

Файлы. Потоки чтения и записи.

Во всех задачах ввод-вывод осуществляется только посредством чтения-записи файлов. Файл со входными данными называется `input.txt`, с выходными — `output.txt`.

Во входных данных всех задач можно рассчитывать на то, что **все** строки файла, включая последнюю, заканчиваются переносом строки (символом `'\n'`).

Во всех задачах требуется записать решение, читающее файл **один** раз.

Открытие файла

В C++ работа с файлами осуществляется посредством нескольких классов стандартной библиотеки. Самые типичные это:

- `ifstream` - предназначен для чтения из файла
- `ofstream`, предназначен для создания нового файла и записи данных в существующий
- `fstream` - поддерживает как чтение, так и запись в файл

Все три потока объявлены в заголовочном файле стандартной библиотеки `fstream`:

```
include <fstream>
```

Для работы с файлом (как чтение, так и запись) нужно его открыть. Делается это с помощью функции `open`, которая имеет две сигнатуры:

```
open(<путь к файлу>)
```

```
open(<путь к файлу>, <режим работы>)
```

- путь к файлу

<путь к файлу> — это строка, в которой хранится имя файла.

```
open("input.txt")
```

- режим работы

Второй параметр — идентификатор режима, равный

- `ios::in`, если файл открывается для чтения данных (Read); может устанавливаться для потоков `ifstream` или `fstream`
- `ios::out`, если файл открывается для создания и записи данных (Write); может устанавливаться для потоков `ofstream` или `fstream`
- `ios::app` если файл открывается для добавления данных в конец файла (Append);
- `ios::trunc` файл усекается при открытии (Truncate). Может быть установлен, если также установлен режим `out`.
- `ios::binary` файл открывается в бинарном режиме.

Если при открытии режим не указан, то по умолчанию для объектов `ofstream` применяется режим `ios::out`, а для объектов `ifstream` - режим `ios::in`. Для объектов `fstream` совмещаются режимы `ios::out` и `ios::in`.

При попытке открыть в режиме чтения несуществующий файл произойдёт ошибка.

Если несуществующий файл открыть в режиме записи или дополнения (`ios::out` или `ios::app`), то файл с таким именем будет создан.

```
std::ifstream fis;
```

```
fis.open(<путь к файлу>);
```

Открыть файл, впрочем, можно и без функции `open`. Можно передать имя файла (и режим) параметром непосредственно конструктора потока:

```
std::ifstream fis("input.txt");
```

Чтение данных из файла

Файл, открытый на чтение, можно читать точно так же, как это делается при чтении значений из консоли, при помощи оператора потокового ввода `>`. И точно так же, как и для чтения строк с консоли, можно использовать функцию `getline`, только вместо потока ввода `cin` нужно указать наш файловый поток ввода:

```
std::string line;
```

```
std::getline(cin, line);
```

Вывод данных в файл

Для вывода данных в файл используется оператор потокового вывода `<`:

```
std::ofstream fout("output.txt");
```

```
fout << "Hello, 57!"
```

Закрытие файла

После окончания работы с файлом необходимо закрыть его при помощи метода `close()`. Вот так:

```
fis.close();
```

```
fout.close();
```

Файл после окончания работы с ним нужно закрывать **ОБЯЗАТЕЛЬНО!**

Внимание! Во всех задачах можно (и нужно) обойтись без сортировки.
Все задачи можно решить, прочитав входной файл только один раз.

A. *Сумма нескольких чисел*

Во входном файле записано некоторое количество чисел, каждое в отдельной строке. Выведите их сумму.
Обратите внимание: функция `int()` правильно обрабатывает строки даже вместе с символом переноса строки (т.е. можно просто читать строку из файла и преобразовывать в число).

Input	Output
1 2 3	6

B. *Поиск символа*

Определите, есть ли во входном файле символ `@`. Выведите слово `YES` или `NO`.
Входной файл может быть очень большим, поэтому считывать файл нужно посимвольно.

Input	Output
Valid email: Bilbo.Baggins@bagend.hobbiton.shire.me	YES
Hello, world!	NO

C. *Шифр Цезаря*

Зашифруйте текстовый файл шифром Цезаря. Символы первой строки файла должны циклически сдвигаться на 1, второй строки — на 2, третьей строки — на 3 и т.д.
Символы, отличные от латинских букв, изменять не требуется.
В этой задаче удобно считывать файл построчно, шифруя каждую строку в отдельности.

Input	Output
Hello	Ifmmp
Hello	Jgnnq
Hello	Khoor
Hello	Lipps

D. *Сумма чисел в файле - I*

Дан файл, каждая строка которого может содержать одно или несколько целых чисел, разделённых одним или несколькими пробелами. Символов, отличных от цифр, пробелов и знаков переноса строки, в файле нет.
Вычислите сумму чисел в каждой строке и выведите эту сумму (для каждой строки выводится сумма чисел в этой строке).
В этой задаче удобно считывать данные построчно и использовать для каждой строки метод `split()`.

Input	Output
2 2	4
3 4	7
1 5	6

E. *Сумма чисел в файле - II*

В файле могут быть записаны десятичные цифры и всё, что угодно.
Натуральное число — последовательность цифр, начинающаяся не с нуля и не являющаяся частью другой последовательности, образующей натуральное число.
Например:
для в строке `abc123 2023 000340004` есть три натуральных числа: 123, 2023, 340004.
Вычислите сумму всех чисел, записанных в файле. В этой задаче удобно считывать данные посимвольно.

Input	Output
123 aaa456 1x2y3 4 5 6	600

F. Статистика по файлу

Определите сколько в файле букв латинского алфавита, слов и строк. Под словом понимается последовательность букв латинского алфавита, ограниченная слева и справа символами-не-буквами или началом (концом) строки.

Выведите три найденных числа в формате, приведённом в примере.

Для экономии памяти читайте файл посимвольно, то есть не сохраняя целиком в памяти файл или отдельные его строки. После последней строки не забудьте про перенос ('\n').

Input	Output
Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. Complex is better than complicated.	Input file contains: 108 letters 20 words 4 lines

G. Максимальный балл по классам

В олимпиаде по информатике принимало участие несколько человек. Информация о результатах олимпиады записана в файле, каждая строка которого имеет вид: фамилия имя класс балл.

Фамилия и имя — текстовые строки, не содержащие пробелов. Класс — одно из трех чисел 9, 10, 11. Балл — целое число от 0 до 100.

Победителем олимпиады становится человек, набравший больше всех баллов. Победители определяются независимо по каждому классу.

Определите количество баллов, которое набрал победитель в каждом классе. Гарантируется, что в каждом классе был хотя бы один участник.

Выведите три числа: баллы победителя олимпиады по 9 классу, по 10 классу, по 11 классу.

В этой задаче файл необходимо считывать построчно, не сохраняя содержимое файла в памяти целиком.

Достаточно хранить в массиве 3 значения: максимальный балл в 9, 10 и 11 классах.

Решения, содержащие разбор трёх случаев вида

```
if (form == 9) {  
    ....  
} else if (form == 10) {  
    ....  
} else {  
    ....  
}
```

приниматься не будут.

Input	Output
Иванов Сергей 9 90 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	93 91 92

H. Средний балл по классам

В условиях задачи G определите и выведите средние баллы участников олимпиады в 9 классе, в 10 классе, в 11 классе.

Input	Output
Иванов Сергей 9 90 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	91.5 91.0 92.0

I. *Количество победителей по классам*

В условиях задачи G определите количество школьников, ставших победителями в каждом классе. Победителями объявляются все, кто набрал наибольшее число баллов по данному классу.

Гарантируется, что в каждом классе был хотя бы один участник.

Выведите три числа: количество победителей олимпиады по 9 классу, по 10 классу, по 11 классу.

Input	Output
Иванов Сергей 9 80 Сергеев Пётр 10 80 Петров Василий 11 81 Васильев Андрей 9 81 Андреев Александр 10 80 Александров Роман 9 81 Романов Иван 11 80	2 2 1

J. *Победитель олимпиады*

Зачёт в олимпиаде проводится без деления на классы. Выведите фамилию и имя победителя олимпиады. Если таких несколько — выведите только их количество.

Input	Output
Иванов Сергей 9 90 Сергеев Пётр 10 95 Петров Иван 11 85	Сергеев Пётр
Иванов Сергей 9 90 Сергеев Пётр 10 85 Петров Иван 11 90	2