**Hello, world**

Язык C++ является компилируемым языком. Для того, чтобы написать программу, вам необходимо в любом текстовом редакторе набрать, например, следующий текст и сохранить его в файле, например, hello.cpp.

#include <iostream>

using namespace std;

int main()

{

cout<<"Hello, world!"<<endl;

return 0;

}

Будьте внимательны: язык C++ является чувствительным к регистру букв, то есть заменить main на Main или MAIN нельзя. Весь текст (за исключением текстовой строки "Hello, world!") нужно набирать в нижнем регистре.

После этого вам нужно откомпилировать этот файл (создать из этого файла исполняемый машинный код) при помощи следующей команды (в системе Linux):

$ c++ hello.cpp

В среде разработки (вроде Dev-C++) для компиляции программы существует пункт меню, вызывающий компилятор. Если ваша программа написана правильно, то компилятор не выдаст никаких сообщений об ошибках и создаст исполняемый файл (a.out в системе Linux или exe-файл в системе Windows). Этот файл содержит исполняемый двоичный машинный код.

Рассмотрим подробней текст этой программы. В первой строчке мы подключаем к нашей программе файл с именем iostream, в котором содержится описание стандартной библиотеки ввода-вывода языка C++. В этом файле находится, в частности, определение объектов cout и endl, который мы будем использовать позднее. Вторая строка указывает компилятору на то, что мы будем использовать все функции, входящие в пространство имен std, то есть все функции, относящиеся к стандартной библиотеке C++. Третья строка содержит объявление функции main, не принимающей никаких аргументов и возвращающей значение int. Эта функция должна быть в каждой программе, именно эта функция получает управление при запуске программы. Четвертая строка содержит открывающуюся фигурную скобку, что означает начало функции main. В пятой строке мы при помощи оператора << помещаем в объект cout строку "Hello, world!", а потом специальный объект endl, означающий символ перевода строки. Это приводит к печати на экране этой строки и последующему переводу каретки. В шестой строке мы даем инструкцию return, завершающую выполнение функции main и возвращающую нулевое значение. Седьмая строка содержит фигурную скобку, синтаксически закрывающую функцию main.

Для начала можно считать, что все строки, кроме пятой, являются некоторым набором “заклинаний”, без которых программа не будет работать и которые обязательно нужно указать, а вот пятую строку можно заменить на другие строки с различными инструкциями.

## Файлы

Данный листочек посвящен основам чтения данных из файлов и вывода данных в файл.

На самом деле, файловый ввод-вывод ничем не отличается от консольного. За единственным исключением – если данные читаются из файла, то в любой момент можно вернуться к началу файла и считать все заново.

Для того, чтобы в C++ работать с файлами, необходимо подключить заголовочный файл fstream:

     #include <fstream>

После этого можно объявлять объекты, привязанные к файлам: для чтения данных из файла используются объекты типа ifstream (аббревиатура от **i**nput **f**ile **stream**, для записи данных в файл используются объекты типа ofstream (**o**utput **f**ile **stream**). Например

     ifstream in;  // Поток in будем использовать для чтения  
     ofstream out; // Поток out будем использовать для записи

Чтобы привязать тот или иной поток к файлу (открыть файл для чтения или для записи) используется метод open, которому необходимо передать параметр – текстовую строку, содержащую имя открываемого файла.

     in.open("input.txt");  
     out.open("output.txt");

После открытия файлов и привязки их к файловым потокам, работать с файлами можно так же, как со стандартными потоками ввода-вывода cin и cout. Например, чтобы вывести значение переменной x в поток out используются следующая операция

     out<<x;

А чтобы считать значение переменной из потока in

     in>>x;

Для закрытия ранее открытого файла используется метод close() без аргументов:

     in.close();  
     out.close();

Закрытый файловый поток можно переоткрыть заново при помощи метода open, привязав его к тому же или другому файлу.

При считывании данных из файла может произойти ситуация достижения конца файла (**end of file**, сокращенно **EOF**). После достижения конца файла никакое чтение из файла невозможно. Для того, чтобы проверить состояние файла, необходимо вызвать метод eof(). Данный метод возвращает true, если достигнут конец файла илиfalse, если не достигнут. Кроме того, состояние файлового потока можно проверить, если просто использовать идентификатор потока в качестве логического условия:

     if (in)  
     {  
     }

Также можно использовать в качестве условия результат, возвращаемой операцией считывания. Если считывание было удачным, то результат считается истиной, а если неудачным  – ложью. Например, организовать считывание последовательности целых чисел можно так:

     int d;  
     while(in>>d)  
     {  
     }

А организовать считывание файла построчно (считая, что строка заканчивается символом перехода на новую строку) так:

     string S;  
     while ( getline(in,S))  
     {  
     }

А еще полезно повторить листок "Строки".

### Упражнения

Во всех заданиях этого листка программа должна читать данные из файла input.txt и выводить результат в файл output.txt. Пробельными символами называются символы пробела, табуляции, перехода на новую строку. Непробельными символами называются все остальные символы, чьи ASCII-коды не меньше 20h. Словом называется последовательность непробельных символов, разделенных пробельными символами.

1. (A) **Сумма**

Входной файл содержит два целых числа. Запишите в выходной файл их сумму.

**Задача №2938. Дележ яблок – 1**

*N* школьников делят *K* яблок поровну, неделящийся остаток остается в корзинке. Сколько яблок достанется каждому школьнику?

**Входные данные**

Программа получает на вход числа *N* и *K*.

**Выходные данные**

Программа должна вывести искомое количество яблок.

**Примеры**

**входные данные**

3

14

**выходные данные**

4

**Задача №2941. Последняя цифра**

Дано натуральное число. Выведите его последнюю цифру.

**Входные данные**

Вводится натуральное число.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

179

**выходные данные**

9

**Задача №2943. Число десятков**

Дано неотрицательное целое число. Найдите число десятков в его десятичной записи (то есть вторую справа цифру его десятичной записи).

**Входные данные**

Вводится неотрицательное целое число.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

179

**выходные данные**

7

**Задача №2944. Сумма цифр**

Дано **трехзначное** число. Найдите сумму его цифр.

**Входные данные**

Вводится трехзначное число.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

179

**выходные данные**

17

**Задача №2947. Электронные часы – 1**

Дано число ***n***. С начала суток прошло ***n*** минут. Определите, сколько часов и минут будут показывать электронные часы в этот момент. Программа должна вывести два числа: количество часов (от 0 до 23) и количество минут (от 0 до 59). Учтите, что число ***n*** может быть больше, чем количество минут в сутках.

**Входные данные**

Вводится целое число ***n***.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

150

**выходные данные**

2 30

**входные данные**

1441

**выходные данные**

0 1

**Задача №2948. Электронные часы – 2**

Электронные часы показывают время в формате **h:mm:ss**, то есть сначала записывается количество часов, потом обязательно двузначное количество минут, затем обязательно двузначное количество секунд. Количество минут и секунд при необходимости дополняются до двузначного числа нулями.

С начала суток прошло ***n*** секунд. Выведите, что покажут часы.

**Входные данные**

Вводится целое число ***n***.

**Выходные данные**

Выведите ответ на задачу, соблюдая требуемый формат.

**Примеры**

**входные данные**

3602

**выходные данные**

1:00:02

**входные данные**

129700

**выходные данные**

12:01:40

**Задача №2950. Конец уроков**

В некоторой школе занятия начинаются в 9:00. Продолжительность урока — 45 минут, после 1-го, 3-го, 5-го и т.д. уроков перемена 5 минут, а после 2-го, 4-го, 6-го и т.д. — 15 минут. Определите, когда заканчивается указанный урок.

**Входные данные**

Дан номер урока (число от 1 до 10).

**Выходные данные**

Выведите два целых числа: время окончания урока в часах и минутах. При решении этой задачи нельзя пользоваться циклами и условными инструкциями.

**Примеры**

**входные данные**

3

**выходные данные**

11 35

**входные данные**

2

**выходные данные**

10 35

**Задача №2951. Стоимость покупки**

Пирожок в столовой стоит *a* рублей и *b* копеек. Определите, сколько рублей и копеек нужно заплатить за *n* пирожков.

**Входные данные**

Программа получает на вход три числа: *a*, *b*, *n*.

**Выходные данные**

Программа должна вывести два числа: стоимость покупки в рублях и копейках.

**Примеры**

**входные данные**

10

15

2

**выходные данные**

20 30

**входные данные**

2

50

4

**выходные данные**

10 0

**Задача №2952. Разность времен**

Даны значения двух моментов времени, принадлежащих одним и тем же суткам: часы, минуты и секунды для каждого из моментов времени. Известно, что второй момент времени наступил не раньше первого. Определите, сколько секунд прошло между двумя моментами времени.

**Входные данные**

Программа на вход получает три целых числа — часы, минуты, секунды, задающие первый момент времени и три целых числа, задающих второй момент времени.

**Выходные данные**

Выведите число секунд между этими моментами времени.

**Примеры**

**входные данные**

1

1

1

2

2

2

**выходные данные**

3661

**входные данные**

1

2

30

1

3

20

**выходные данные**

50

**Задача №2953. Автопробег**

За день машина проезжает *n* километров. Сколько дней нужно, чтобы проехать маршрут длиной *m* километров?

При решении этой задачи нельзя пользоваться условной инструкцией if и циклами.

**Входные данные**

Программа получает на вход числа *n* и *m*.

**Выходные данные**

Выведите ответ на задачу.

**Примеры**

**входные данные**

700

750

**выходные данные**

2

**входные данные**

700

2100

**выходные данные**

3

**Задача №2955. Улитка**

Улитка ползёт по вертикальному шесту высотой *h* метров, поднимаясь за день на *a* метров, а за ночь спускаясь на *b* метров. На какой день улитка доползёт до вершины шеста?

**Входные данные**

Программа получает на вход натуральные числа *h*, *a*, *b*. Гарантируется, что *a*>*b*.

**Выходные данные**

Программа должна вывести одно натуральное число.

**Примечание**

При решении этой задачи нельзя пользоваться условной инструкцией if и циклами.

**Примеры**

**входные данные**

10

3

2

**выходные данные**

8

**входные данные**

20

7

3

**выходные данные**

5