем если они удовлетворяют заданному условию. Правая граница проверяется в условии цикла, левая – при выводе в теле цикла.

### Результат работы программы



## Задача 2

### Постановка

Числа Фибоначчи — числовая последовательность, в которой каждое последующее число равно сумме двух предыдущих чисел (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …). Даны числа A (0<=A<=9) и M, вывести все элементы последовательности меньше M, у которых последняя цифра равна A.

Пример:

A=5

M=100000

Результат:

5 55 6765 75025

### Текст программы

#include <iostream>

using namespace std;

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int d = 0, e = 1, c, a, m;

// ввод данных с "защитой от дурака"

do {

a = m = 0;

cout << "Введите A и M> ";

cin >> a >> m;

if (a < 0 || a > 9)

cout << "Неверные исходные данные\n";

} while (a < 0 || a > 9);

// основной цикл вывода чисел

while (d < m) {

if (d % 10 == a)

cout << d << " ";

c = d + e;

d = e;

e = c;

}

// просмотр результата программы до нажатия любой клавиши

cout << endl;

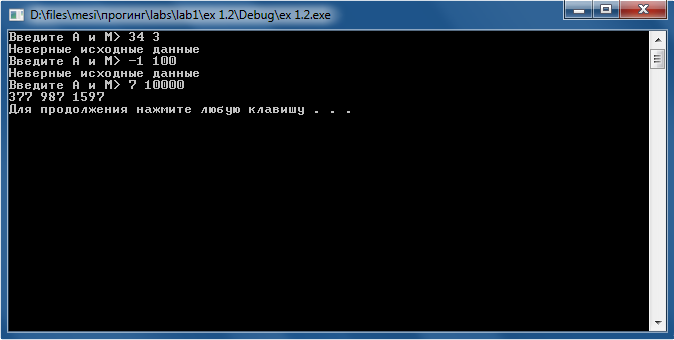
system("pause");

}

### Описание программы

Тривиальные действия пояснены в комментариях. В основном цикле чисел в переменные c и d записываются значения чисел из ряда Фибоначчи. Также в цикле идет проверка на нужность текущего элемента ряда. Если остаток деления этого числа на 10 равен a (то есть последняя цифра числа – a), то мы выводим это число.

### Результат работы программы

****

## Задача 3

### Постановка

Простое число — это натуральное число, имеющее ровно два различных натуральных делителя: единицу и само себя. Последовательность простых чисел: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, …. Дано число N. Найти первые 10 простых чисел, начиная с N, у которых сумма цифр четная.

Пример:

N=50

Результат:

53 59 71 73 79 97 101 103 107 109

### Текст программы

#include <iostream>

using namespace std;

// прототип функции определения четности суммы цифр числа

bool need(int a);

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int n, i, j, k = 0;

bool f;

// ввод входных данных с защитой от дурака

do {

n = 0;

cout << "Введите N> ";

cin >> n;

if (n < 1)

cout << "Неверные исходные данные" << endl;

} while (n < 1);

if (n == 1)

n = 2;

// основной цикл вывода

for (i = n; k < 10 ; i++) {

f = true;

if (!need(i))

f = false;

else

for (j = 2; j <= sqrt((double(i))); j++)

if (i % j == 0) {

f = false;

break;

}

if (f) {

cout << i << " ";

k++;

}

}

// просмотр результатов до нажатия любой клавиши

cout << endl;

system("pause");

return 0;

}

// реализации функции need

bool need(int a) {

int sum = 0;

while (a != 0) {

sum += a % 10;

a /= 10;

}

if (sum % 2 == 0)

return true;

else

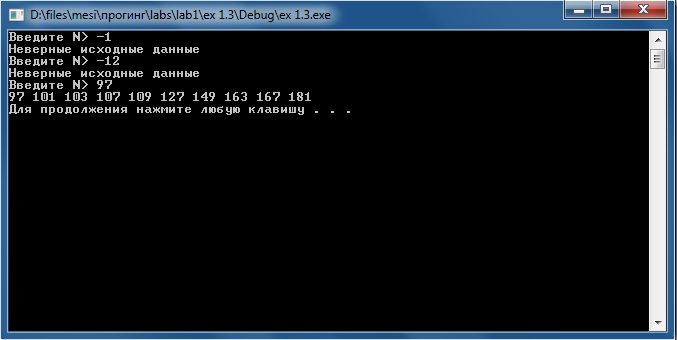
return false;

}

### Описание программы

Основной цикл: начинаем рассмотрение с числа n, заканчиваем цикл тогда, когда найдем 10 таких чисел. Эти числа найдутся всегда, т.к. ряд простых чисел бесконечен. Функция need получает на вход число, считает сумму его цифр и возвращает true, если сумма цифр четна или false, если это не так. Если число нам не нужно по признаку четности суммы цифр, нам уже не нужно проверять, простое ли это число. Иначе же мы делаем проверку. Если число снова подходит нам, т.е. оно простое, то мы выводим его и увеличиваем счетчик нужных нам чисел на единицу.

### Результат работы программы

****

## Задача 4

### Постановка

Простое число — это натуральное число, имеющее ровно два различных натуральных делителя: единицу и само себя. Последовательность простых чисел: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, …. Число K – индекс элемента последовательности, начиная с нуля. Например, у простого числа 11 индекс равен 4, а у 29 индекс равен 9. Даны числа M (M>0) и N (N>=M). Вывести все элементы последовательности, у которых индекс нечетный и удовлетворяющие условию M<=K<=N.

Пример:

M=4

N=7

Результат:

13 19

### Текст программы

#include <iostream>

using namespace std;

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int m, k, n, i, j;

bool f;

// ввод входных данных с защитой от дурака

do {

m = n = 0;

cout << "Введите M и N> ";

cin >> m >> n;

if (m <= 0 || n < m)

cout << "Неверные исходные данные" << endl;

} while (m <= 0 || n < m);

// основной цикл вывода

i = 1;

for (k = 0; k <= n; k++) {

do {

f = true;

i++;

for (j = 2; j <= sqrt((double(i))); j++)

if (i % j == 0) {

f = false;

break;

}

} while (f == false);

if (k >= m && k % 2 != 0)

cout << i << " ";

}

// просмотр результатов до нажатия любой клавиши

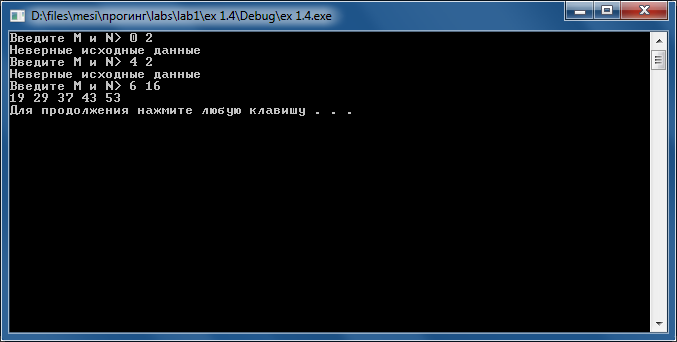
cout << endl;

system("pause");

}

### Описание программы

Внешний цикл for со счетчиком индексов числового ряда начинается с 0 и заканчивается заданной границей. Внутренней цикл с постусловием дает перейти циклу for на следующий шаг только когда внутренний обнаружит следующее простое число. После обнаружения и перед следующим тиком цикла выполняется проверка на соблюдение левой границы и нечетность числа с выводом при положительном результате.

Результат работы программы

## Задачи 5, 6

### Постановка

Вычислить сумму ряда. Вычисления заканчивать при достижении заданной степени точности (т.е. очередной член ряда становится меньше введенной точности, либо разность между двумя соседними членами ряда становится меньше введенного значения точности).

### Тексты программ

// 3

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int i;

double x, z = 1, t, g = 0, p = 0;

// ввод данных

cout << "Введите точность(Q) и значение X(Q)> ";

cin >> t >> x;

// основной цикл вычисления

double iks = 1;

double fctrl = 1;

double fctrlm = 2;

double med = 1;

for (i = 1; ; i++) {

g = med \* iks / fctrl;

cout << g << endl;

med \*= -1;

iks = iks\*x\*x;

fctrl \*= fctrlm;

fctrlm++;

z += g;

if (abs(g) < t || (i!=1&&abs(p)-abs(g)<t))

break;

p = g;

}

// результат

cout.precision(15);

cout << z << endl;

// просмотр результата до нажатия любой клавиши

system("pause");

return 0;

}

// 16

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

// русская кодировка

setlocale(LC\_ALL, "rus");

// объявление переменных

int i;

double p, g, t = 0, x, z = 0;

// ввод данных

do {

x = 0;

cout << "Введите точность(Q) и значение X(Q, x != 0)> ";

cin >> t >> x;

} while (x == 0);

g = 0;

// основной цикл вычисления

double iks = x;

double fctrl = 1;

double fctrlm = 2;

double med = 1;

for (i = 1; ; i++) {

g = med \* (iks/fctrl-1/iks) \* (iks/fctrl-1/iks);

iks \*= x;

med \*= -1;

fctrl \*= fctrlm;

fctrlm++;

z += g;

if (abs(g) < t || (i!=1&&abs(p)-abs(g)<t))

break;

p = g;

}

// результат

cout.precision(t);

cout << z << endl;

// просмотр результата до нажатия любой клавиши

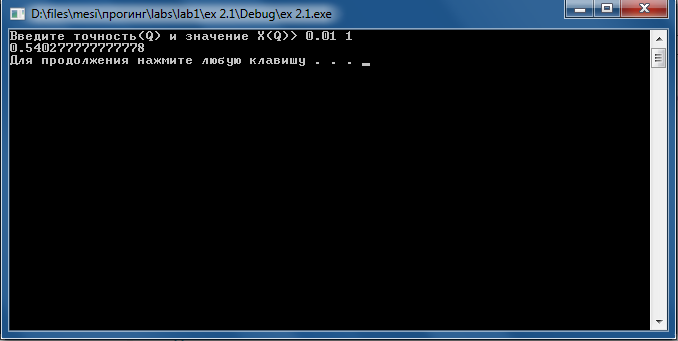
system("pause");

return 0;

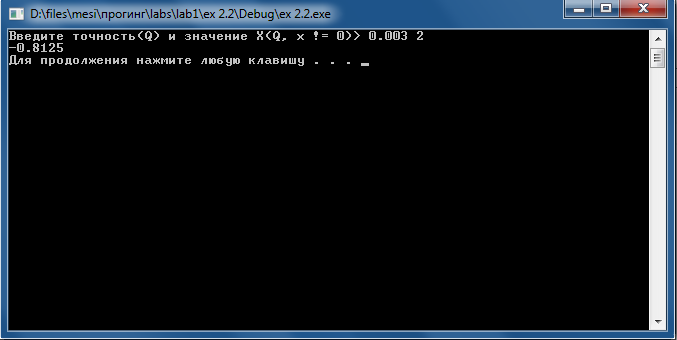
}

### Результаты работы программ

3

****

16



# Используемая литература и Internet-ресурсы

«C++. Руководство для начинающих» Г. Шилдт