



## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

## Отметки

## Примечания

## Форумы обсуждений

## Сообщения

## Информация о курсе

## Неделя 1

## Знакомство с искусством C++

Вideo ГотовоМатериалы для  
самостоятельного изучения ГотовоТренировочные упражнения ГотовоДругое Готово

ОБЯЗАТЕЛЬНО	ОЦЕНКА	НЕОБХОДИМО СДАТЬ
✓ Тест Знакомство с C++ 30 мин	100 %	12 июля г. 12:59 +06
✓ Тест Языковые конструкции 30 мин	83,33 %	12 июля г. 12:59 +06
✓ Задание по программированию A + B 20 мин	100 %	12 июля г. 12:59 +06
✓ Задание по программированию Минимальная строка 24 мин	100 %	12 июля г. 12:59 +06
✓ Задание по программированию Уравнение 3 ч 13 м	100 %	12 июля г. 12:59 +06
✓ Задание по программированию Второе вхождение 20 мин	100 %	12 июля г. 12:59 +06
✓ Задание по программированию Наибольший общий делитель 34 мин	100 %	12 июля г. 12:59 +06
✓ Задание по программированию Двоичные числа 20 мин	100 %	12 июля г. 12:59 +06



## Задание по программированию: A + B

Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

Инструкции

Моя работа

Обсуждения

На вход программе через стандартный ввод передаются два целых числа, по модулю не превышающие 100000. Выведите в стандартный вывод их сумму.

stdin	stdout
2 3	5
-4 -9	-13
-1 2	1

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Моя работа'.

### Подсказки

- Ваша программа должна выводить в cout только одно число — сумму введённых чисел (т.е. не надо выводить в cout входные числа или строки типа "A = ")
- Вам не надо проверять, что входные числа не превышают по модулю 100000. Мы гарантируем, что в тестирующей системе вашей программы будут подаваться только числа, удовлетворяющие указанным ограничениям

### Пояснение

Если вы не знаете, как компилировать и запускать программы на C++, вы можете просмотреть видео из блока "Компиляция, запуск, отладка" и, настроив среду разработки, вернуться к этой задаче.

[Основы разработки на C++: белый пояс](#) > [Неделя 1](#) > [Минимальная строка](#)

&lt; Предыдущий Дальше &gt;

## Задание по программированию: Минимальная строка

Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

[Инструкции](#)[Моя работа](#)[Обсуждения](#)

В стандартном потоке даны три строки, разделённые пробелом. Каждая строка состоит из строчных латинских букв и имеет длину не более 30 символов. Выведите в стандартный вывод лексикографически минимальную из них.

### Примеры

stdin	stdout
milk milkshake month	milk
c a b	a
fire fog wood	fire

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Задание по программированию: Уравнение

✓ Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

[Инструкции](#) [Моя работа](#)

[Обсуждения](#)

На вход вашей программы в стандартном потоке ввода подаются действительные коэффициенты  $A$ ,  $B$  и  $C$  уравнения  $Ax^2 + Bx + C = 0$ . Выведите все его различные действительные корни в поток вывода в любом порядке, при этом разделяя корни пробелами. Гарантируется, что хотя бы один из коэффициентов уравнения не равен нулю.

### Пример

stdin	stdout
2 5 2	-0.5 -2
2 4 2	-1
2 1 2	
0 4 10	-2.5

### Подсказка

Для вычисления квадратного корня используйте функцию `sqrt` из библиотеки `cmath`. Чтобы подключить библиотеку, в начале программы напишите

```
1 #include <cmath>
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке "Моя работа".



## Тренировочное задание по программированию: Деление

✓ Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

**Инструкции** Моя работа

Обсуждения

Дано два целых числа A и B в диапазоне от 0 до 1 000 000 включительно.  
Напишите программу, которая вычисляет целую часть частного от деления  
A на B.

Если B = 0, выведите "Impossible".

### Пример

stdin	stdout
10 2	5
3 5	0
11 0	Impossible

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.



## Тренировочное задание по программированию: Расчёт стоимости товара

Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

**Инструкции** Моя работа

Обсуждения

Написать программу вычисления стоимости покупки товара с учётом скидки. Если исходная стоимость товара больше А рублей, то на неё устанавливается скидка в X процентов. Если исходная стоимость товара больше В рублей, то на неё устанавливается скидка в Y процентов.

На вход программе даны пять вещественных чисел: N, A, B, X, Y (A < B) - где N - исходная стоимость товара. Выведите стоимость покупки товара с учётом скидки.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Пример

stdin	stdout
100 110 120 5 10	100
115 110 120 5 10	109.25
150 110 120 5 12.5	131.25



## Программирование: Чётные числа

✓ Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

**Инструкции** Моя работа

Обсуждения

Дано два целых числа A и B (A <= B, A >= 1, B <= 30000). Выведите через пробел все чётные числа от A до B (включительно).

Для проверки целого числа  $x$  на чётность используется операция *взятия остатка от деления на 2*, которая в C++ оформляется с помощью символа "%". Например,

```
1 int x;
2 cin >> x;
3 if (x % 2 == 0) {
4     // x - чётное число
5 } else {
6     // x - нечётное число
7 }
```

**Пример**

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

stdin	stdout
1 10	2 4 6 8 10
2 3	2
9 11	10



## Задание по программированию: Второе вхождение

✓ Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

**Инструкции**

Моя работа

Обсуждения

Дана строка. Найдите в этой строке второе вхождение буквы **f** и выведите индекс этого вхождения. Если буква **f** в данной строке встречается только один раз, выведите число **-1**, а если не встречается ни разу, выведите число **-2**. Индексы нумеруются с нуля.

Пример

stdin	stdout
comfort	-1
coffee	3
car	-2

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



Основы разработки на C++: белый пояс > Неделя 1 > Наибольший общий делитель

< Предыдущий Дальше >

## Задание по программированию: Наибольший общий делитель

Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06

Вы все еще можете выполнить задание до завершения курса.

**Инструкции**

Моя работа

Обсуждения

В `stdin` даны два натуральных числа. Выведите в `stdout` их наибольший общий делитель.

**Пример**

<code>stdin</code>	<code>stdout</code>
25 27	1
12 16	4
13 13	13

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Задание по программированию: Двоичные числа

✓ Зачет · 1/1 баллов

**Срок сдачи** Задание необходимо было сдать 12 июля г., 12:59 +06  
Вы все еще можете выполнить задание до завершения курса.

**Инструкции** Моя работа

Обсуждения

На вход дано целое положительное число N. Выведите его в двоичной системе счисления без ведущих нулей.

### Пример

stdin	stdout
5	101
32	100000
1	1

### Подсказка

- число N можно считывать в переменную типа `int`
- алгоритм перевода из десятичной системы счисления в двоичную, может быть найден, например, здесь:  
<https://www.yaklass.ru/p/informatika/10-klass/informatciia-i-informatcionnye-protsessy-11955/predstavlenie-chislovoi-informatciyi-v-kompiutere-11901/re-62e8903e-ba93-478a-85b5-ae45af40c658>
- в этой задаче вам может пригодиться контейнер `vector`; напоминаем, что добавление значения `x` в конец вектора `v` делается так:  
`v.push_back(x);`

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

Отметки

Примечания

Форумы обсуждений

Сообщения

Информация о курсе

## Неделя 2

Техника владения функциями и контейнерами

Вideo Готово

Материалы для самостоятельного изучения Осталось 42 мин

Тренировочные упражнения Осталось 9 ч 20 м

## ОБЯЗАТЕЛЬНО

	ОЦЕНКА	НЕОБХОДИМО СДАТЬ
✓ Задание по программированию Факториал 20 мин	100 %	19 июля г. 12:59 +06
✓ Задание по программированию Отбор палиндромов 20 мин	100 %	19 июля г. 12:59 +06
✓ Тест Функции 30 мин	97,22 %	19 июля г. 12:59 +06
✓ Задание по программированию Перемещение строк 20 мин	100 %	19 июля г. 12:59 +06
✓ Задание по программированию Разворот последовательности 20 мин	100 %	19 июля г. 12:59 +06
✓ Тест Векторы 25 мин	100 %	19 июля г. 12:59 +06
✓ Задание по программированию Очередь 22 мин	100 %	19 июля г. 12:59 +06
✓ Задание по программированию Ежемесячные дела 3 ч 32 м	100 %	19 июля г. 12:59 +06





## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

## Отметки

## Примечания

## Форумы обсуждений

## Сообщения

## Информация о курсе

- |                               |                             |       |            |
|-------------------------------|-----------------------------|-------|------------|
| ✓ Задание по программированию | Разворот последовательности | 100 % | 19 июля г. |
|                               | 20 мин                      |       | 12:59 +06  |
| ✓ Тест                        | Векторы                     | 100 % | 19 июля г. |
|                               | 25 мин                      |       | 12:59 +06  |
| ✓ Задание по программированию | Очередь                     | 100 % | 19 июля г. |
|                               | 22 мин                      |       | 12:59 +06  |
| ✓ Задание по программированию | Ежемесячные дела            | 100 % | 19 июля г. |
|                               | 3 ч 32 м                    |       | 12:59 +06  |
| ✓ Тест                        | Словари                     | 75 %  | 19 июля г. |
|                               | 25 мин                      |       | 12:59 +06  |
| ✓ Задание по программированию | Автобусные остановки — 1    | 100 % | 19 июля г. |
|                               | 1 ч 56 м                    |       | 12:59 +06  |
| ✓ Задание по программированию | Автобусные остановки — 2    | 100 % | 19 июля г. |
|                               | 20 мин                      |       | 12:59 +06  |
| ✓ Тест                        | Множества                   | 75 %  | 19 июля г. |
|                               | 15 мин                      |       | 12:59 +06  |
| ✓ Задание по программированию | Синонимы                    | 100 % | 19 июля г. |
|                               | 1 ч 20 м                    |       | 12:59 +06  |
| ✓ Задание по программированию | Автобусные остановки — 3    | 100 % | 19 июля г. |
|                               | 20 мин                      |       | 12:59 +06  |





## Задание по программированию: Факториал

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции**

Моя работа

Обсуждения

Напишите функцию, которая

- называется Factorial
- возвращает int
- принимает int и возвращает факториал своего аргумента.  
Гарантируется, что аргумент функции по модулю не превышает 10. Для отрицательных аргументов функция должна возвращать 1.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

Аргумент функции	Результат
1	1
-2	1
4	24



## Тренировочное задание по программированию: Палиндром

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Напишите функцию, которая

- называется `IsPalindrom`
- возвращает `bool`
- принимает параметр типа `string` и возвращает, является ли переданная строка палиндромом

Палиндром - это слово или фраза, которые одинаково читаются слева направо и справа налево. Пустая строка **является** палиндромом.

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.

Аргумент функции	Результат
madam	true
gentleman	false
X	true



## Задание по программированию: Отбор палиндромов

✓ Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 19 июля г., 12:59 +06

[Инструкции](#)    [Моя работа](#)

[Обсуждения](#)

Напишите функцию, которая

- называется PalindromFilter
- возвращает vector<string>
- принимает vector<string> words и int minLength и возвращает все строки из вектора words, которые являются палиндромами и имеют длину не меньше minLength

Входной вектор содержит не более 100 строк, длина каждой строки не больше 100 символов.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Примеры

words	minLength	результат
abacaba, aba	5	abacaba
abacaba, aba	2	abacaba, aba
weew, bro, code	4	weew

На проверку пришлите файл с реализацией функции PalindromFilter. Ваш файл может содержать и другие функции, если они вам необходимы. В этой



## Тренировочное задание по программированию: Максимизатор

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Напишите функцию `UpdateIfGreater`, которая принимает два целочисленных аргумента: `first` и `second`. Если `first` оказался больше `second`, Ваша функция должна записывать в `second` значение параметра `first`. При этом указанная функция не должна ничего возвращать, а изменение параметра `second` должно быть видно на вызывающей стороне.

Пример

```
1 int a = 4;
2 int b = 2;
3 UpdateIfGreater(a, b);
4 // b должно стать равно 4
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

Замечание

В данной задаче вам нужно самостоятельно выбирать типы для параметров функции `UpdateIfGreater`. Подумайте, какой из них обязан быть ссылкой, а какой может быть просто `int`. При этом учтите разные варианты вызова функции:

- `UpdateIfGreater(x, y);`
- `UpdateIfGreater(5, y);`



## Задание по программированию: Перемещение строк

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

[Инструкции](#) [Моя работа](#)

[Обсуждения](#)

Напишите функцию `MoveStrings`, которая принимает два вектора строк, `source` и `destination`, и дописывает все строки из первого вектора в конец второго. После выполнения функции вектор `source` должен оказаться пустым.

Чтобы очистить содержимое вектора, надо вызывать у него метод `clear`:

```
1 vector<string> words = {"hey", "hi"};
2 words.clear();
3 // Теперь вектор words пуст
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Пример

```
1 vector<string> source = {"a", "b", "c"};
2 vector<string> destination = {"z"};
3 MoveStrings(source, destination);
4 // source должен оказаться пустым
5 // destination должен содержать строки "z", "a", "b", "c" им
```



## Задание по программированию: Разворот последовательности

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

Инструкции

Моя работа

Обсуждения

Реализуйте функцию void Reverse(vector<int>& v), которая переставляет элементы вектора в обратном порядке.

**Пример**

```
1 vector<int> numbers = {1, 5, 3, 4, 2};  
2 Reverse(numbers);  
3 // numbers должен оказаться равен {2, 4, 3, 5, 1}
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Тренировочное задание по программированию: Разворот последовательности — 2

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

### Инструкции

Моя работа

Обсуждения

Реализуйте функцию `vector<int> Reversed(const vector<int>& v)`,  
возвращающую копию вектора `v`, в которой числа переставлены в  
обратном порядке.

### Пример

Вектор v	Результат функции
1, 5, 3, 4, 2	2, 4, 3, 5, 1

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.



## Тренировочное задание по программированию: Средняя температура

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Даны значения температуры, наблюдавшиеся в течение  $N$  подряд идущих дней. Найдите номера дней (в нумерации с нуля) со значением температуры выше среднего арифметического за все  $N$  дней.

Гарантируется, что среднее арифметическое значений температуры является целым числом.

### Формат ввода

Вводится число  $N$ , затем  $N$  неотрицательных целых чисел — значения температуры в 0-й, 1-й, ... ( $N-1$ )-й день.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Формат ввода

Вводится число  $N$ , затем  $N$  неотрицательных целых чисел — значения температуры в 0-й, 1-й, ... ( $N-1$ )-й день.

## Формат вывода

Первое число  $K$  — количество дней, значение температуры в которых выше среднего арифметического. Затем  $K$  целых чисел — номера этих дней.

## Пример

### Ввод

```
1 5
2 7 6 3 0 9
```

### Выход

```
1 3
2 0 1 4
```



## Задание по программированию: Очередь

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

### Инструкции

### Моя работа

### Обсуждения

Люди стоят в очереди, но никогда не уходят из её начала, зато могут приходить в конец и уходить оттуда. Более того, иногда некоторые люди могут прекращать и начинать беспокоиться из-за того, что очередь не продвигается.

Будем считать, что люди в очереди нумеруются целыми числами, начиная с 0.

Реализуйте обработку следующих операций над очередью:

- **WORRY *i***: пометить *i*-го человека с начала очереди как беспокоящегося;
- **QUIET *i***: пометить *i*-го человека как успокоившегося;
- **COME *k***: добавить *k* спокойных человек в конец очереди;
- **COME-*k***: убрать *k* человек из конца очереди;
- **WORRY\_COUNT**: узнать количество беспокоящихся людей в очереди.

Изначально очередь пуста.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Формат ввода

Количество операций  $Q$ , затем описания операций.

Для каждой операции **WORRY  $i$**  и **QUIET  $i$**  гарантируется, что человек с номером  $i$  существует в очереди на момент операции.

Для каждой операции **COME - $k$**  гарантируется, что  $k$  не больше текущего размера очереди.

## Формат вывода

Для каждой операции **WORRY\_COUNT** выведите одно целое число — количество беспокоящихся людей в очереди.

## Пример

### Ввод

```
1 8
2 COME 5
3 WORRY 1
4 WORRY 4
5 COME -2
6 WORRY_COUNT
7 COME 3
8 WORRY 3
9 WORRY_COUNT
10
```

### Выход

```
1 1
2 2
```



## Задание по программированию: Ежемесячные дела

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

У каждого из нас есть повторяющиеся ежемесячные дела, каждое из которых нужно выполнять в конкретный день каждого месяца: оплата счетов за электричество, абонентская плата за связь и пр. Вам нужно реализовать работу со списком таких дел на месяц, а именно, реализовать набор следующих операций:

### ADD *i s*

Назначить дело с названием *s* на день *i* текущего месяца.

### DUMP *i*

Вывести все дела, запланированные на день *i* текущего месяца.

### NEXT

Перейти к списку дел на новый месяц. При выполнении данной команды вместо текущего (старого) списка дел на текущий месяц создаётся и становится активным (новым) список дел на следующий месяц: все дела со старого списка дел копируются в новый список. После выполнения данной команды новый список дел и следующий месяц становятся текущими, а работа со старым списком дел прекращается. При переходе к новому месяцу необходимо обратить внимание на разное количество дней в месяцах:

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Моя работа'.



## NEXT

Перейти к списку дел на новый месяц. При выполнении данной команды вместо текущего (старого) списка дел на текущий месяц создаётся и становится активным (новый) список дел на следующий месяц; все дела со старого списка дел копируются в новый список. После выполнения данной команды новый список дел и следующий месяц становятся текущими, а работа со старым списком дел прекращается. При переходе к новому месяцу необходимо обратить внимание на разное количество дней в месяцах:

- если следующий месяц имеет больше дней, чем текущий, «дополнительные» дни необходимо оставить пустыми (не содержащими дел);
- если следующий месяц имеет меньше дней, чем текущий, дела со всех «лишних» дней необходимо переместить на последний день следующего месяца.

## Замечания

- Историю списков дел хранить **не требуется**, работа ведется только с текущим списком дел текущего месяца. Более того, при создании списка дел на следующий месяц, он **«перетирает»** предыдущий список.
- Обратите внимание, что количество команд **NEXT** в общей последовательности команд при работе со списком дел может превышать 11.
- Начальным текущим месяцем считается январь.
- Количество дней в месяцах соответствуют Григорианскому календарю с той лишь разницей, что в феврале всегда 28 дней.

## Формат ввода

Сначала число операций  $Q$ , затем описания операций.



## Формат ввода

Сначала число операций  $Q$ , затем описания операций.

Названия дел  $s$  уникальны и состоят только из латинских букв, цифр и символов подчёркивания. Номера дней  $i$  являются целыми числами и нумеруются от 1 до размера текущего месяца.

## Формат вывода

Для каждой операции типа **DUMP** в отдельной строке выведите количество дел в соответствующий день, а затем их названия, разделяя их пробелом. Порядок вывода дел в рамках каждой операции значения не имеет.

## Пример

### Ввод

```
1 12
2 ADD 5 Salary
3 ADD 31 Walk
4 ADD 30 WalkPreparations
5 NEXT
6 DUMP 5
7 DUMP 28
8 NEXT
9 DUMP 31
10 DUMP 30
11 DUMP 28
12 ADD 28 Payment
13 DUMP 28
14
```

### Выход



## Вывод

```
1 1 Salary
2 2 WalkPreparations Walk
3 0
4 0
5 2 WalkPreparations Walk
6 3 WalkPreparations Walk Payment
7
```

## Указание

Для дописывания всех элементов вектора v2 в конец вектора v1 удобно использовать метод insert:

```
1 v1.insert(end(v1), begin(v2), end(v2));
```

Кроме того, элементом вектора может быть любой тип, в том числе и другой вектор. Например, `vector<vector<int>>` — это вектор, элементами которого являются вектора целых чисел (то есть двумерный массив). Пример использования:

```
1 vector<vector<int>> m(10); // Создаём вектор из десяти вект
2 m[0].push_back(5); // Добавляем элементы в первый вектор
3 m[0].push_back(15);
4 cout << m[0][1]; // Выведет 15 – второй элемент первого вект
5
6 m[1].push_back(3);
7 for (int x : m[1]) {
8     // Перебираем все элементы второго
9 }
```



## Тренировочное задание по программированию: Анаграммы

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Слова называются *анаграммами* друг друга, если одно из них можно получить перестановкой букв в другом. Например, слово «eat» можно получить перестановкой букв слова «tea», поэтому эти слова являются анаграммами друг друга. Даны пары слов, проверьте для каждой из них, являются ли слова этой пары анаграммами друг друга.

### Указание

Один из способов проверки того, являются ли слова анаграммами друг друга, заключается в следующем. Для каждого слова с помощью словаря подсчитаем, сколько раз в нём встречается каждая буква. Если для обоих слов эти словари равны (это проверяется с помощью обычного оператора `==`), то слова являются анаграммами друг друга, в противном случае не являются.

При этом построение такого словаря по слову удобно вынести в отдельную функцию `BuildCharCounters`.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Формат ввода

Сначала дано число пар слов  $N$ , затем в  $N$  строках содержатся пары слов, которые необходимо проверить. Гарантируется, что все слова состоят лишь из строчных латинских букв.

## Формат вывода

Выведите  $N$  строк: для каждой введённой пары слов **YES**, если эти слова являются анаграммами, и **NO** в противном случае.

## Пример

### Ввод

```
1  3
2  eat tea
3  find search
4  master stream
5
```

### Выход

```
1  YES
2  NO
3  YES
4
```



## Тренировочное задание по программированию: Справочник столиц

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

[Инструкции](#)    [Моя работа](#)

[Обсуждения](#)

Реализуйте справочник столиц стран.

На вход программе поступают следующие запросы:

- **CHANGE\_CAPITAL *country* *new\_capital*** — изменение столицы страны *country* на *new\_capital*, либо добавление такой страны с такой столицей, если раньше её не было.
- **RENAME *old\_country\_name* *new\_country\_name*** — переименование страны из *old\_country\_name* в *new\_country\_name*.
- **ABOUT *country*** — вывод столицы страны *country*.
- **DUMP** — вывод столиц всех стран.

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.

### Формат ввода

В первой строке содержится количество запросов *Q*, в следующих *Q* строках — описания запросов. Все названия стран и столиц состоят лишь из латинских букв, цифр и символов подчёркивания.



## Формат ввода

В первой строке содержится количество запросов  $Q$ , в следующих  $Q$  строках — описания запросов. Все названия стран и столиц состоят лишь из латинских букв, цифр и символов подчёркивания.

## Формат вывода

Выведите результат обработки каждого запроса:

В ответ на запрос **CHANGE\_CAPITAL** *country new\_capital* выведите

- **Introduce new country** *country* **with capital** *new\_capital*, если страны *country* раньше не существовало;
- **Country** *country* **hasn't changed its capital**, если страна *country* до текущего момента имела столицу *new\_capital*;
- **Country** *country* **has changed its capital from** *old\_capital* **to** *new\_capital*, если страна *country* до текущего момента имела столицу *old\_capital*, название которой не совпадает с названием *new\_capital*.

В ответ на запрос **RENAME** *old\_country\_name new\_country\_name* выведите

- **Incorrect rename, skip**, если новое название страны совпадает со старым, страна *old\_country\_name* не существует или страна *new\_country\_name* уже существует;
- **Country** *old\_country\_name* **with capital** *capital* **has been renamed to** *new\_country\_name*, если запрос корректен и страна имеет столицу *capital*.

В ответ на запрос **ABOUT** *country* выведите

- **Country** *country* **doesn't exist**, если страны с названием *country* не



В ответ на запрос **ABOUT country** выводите

- **Country *country* doesn't exist.**, если страны с названием *country* не существует;
- **Country *country* has capital *capital*.**, если страна *country* существует и имеет столицу *capital*.

В ответ на запрос **DUMP** выводите

- **There are no countries in the world.**, если пока не было добавлено ни одной страны;
- последовательность пар вида ***country/capital***, описывающую столицы всех стран, если в мире уже есть хотя бы одна страна. При выводе последовательности пары указанного вида необходимо упорядочить по названию страны и разделять между собой пробелом.

### Пример 1

Ввод

```
1 6
2 CHANGE_CAPITAL RussianEmpire Petrograd
3 RENAME RussianEmpire RussianRepublic
4 ABOUT RussianRepublic
5 RENAME RussianRepublic USSR
6 CHANGE_CAPITAL USSR Moscow
7 DUMP
8
```

Выход

```
1 Introduce new country RussianEmpire with capital Petrograd
2 Country RussianEmpire with capital Petrograd has been rename
```

## Вывод

```
1 Introduce new country RussianEmpire with capital Petrograd
2 Country RussianEmpire with capital Petrograd has been rename
3 Country RussianRepublic has capital Petrograd
4 Country RussianRepublic with capital Petrograd has been renam
5 Country USSR has changed its capital from Petrograd to Mosco
6 USSR/Moscow
7
```

## Пример 2

## Ввод

```
1 24
2 RENAME FakeCountry FarFarAway
3 ABOUT FarFarAway
4 DUMP
5 CHANGE_CAPITAL TsardomOfRussia Moscow
6 CHANGE_CAPITAL TsardomOfRussia Moscow
7 CHANGE_CAPITAL ColonialBrazil Salvador
8 CHANGE_CAPITAL TsardomOfRussia SaintPetersburg
9 RENAME TsardomOfRussia RussianEmpire
10 CHANGE_CAPITAL RussianEmpire Moscow
11 CHANGE_CAPITAL RussianEmpire SaintPetersburg
12 CHANGE_CAPITAL ColonialBrazil RioDeJaneiro
13 DUMP
14 RENAME ColonialBrazil EmpireOfBrazil
15 ABOUT RussianEmpire
16 RENAME EmpireOfBrazil UnitedStatesOfBrazil
17 CHANGE_CAPITAL RussianEmpire Petrograd
18 RENAME RussianEmpire RussianRepublic
19 RENAME RussianRepublic USSR
20 CHANGE_CAPITAL USSR Moscow
21 CHANGE_CAPITAL UnitedStatesOfBrazil Brasilia
22 RENAME UnitedStatesOfBrazil FederativeRepublicOfBrazil
23 ABOUT RussianEmpire
24 DUMP
25 RENAME USSR USSR
26
```

Выход

1 Incorrect rename, skip  
2 Country FarFarAway doesn't exist  
3 There are no countries in the world  
4 Introduce new country TsardomOfRussia with capital Moscow  
5 Country TsardomOfRussia hasn't changed its capital  
6 Introduce new country ColonialBrazil with capital Salvador  
7 Country TsardomOfRussia has changed its capital from Moscow  
8 Country TsardomOfRussia with capital SaintPetersburg has been  
9 Country RussianEmpire has changed its capital from SaintPete  
10 Country RussianEmpire has changed its capital from Moscow  
11 Country ColonialBrazil has changed its capital from Salvador  
12 ColonialBrazil/RioDeJaneiro RussianEmpire/SaintPetersburg  
13 Country ColonialBrazil with capital RioDeJaneiro has been re  
14 Country RussianEmpire has capital SaintPetersburg  
15 Country EmpireOfBrazil with capital RioDeJaneiro has been re  
16 Country RussianEmpire has changed its capital from SaintPete  
17 Country RussianEmpire with capital Petrograd has been renamed  
18 Country RussianRepublic with capital Petrograd has been renamed  
19 Country USSR has changed its capital from Petrograd to Moscow  
20 Country UnitedStatesOfBrazil has changed its capital from Rio  
21 Country UnitedStatesOfBrazil with capital Brasilia has been  
22 Country RussianEmpire doesn't exist  
23 FederativeRepublicOfBrazil/Brasilia USSR/Moscow  
24 Incorrect rename, skip  
25

### Пример 3

Ввод

```
1 10
2 CHANGE_CAPITAL RussianEmpire Petrograd
3 RENAME USSR RussianEmpire
4 DUMP
5 ABOUT USSR
6 DUMP
7 RENAME RussianEmpire RussianEmpire
8 DUMP
```



## Пример 3

## Ввод

```
1 10
2 CHANGE_CAPITAL RussianEmpire Petrograd
3 RENAME USSR RussianEmpire
4 DUMP
5 ABOUT USSR
6 DUMP
7 RENAME RussianEmpire RussianEmpire
8 DUMP
9 CHANGE_CAPITAL UnitedStatesOfBrazil Brasilia
10 RENAME RussianEmpire UnitedStatesOfBrazil
11 DUMP
12
```

## Выход

```
1 Introduce new country RussianEmpire with capital Petrograd
2 Incorrect rename, skip
3 RussianEmpire/Petrograd
4 Country USSR doesn't exist
5 RussianEmpire/Petrograd
6 Incorrect rename, skip
7 RussianEmpire/Petrograd
8 Introduce new country UnitedStatesOfBrazil with capital Bras
9 Incorrect rename, skip
10 RussianEmpire/Petrograd UnitedStatesOfBrazil/Brasilia
11
```



## Задание по программированию: Автобусные остановки — 1

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Реализуйте систему хранения автобусных маршрутов. Вам нужно обрабатывать следующие запросы:

- **NEW\_BUS bus stop\_count stop1 stop2 ...** — добавить маршрут автобуса с названием **bus** и **stop\_count** остановками с названиями **stop1**, **stop2**, ...
- **BUSES\_FOR\_STOP stop** — вывести названия всех маршрутов автобуса, проходящих через остановку **stop**.
- **STOPS\_FOR\_BUS bus** — вывести названия всех остановок маршрута **bus** со списком автобусов, на которые можно пересесть на каждой из остановок.
- **ALL\_BUSES** — вывести список всех маршрутов с остановками.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Формат ввода

В первой строке ввода содержится количество запросов **Q**, затем в **Q** строках следуют описания запросов.

Гарантируется, что все названия маршрутов и остановок состоят лишь из латинских букв, цифр и знаков подчёркивания.

Для каждого запроса **NEW\_BUS bus stop\_count stop1 stop2 ...** гарантируется, что маршрут **bus** отсутствует, количество остановок больше 0, а после числа



Гарантируется, что все названия маршрутов и остановок состоят лишь из латинских букв, цифр и знаков подчёркивания.

Для каждого запроса **NEW\_BUS bus stop\_count stop1 stop2...** гарантируется, что маршрут **bus** отсутствует, количество остановок больше 0, а после числа **stop\_count** следует именно такое количество названий остановок, причём все названия в каждом списке различны.

### Формат вывода

Для каждого запроса, кроме **NEW\_BUS**, выведите соответствующий ответ на него:

- На запрос **BUSES\_FOR\_STOP stop** выведите через пробел список автобусов, проезжающих через эту остановку, в том порядке, в котором они создавались командами **NEW\_BUS**. Если остановка **stop** не существует, выведите **No stop**.
- На запрос **STOPS\_FOR\_BUS bus** выведите описания остановок маршрута **bus** в отдельных строках в том порядке, в котором они были заданы в соответствующей команде **NEW\_BUS**. Описание каждой остановки **stop** должно иметь вид **Stop stop: bus1 bus2...**, где **bus1** **bus2...** — список автобусов, проезжающих через остановку **stop**. В порядке, в котором они создавались командами **NEW\_BUS**, за исключением исходного маршрута **bus**. Если через остановку **stop** не проезжает ни один автобус, кроме **bus**, вместо списка автобусов для неё выведите **no interchange**. Если маршрут **bus** не существует, выведите **No bus**.
- На запрос **ALL\_BUSES** выведите описания всех автобусов в алфавитном порядке. Описание каждого маршрута **bus** должно иметь вид **Bus bus: stop1 stop2...**, где **stop1 stop2...** — список остановок автобуса **bus** в порядке, в котором они были заданы в соответствующей команде **NEW\_BUS**. Если автобусы отсутствуют, выведите **No buses**.

### Предупреждение

## Предупреждение

Условие задачи выше содержит много важных деталей. Если вы не уверены в том, что не упустили ни одной, перечитайте условие ещё раз.

## Пример

### Ввод

```
1 10
2 ALL_BUSES
3 BUSES_FOR_STOP Marushkino
4 STOPS_FOR_BUS 32K
5 NEW_BUS 32 3 Tolstopaltsevo Marushkino Vnukovo
6 NEW_BUS 32K 6 Tolstopaltsevo Marushkino Vnukovo Peredelkino
7 BUSES_FOR_STOP Vnukovo
8 NEW_BUS 950 6 Kokoshkino Marushkino Vnukovo Peredelkino Solntsevo
9 NEW_BUS 272 4 Vnukovo Moskovsky Rumyantsevo Troparyovo
10 STOPS_FOR_BUS 272
11 ALL_BUSES
```

### Выход

```
1 No buses
2 No stop
3 No bus
4 32 32K
5 Stop Vnukovo: 32 32K 950
6 Stop Moskovsky: no interchange
7 Stop Rumyantsevo: no interchange
8 Stop Troparyovo: 950
9 Bus 272: Vnukovo Moskovsky Rumyantsevo Troparyovo
10 Bus 32: Tolstopaltsevo Marushkino Vnukovo Peredelkino Solntsevo
11 Bus 32K: Tolstopaltsevo Marushkino Vnukovo Peredelkino Solntsevo
12 Bus 950: Kokoshkino Marushkino Vnukovo Peredelkino Solntsevo
```



## Задание по программированию: Автобусные остановки — 2

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции**

Моя работа

Обсуждения

В этой задаче вам нужно присваивать номера автобусным маршрутам.

А именно, для каждого маршрута, заданного набором названий остановок, нужно либо выдать новый номер (первому маршруту — 1, второму — 2 и т. д.), либо вернуть номер существующего маршрута, которому соответствует такой набор остановок.

Наборы остановок, полученные друг из друга перестановкой остановок, считаются различными (см. пример).

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Указание

В C++ ключом словаря может быть не только число или строка, но и другой контейнер, например, vector.

### Формат ввода

Сначала вводится количество запросов  $Q$ , затем  $Q$  описаний запросов.

Каждый запрос представляет собой положительное количество остановок  $N$ , за которым следуют разделённые пробелом  $N$  различных названий остановок соответствующего маршрута. Названия остановок состоят лишь из латинских букв и символов подчёркивания.



## Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

Если маршрут с данным набором остановок уже существует, в ответ на соответствующий запрос выведите **Already exists for *i***, где *i* — номер маршрута с таким набором остановок. В противном случае нужно выделить введённому набору остановок новый номер *i* и вывести его в формате **New bus *i***.

## Пример

### Ввод

```
1 4
2 2 Marushkino Kokoshkino
3 1 Kokoshkino
4 2 Marushkino Kokoshkino
5 2 Kokoshkino Marushkino
6
```

### Выход

```
1 New bus 1
2 New bus 2
3 Already exists for 1
4 New bus 3
5
```

## Замечание

Будьте внимательны при обновлении словаря: если одновременно в одном



```
4 New bus 3  
5
```

### Замечание

Будьте внимательны при обновлении словаря: если одновременно в одном выражении вычислять размер словаря и добавлять в него новый элемент, результат может отличаться от ожидаемого на единицу. Чтобы этого избежать, используйте временную переменную.

Например, вместо кода

```
1 map<string, int> m;  
2 // ...  
3 m["two"] = m.size();  
4 // что нужно сделать раньше: вычислить size или добавить нов
```

стоит написать такой код:

```
1 map<string, int> m;  
2 // ...  
3 int new_value = m.size();  
4 m["two"] = new_value;  
5 // теперь всё однозначно: нужен размер до добавления элемент
```



## Тренировочное задание по программированию: Количество уникальных строк

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Дан набор строк. Найдите количество уникальных строк в этом наборе.

### Формат ввода

Сначала вводится количество строк  $N$ , затем — сами  $N$  строк, разделённые символом перевода строки. Все строки состоят лишь из латинских букв, цифр и символов подчёркивания.

### Формат вывода

Выведите единственное целое число — количество уникальных строк в данном наборе.

### Пример

#### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Формат вывода

Выведите единственное целое число — количество уникальных строк в данном наборе.

### Пример

#### Ввод

```
1 6
2 first
3 second
4 first
5 third
6 second
7 second
8
```

#### Выход

```
1 3
```

[Основы разработки на C++: белый пояс](#) > [Неделя 2](#) > [Множество значений словаря](#)

&lt; Предыдущий Дальше &gt;

## Тренировочное задание по программированию: Множество значений словаря

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

[Инструкции](#)[Моя работа](#)[Обсуждения](#)

Напишите функцию BuildMapViewSet, принимающую на вход словарь `map<int, string>` и возвращающую множество значений этого словаря:

```
1 set<string> BuildMapViewSet(const map<int, string>& m) {  
2     // ...  
3 }
```

### Пример

[Код](#)

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Пример

### Код

```
1  set<string> values = BuildMapValuesSet({  
2      {1, "odd"},  
3      {2, "even"},  
4      {3, "odd"},  
5      {4, "even"},  
6      {5, "odd"}  
7  });  
8  
9  for (const string& value : values) {  
10     cout << value << endl;  
11 }
```

### Вывод

```
1  even  
2  odd
```

## Пояснение

В этой задаче на проверку вам надо прислать файл с реализацией функции `BuildMapValuesSet`. **Этот файл не должен содержать функцию main**. Если в нём будет функция `main`, вы получите ошибку компиляции.

## Задание по программированию: Синонимы

 Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 19 июля г., 12:59 +06

### Инструкции

### Моя работа

### Обсуждения

Два слова называются *синонимами* друг друга, если они имеют похожие значения. Реализуйте следующие операции над словарём синонимов:

- **ADD *word1 word2*** — добавить в словарь пару синонимов (*word1*, *word2*).
- **COUNT *word*** — узнать количество синонимов слова *word*.
- **CHECK *word1 word2*** — проверить, являются ли слова *word1* и *word2* синонимами. Слова *word1* и *word2* считаются синонимами, если среди запросов **ADD** был хотя бы один запрос **ADD *word1 word2*** или **ADD *word2 word1***.

### Замечание

Для упрощения, будем считать, что синонимы не обладают транзитивностью, то есть, если **A** - синоним **B**, а **B** - синоним **C**, то из этого *НЕ* следует, что **A** - синоним **C**.

### Формат ввода

Сначала вводится количество запросов *Q*, затем *Q* строк с описаниями запросов. Гарантируется, что в каждом запросе **CHECK** и **ADD** слова *word1* и *word2* различны. Все слова состоят лишь из латинских букв, цифр и символов подчёркивания.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Формат вывода

Для каждого запроса в соответствующей строке выведите ответ на него:

- В ответ на запрос **COUNT word** выведите единственное целое число — количество синонимов слова **word**.
- В ответ на запрос **CHECK word1 word2** выведите строку **YES**, если **word1** и **word2** являются синонимами, и **NO** в противном случае.

## Пример

### Ввод

```
1 8
2 ADD program code
3 COUNT cipher
4 ADD code cipher
5 COUNT code
6 COUNT program
7 CHECK code program
8 CHECK program cipher
9 CHECK cpp java
10
```

### Выход

```
1 0
2 2
3 1
4 YES
5 NO
6 NO
7
```



## Задание по программированию: Автобусные остановки — 3

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 19 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

В этой задаче вам нужно присваивать номера автобусным маршрутам.

А именно, для каждого маршрута, заданного множеством названий остановок, нужно либо выдать новый номер (первому маршруту — 1, второму — 2 и т. д.), либо вернуть номер существующего маршрута, которому соответствует такое множество остановок.

В отличие от задачи «Автобусные остановки — 2», наборы остановок, которые можно получить друг из друга перестановкой элементов или добавлением/удалением повторяющихся, следует считать одинаковыми.

### Формат ввода

Сначала вводится количество запросов  $Q$ , затем  $Q$  описаний запросов.

Каждый запрос представляет собой положительное количество остановок  $N$ , за которым следуют разделённые пробелом  $N$  названий остановок соответствующего маршрута (не обязательно различных). Названия остановок состоят лишь из латинских букв и символов подчёркивания.

### Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Моя работа'.



## Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

Если маршрут с данным набором остановок уже существует, в ответ на соответствующий запрос выведите **Already exists for *i***, где *i* — номер маршрута с таким набором остановок. В противном случае нужно выделить введённому набору остановок новый номер *i* и вывести его в формате **New bus *i***.

## Пример

Ввод

```
1 5
2 2 Marushkino Kokoshkino
3 1 Kokoshkino
4 2 Marushkino Kokoshkino
5 2 Kokoshkino Marushkino
6 2 Kokoshkino Kokoshkino
7
```

Выход

```
1 New bus 1
2 New bus 2
3 Already exists for 1
4 Already exists for 1
5 Already exists for 2
6
```



## Замечание

Будьте внимательны при обновлении словаря: если одновременно в одном выражении вычислять размер словаря и добавлять в него новый элемент, результат может отличаться от ожидаемого на единицу. Чтобы этого избежать, используйте временную переменную.

Например, вместо кода

```
1 map<string, int> m;
2 // ...
3 m["two"] = m.size();
4 // что нужно сделать раньше: вычислить size или добавить нов
5
```

стоит написать такой код:

```
1 map<string, int> m;
2 // ...
3 int new_value = m.size();
4 m["two"] = new_value;
5 // теперь всё однозначно: нужен размер до добавления элемент
6
```



## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

Отметки

Примечания

Форумы обсуждений

Сообщения

Информация о курсе

## Неделя 3

Ожидаемое время: 31 мин ^

Медитация над алгоритмами и классами

Вideo Готово

Материалы для  
самостоятельного изучения Осталось  
10 мин

Тренировочные упражнения Готово

ОБЯЗАТЕЛЬНО

ОЦЕНКА

НЕОБХОДИМО  
СДАТЬ✓ Тест  
Алгоритмы  
30 мин75 % 26 июля г.  
12:59 +06✓ Задание по программированию  
Сортировка целых чисел по мо...  
20 мин100 % 26 июля г.  
12:59 +06✓ Задание по программированию  
Сортировка без учёта регистра  
20 мин100 % 26 июля г.  
12:59 +06✓ Тест  
Видимость и инициализация пе...  
30 мин100 % 26 июля г.  
12:59 +06✓ Тест  
Введение в структуры и классы  
20 мин88,88 % 26 июля г.  
12:59 +06✓ Задание по программированию  
Имена и фамилии — 1  
1 ч 19 м100 % 26 июля г.  
12:59 +06



## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

Отметки

Примечания

Форумы обсуждений

Сообщения

Информация о курсе

Видимость и инициализация пе...

30 мин



Тест

12:59 +06

Введение в структуры и классы

20 мин

88,88 % 26 июля г.  
12:59 +06

Задание по программированию

100 % 26 июля г.  
12:59 +06

Имена и фамилии — 1

1 ч 19 м



Задание по программированию

100 % 26 июля г.  
12:59 +06

Имена и фамилии — 2

47 мин



Тест

87,50 % 26 июля г.  
12:59 +06

Константность методов

15 мин



Тест

87,50 % 26 июля г.  
12:59 +06

Конструкторы

10 мин



Задание по программированию

0 % 26 июля г.  
12:59 +06

Имена и фамилии — 3

21 мин



Тест

100 % 26 июля г.  
12:59 +06

Деструкторы и время жизни об...

35 мин





## Задание по программированию: Сортировка целых чисел по модулю

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

[Инструкции](#)

[Моя работа](#)

[Обсуждения](#)

### Условие

В стандартном потоке дана одна строка, состоящая из  $N + 1$  целых чисел. Первым числом идёт само число  $N$ . Далее следуют ещё  $N$  чисел, обозначим их за массив  $A$ . Между собой числа разделены пробелом.

Отсортируйте массив  $A$  по модулю и выведите его в стандартный поток.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Ограничения

- $0 \leq N \leq 1000$
- $-1000000 \leq A[i] \leq 1000000$

### Примеры

stdin	stdout
2 -4 3	3 -4
3 1 -3 2	1 2 -3



## Задание по программированию: Сортировка без учёта регистра

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

[Инструкции](#)    [Моя работа](#)

[Обсуждения](#)

### Условие

В стандартном потоке дана одна строка, состоящая из числа N и следующих за ним N строк S. Между собой число и строки разделены пробелом.

Отсортируйте строки S в лексикографическом порядке по возрастанию, игнорируя регистр букв, и выведите их в стандартный поток вывода.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Ограничения

- $0 \leq N \leq 1000$
- $1 \leq |S| \leq 15$
- Каждая строка  $S[i]$  может состоять из следующих символов: [0-9,a-z,A-Z]

### Примеры

stdin	stdout
2 qq Aa	Aa qq
3 aA Cc bb	aA bb Cc



## Тренировочное задание по программированию: Отсортированные строки

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Реализуйте класс, поддерживающий набор строк в отсортированном порядке. Класс должен содержать два публичных метода:

```
1 class SortedStrings {
2 public:
3     void AddString(const string& s) {
4         // добавить строку s в набор
5     }
6     vector<string> GetSortedStrings() {
7         // получить набор из всех добавленных строк в отсортиро
8     }
9 private:
10    // приватные поля
11 };
12
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

Пример



## Пример

## Код

```
1 void PrintSortedStrings(SortedStrings& strings) {
2     for (const string& s : strings.GetSortedStrings()) {
3         cout << s << " ";
4     }
5     cout << endl;
6 }
7
8 int main() {
9     SortedStrings strings;
10
11    strings.AddString("first");
12    strings.AddString("third");
13    strings.AddString("second");
14    PrintSortedStrings(strings);
15
16    strings.AddString("second");
17    PrintSortedStrings(strings);
18
19    return 0;
20 }
21
```

## Вывод

```
1 first second third
2 first second second third
3
```





# Задание по программированию: Имена и фамилии — 1

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Реализуйте класс для человека, поддерживающий историю изменений человеком своих фамилии и имени.

```
1 class Person {
2 public:
3     void ChangeFirstName(int year, const string& first_name) {
4         // добавить факт изменения имени на first_name в год уea
5     }
6     void ChangeLastName(int year, const string& last_name) {
7         // добавить факт изменения фамилии на last_name в год уе
8     }
9     string GetFullName(int year) {
10        // получить имя и фамилию по состоянию на конец года уе
11    }
12 private:
13     // приватные поля
14 };
15
```

## Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



Считайте, что в каждый год может произойти не более одного изменения фамилии и не более одного изменения имени. При этом с течением времени могут открываться все новые факты из прошлого человека, поэтому года в последовательных вызовах методов ChangeLastName и ChangeFirstName не обязаны возрастиать.

Гарантируется, что все имена и фамилии непусты.

Строка, возвращаемая методом GetFullName, должна содержать разделённые одним пробелом имя и фамилию человека по состоянию на конец данного года.

- Если к данному году не случилось ни одного изменения фамилии и имени, верните строку "**Incognito**".
- Если к данному году случилось изменение фамилии, но не было ни одного изменения имени, верните "**last\_name with unknown first name**".
- Если к данному году случилось изменение имени, но не было ни одного изменения фамилии, верните "**first\_name with unknown last name**".

## Пример

### Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
```

## Пример

## Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
7         cout << person.GetFullName(year) << endl;
8     }
9
10    person.ChangeFirstName(1970, "Appolinaria");
11    for (int year : {1969, 1970}) {
12        cout << person.GetFullName(year) << endl;
13    }
14
15    person.ChangeLastName(1968, "Volkova");
16    for (int year : {1969, 1970}) {
17        cout << person.GetFullName(year) << endl;
18    }
19
20    return 0;
21 }
22 }
```

## ВЫВОД

```
1 Incognito
2 Polina with unknown last name
3 Polina Sergeeva
4 Polina Sergeeva
5 Appolinaria Sergeeva
6 Polina Volkova
7 Appolinaria Volkova
8
```



## Задание по программированию: Имена и фамилии — 2

✓ Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 26 июля г., 12:59 +06

Инструкции Моя работа

Обсуждения

Дополните класс из предыдущей задачи «Имена и фамилии — 1» методом GetFullNameWithHistory:

```
1 class Person {
2 public:
3     void ChangeFirstName(int year, const string& first_name) {
4         // добавить факт изменения имени на first_name в год year
5     }
6     void ChangeLastName(int year, const string& last_name) {
7         // добавить факт изменения фамилии на last_name в год year
8     }
9     string GetFullName(int year) {
10        // получить имя и фамилию по состоянию на конец года year
11    }
12    string GetFullNameWithHistory(int year) {
13        // получить все имена и фамилии по состоянию на конец года year
14    }
15 private:
16     // приватные поля
17 };
18 }
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

В отличие от метода GetFullName, метод GetFullNameWithHistory должен вернуть не только последние имя и фамилию к концу данного года, но ещё и все предыдущие имена и фамилии в обратном хронологическом порядке. Если такущий факт говорят о том, что человек два раза поменял имена,



В отличие от метода GetFullName, метод GetFullNameWithHistory должен вернуть не только последние имя и фамилию к концу данного года, но ещё и все предыдущие имена и фамилии в обратном хронологическом порядке. Если текущие факты говорят о том, что человек два раза подряд изменил фамилию или имя на одно и то же, второе изменение при формировании истории нужно игнорировать.

Для лучшего понимания формата см. примеры.

## Пример 1

### Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1900, "Eugene");
5     person.ChangeLastName(1900, "Sokolov");
6     person.ChangeLastName(1910, "Sokolov");
7     person.ChangeFirstName(1920, "Evgeny");
8     person.ChangeLastName(1930, "Sokolov");
9     cout << person.GetFullNameWithHistory(1940) << endl;
10
11     return 0;
12 }
13
```

### Выход

```
1 Evgeny (Eugene) Sokolov
2
```

## Пример 2

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
7         cout << person.GetFullNameWithHistory(year) << endl;
8     }
9
10    person.ChangeFirstName(1970, "Appolinaria");
11    for (int year : {1969, 1970}) {
12        cout << person.GetFullNameWithHistory(year) << endl;
13    }
14
15    person.ChangeLastName(1968, "Volkova");
16    for (int year : {1969, 1970}) {
17        cout << person.GetFullNameWithHistory(year) << endl;
18    }
19
20    person.ChangeFirstName(1990, "Polina");
21    person.ChangeLastName(1990, "Volkova-Sergeeva");
22    cout << person.GetFullNameWithHistory(1990) << endl;
23
24    person.ChangeFirstName(1966, "Pauline");
25    cout << person.GetFullNameWithHistory(1966) << endl;
26
27    person.ChangeLastName(1960, "Sergeeva");
28    for (int year : {1960, 1967}) {
29        cout << person.GetFullNameWithHistory(year) << endl;
30    }
31
32    person.ChangeLastName(1961, "Ivanova");
33    cout << person.GetFullNameWithHistory(1967) << endl;
34
35    return 0;
36 }
```



## Вывод

```
1 Incognito
2 Polina with unknown last name
3 Polina Sergeeva
4 Polina Sergeeva
5 Appolinaria (Polina) Sergeeva
6 Polina Volkova (Sergeeva)
7 Appolinaria (Polina) Volkova (Sergeeva)
8 Polina (Appolinaria, Polina) Volkova-Sergeeva (Volkova, Serg
9 Pauline (Polina) with unknown last name
10 Sergeeva with unknown first name
11 Pauline (Polina) Sergeeva
12 Pauline (Polina) Sergeeva (Ivanova, Sergeeva)
13
```

## Пример 3

## Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeFirstName(1965, "Appolinaria");
6
7     person.ChangeLastName(1965, "Sergeeva");
8     person.ChangeLastName(1965, "Volkova");
9     person.ChangeLastName(1965, "Volkova-Sergeeva");
10
11    for (int year : {1964, 1965, 1966}) {
12        cout << person.GetFullNameWithHistory(year) << endl;
13    }
14
15    return 0;
16 }
```



## Пример 3

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeFirstName(1965, "Appolinaria");
6
7     person.ChangeLastName(1965, "Sergeeva");
8     person.ChangeLastName(1965, "Volkova");
9     person.ChangeLastName(1965, "Volkova-Sergeeva");
10
11    for (int year : {1964, 1965, 1966}) {
12        cout << person.GetFullNameWithHistory(year) << endl;
13    }
14
15    return 0;
16 }
```

Вывод

```
1 Incognito
2 Appolinaria Volkova-Sergeeva
3 Appolinaria Volkova-Sergeeva
4
```



## Тренировочное задание по программированию: Стока

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

**Инструкции**

Моя работа

Обсуждения

Реализуйте класс ReversibleString, хранящий строку и поддерживающий методы Reverse для переворота строки и ToString для получения строки.

### Пример

#### Код

```
1 int main() {
2     ReversibleString s("live");
3     s.Reverse();
4     cout << s.ToString() << endl;
5
6     s.Reverse();
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Пример

## Код

```
1 int main() {
2     ReversibleString s("live");
3     s.Reverse();
4     cout << s.ToString() << endl;
5
6     s.Reverse();
7     const ReversibleString& s_ref = s;
8     string tmp = s_ref.ToString();
9     cout << tmp << endl;
10
11    ReversibleString empty;
12    cout << "" << empty.ToString() << "" << endl;
13
14    return 0;
15 }
16
```

Вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

## Вывод

```
1 evil
2 live
3 ""
4
```



## Задание по программированию: Имена и фамилии — 3

Повторить попытку · Получено баллов: 0/1  
Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Дополните класс Person из задачи «Имена и фамилии — 2» конструктором, позволяющим задать имя и фамилию человека при рождении, а также сам год рождения. Класс не должен иметь конструктора по умолчанию.

При получении на вход года, который меньше года рождения:

- методы GetFullName и GetFullNameWithHistory должны отдавать "No person";
- методы ChangeFirstName и ChangeLastName должны игнорировать запрос.

Кроме того, необходимо объявить константными все методы, которые по сути ими являются.

### Пример

#### Код

```
1 int main() {
2     Person person("Polina", "Sergeeva", 1960);
3     for (int year : {1959, 1960}) {
4         cout << person.GetFullNameWithHistory(year) << endl;
5     }
6 }
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



Кроме того, необходимо объявить константными все методы, которые по сути ими являются.

### Пример

#### Код

```
1 int main() {
2     Person person("Polina", "Sergeeva", 1960);
3     for (int year : {1959, 1960}) {
4         cout << person.GetFullNameWithHistory(year) << endl;
5     }
6
7     person.ChangeFirstName(1965, "Appolinaria");
8     person.ChangeLastName(1967, "Ivanova");
9     for (int year : {1965, 1967}) {
10        cout << person.GetFullNameWithHistory(year) << endl;
11    }
12
13    return 0;
14 }
15
```

#### Вывод

```
1 No person
2 Polina Sergeeva
3 Appolinaria (Polina) Sergeeva
4 Appolinaria (Polina) Ivanova (Sergeeva)
5
```



## Тренировочное задание по программированию: Способы инициализации

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 26 июля г., 12:59 +06

**Инструкции**

Моя работа

Обсуждения

Определите тип `Incognizable`, для которого следующий код будет корректен:

```
1 int main() {  
2     Incognizable a;  
3     Incognizable b = {};  
4     Incognizable c = {0};  
5     Incognizable d = {0, 1};  
6     return 0;  
7 }
```

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.

## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

Отметки

Примечания

Форумы обсуждений

Сообщения

Информация о курсе

## Неделя 4

Ожидаемое время: 24 ч 21 м

Ката по вводу/выводу, исключениям и перегрузке операторов

Видео  ГотовоМатериалы для самостоятельного изучения  Осталось 26 минТренировочные упражнения  Осталось 3 ч 57 м

ОБЯЗАТЕЛЬНО ОЦЕНКА НЕОБХОДИМО СДАТЬ

 Задание по программированию Структура LectureTitle 100 % 2 авг. г. 12:59 +06 20 мин Задание по программированию Обратимая функция с умножен... 100 % 2 авг. г. 12:59 +06 20 мин Тест Работа с текстовыми файлами и... 80 % 2 авг. г. 12:59 +06 30 мин Задание по программированию Работа с файлами 50 % 2 авг. г. 12:59 +06 20 мин Задание по программированию Чтение и вывод таблицы 100 % 2 авг. г. 12:59 +06 28 мин Задание по программированию Класс Rational 2 авг. г. 12:59 +06 18 ч 58 м Тест Перегрузка операторов 100 % 2 авг. г. 12:59 +06 30 мин Тест Исключения 100 % 2 авг. г. 12:59 +06 30 мин Задание по программированию Исключения в классе Rational 2 авг. г. 12:59 +06 20 мин Задание по программированию Работа с сервером времени 2 авг. г. 12:59 +06 20 мин



## Задание по программированию: Структура LectureTitle

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Дана структура LectureTitle:

```
1 struct LectureTitle {  
2     string specialization;  
3     string course;  
4     string week;  
5 };  
6
```

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.

Допишите конструктор и структуры Specialization, Course, Week так, чтобы  
объект LectureTitle можно было создать с помощью кода

```
1 LectureTitle title(  
2     Specialization("C++"),  
3     Course("White belt"),  
4     Week("4th")  
5 );  
6
```



Допишите конструктор и структуры Specialization, Course, Week так, чтобы объект LectureTitle можно было создать с помощью кода

```
1  LectureTitle title(  
2      |   Specialization("C++"),  
3      |   Course("White belt"),  
4      |   Week("4th")  
5  );  
6
```

но нельзя было с помощью следующих фрагментов кода:

```
4  
5  LectureTitle title = {"C++", "White belt", "4th"};  
6  
7  LectureTitle title = {{"C++"}, {"White belt"}, {"4th"} };  
8  
9  LectureTitle title(  
10     |   Course("White belt"),  
11     |   Specialization("C++"),  
12     |   Week("4th")  
13  );  
14  
15  LectureTitle title(  
16      |   Specialization("C++"),  
17      |   Week("4th"),  
18      |   Course("White belt")  
19  );  
20
```



## Тренировочное задание по программированию: Обратимая функция

✓ Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Реализуйте рассказанный на лекции класс Function, позволяющий создавать, вычислять и инвертировать функцию, состоящую из следующих элементарных операций:

- прибавить вещественное число  $x$ ;
- вычесть вещественное число  $x$ .

При этом необходимо объявить константными все методы, которые по сути такими являются.

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Замечание

- Более детальное описание задачи с подробным разбором реализации вышеуказанного класса приводится в двух предшествующих видеолекциях.
- На проверку пришлите файл, содержащий реализацию вышеуказанного класса Function.

### Пример



## Пример

## Код

```
1 struct Image {
2     double quality;
3     double freshness;
4     double rating;
5 };
6
7 struct Params {
8     double a;
9     double b;
10    double c;
11 };
12
13 Function MakeWeightFunction(const Params& params,
14                             const Image& image) {
15     Function function;
16     function.AddPart('-', image.freshness * params.a + params.
17     function.AddPart('+', image.rating * params.c);
18     return function;
19 }
20
21 double ComputeImageWeight(const Params& params, const Image&
22                           Function function = MakeWeightFunction(params, image);
23                           return function.Apply(image.quality);
24 }
25
26 double ComputeQualityByWeight(const Params& params,
27                               const Image& image,
28                               double weight) {
29     Function function = MakeWeightFunction(params, image);
30     function.Invert();
31     return function.Apply(weight);
32 }
33
34 int main() {
35     Image image = {10, 2, 6};
36     Params params = {4, 2, 6};
37     cout << ComputeImageWeight(params, image) << endl;
38     cout << ComputeQualityByWeight(params, image, 46) << endl;
39     return 0;
40 }
```



## Пример

## Код

```
1  e {
2    ality;
3    eshness;
4    ting;
5
6
7  ms {
8
9
10
11
12
13 keWeightFunction(const Params& params,
14   const Image& image) {
15   function;
16   AddPart('-', image.freshness * params.a + params.b);
17   AddPart('+', image.rating * params.c);
18   nction;
19
20
21  uteImageWeight(const Params& params, const Image& image) {
22   function = MakeWeightFunction(params, image);
23   nction.Apply(image.quality);
24
25
26  uteQualityByWeight(const Params& params,
27   const Image& image,
28   double weight) {
29   function = MakeWeightFunction(params, image);
30   Invert();
31   nction.Apply(weight);
32
33
34 {
35   ge = {10, 2, 6};
36   rams = {4, 2, 6};
37   omputeImageWeight(params, image) << endl;
38   omputeQualityByWeight(params, image, 46) << endl;
39
40
```

```
14     function;
15     AddPart('-', image.freshness * params.a + params.b);
16     AddPart('+', image.rating * params.c);
17     nction;
18
19
20
21     uteImageWeight(const Params& params, const Image& image) {
22     function = MakeWeightFunction(params, image);
23     nction.Apply(image.quality);
24
25
26     uteQualityByWeight(const Params& params,
27                         const Image& image,
28                         double weight) {
29     function = MakeWeightFunction(params, image);
30     Invert();
31     nction.Apply(weight);
32
33
34     {
35     ge = {10, 2, 6};
36     rams = {4, 2, 6};
37     omputeImageWeight(params, image) << endl;
38     omputeQualityByWeight(params, image, 46) << endl;
39
40
```

## Выход

```
1 36
2 20
```



## Задание по программированию: Обратимая функция с умножением и делением

✓ Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 2 авг. г., 12:59 +06

Инструкции Моя работа

Обсуждения

Добавьте в класс Function из задачи «Обратимая функция» обработку умножения ('\*') и деления ('/'). Гарантируется отсутствие элементарных операций умножения и деления на 0.

### Пример

#### Код

```
1 struct Image {
2     double quality;
3     double freshness;
4     double rating;
5 };
6
7 struct Params {
8     double a;
9     double b;
10    double c;
11 };
12
13 Function MakeWeightFunction(const Params& params,
14                             const Image& image) {
15     Function function;
16     function.AddPart('*', params.a);
17     function.AddPart('/', image.freshness * params.b);
```

#### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



```
12
13     Function MakeWeightFunction(const Params& params,
14                               const Image& image) {
15         Function function;
16         function.AddPart('*', params.a);
17         function.AddPart('-', image.freshness * params.b);
18         function.AddPart('+', image.rating * params.c);
19         return function;
20     }
21
22     double ComputeImageWeight(const Params& params, const Image&
23                               Function function = MakeWeightFunction(params, image);
24                               return function.Apply(image.quality);
25     }
26
27     double ComputeQualityByWeight(const Params& params,
28                                   const Image& image,
29                                   double weight) {
30         Function function = MakeWeightFunction(params, image);
31         function.Invert();
32         return function.Apply(weight);
33     }
34
35     int main() {
36         Image image = {10, 2, 6};
37         Params params = {4, 2, 6};
38         cout << ComputeImageWeight(params, image) << endl;
39         cout << ComputeQualityByWeight(params, image, 52) << endl;
40         return 0;
41 }
```

## ВЫВОД

```
1 72
2 5
```



## Задание по программированию: Работа с файлами

Повторить попытку · Получено баллов: 1/2  
Для успешной сдачи вам необходимо набрать 2/2 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

В этом задании вам предстоит написать две небольшие программы.  
Каждый пункт - отдельная задача, решение отправляйте в поле с  
соответствующим номером.

### Часть 1

Ваша программа должна считать содержимое файла input.txt и напечатать  
его на экран без изменений. Гарантируется, что содержимое файла input.txt  
заканчивается переводом строки.

#### Пример

input.txt

```
1 Keep calm
2 and
3 learn C++
4
```

stdout

```
1 Keep calm
```

### Как отправить

Когда работа будет готова,  
вы можете загрузить файлы  
для каждой части задания на  
вкладке 'Мои работы'.

**stdout**

```
1 Keep calm
2 and
3 learn C++
4
```

**Часть 2**

Снова считайте все содержимое файла `input.txt`, но на этот раз выведите его в файл `output.txt`. Точно так же гарантируется, что содержимое файла `input.txt` заканчивается переводом строки.

**Пример****input.txt**

```
1 Keep calm
2 and
3 learn C++
4
```

**output.txt**

```
1 Keep calm
2 and
3 learn C++
4
```



## Тренировочное задание по программированию: Вывод с точностью

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

В файле input.txt записаны вещественные числа, по одному на строчку. На стандартный вывод напечатайте эти числа в том же порядке, по одному на строке, но сделайте так, чтобы у каждого из них было ровно три знака после десятичной точки.

### Пример ввода 1

```
1    5
2    0.34567
3    10.4
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Пример вывода 1

```
1    5.000
2    0.346
3    10.400
```



3 10.4

## Пример вывода 1

1 5.000  
2 0.346  
3 10.400

## Пример ввода 2

1 3.33353  
2 3.333  
3 3.3

## Пример вывода 2

1 3.334  
2 3.333  
3 3.300

## Примечание:

Воспользуйтесь стандартными манипуляторами потоков вывода.



## Задание по программированию: Чтение и вывод таблицы

✓ Зачет · 1/1 баллов

Срок сдачи Сдайте это задание до 2 авг. г., 12:59 +06

Инструкции Моя работа

Обсуждения

В первой строке файла input.txt записаны два числа N и M. Далее в файле находится таблица из N строк и M столбцов, представленная в формате CSV (comma-separated values). Такой формат часто используется для текстового представления таблиц с данными: в файле несколько строк, значения из разных ячеек внутри строки отделены друг от друга запятыми. Ваша задача — вывести данные на экран в виде таблицы. Формат таблицы:

- 1) размер ячейки равен 10,
- 2) соседние ячейки отделены друг от друга пробелом,
- 3) последняя ячейка в строке НЕ оканчивается пробелом,
- 4) последняя строка НЕ оканчивается символом перевода строки.

Гарантируется, что в таблице будет ровно N строк и M столбцов, значение каждой из ячеек — целое число. Пример ввода

```
1 2 3
2 1,2,3
3 4,5,6
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

### Пример вывода

1		1	2	3	
2		4	5	6	



## Тренировочное задание по программированию: Список студентов

Повторить попытку · Получено баллов: 0/1

Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Определите структуру «Студент» со следующими полями: имя, фамилия, день, месяц и год рождения. Создайте вектор из таких структур, заполните его из входных данных и затем по запросам выведите нужные поля. Чтение и запись данных в этой задаче производится с использованием стандартных потоков.

### Формат ввода

Первая строка содержит одно целое число  $N$  от 0 до 10000 — число студентов.

Далее идут  $N$  строк, каждая из которых содержит две строки длиной от 1 до 15 символов — имя и фамилию очередного студента, и три целых числа от 0 до 1000000000 — день, месяц и год рождения.

Следующая строка содержит одно целое число  $M$  от 0 до 10000 — число запросов.

Следующие  $M$  строк содержат строку длиной от 1 до 15 символов — запрос, и целое число от 1 до 1000000000 — номер студента (нумерация начинается с 1).

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



### Формат вывода

Для запроса вида **name *K***, где *K* от 1 до *N*, выведите через пробел имя и фамилию *K*-го студента.

Для запроса вида **date *K***, где *K* от 1 до *N*, выведите через точку день, месяц и год рождения *K*-го студента.

Для остальных запросов выведите **bad request**.

**NB:** неверные запросы могут выходить за границы, указанные для данных.

### Пример ввода

```
1 3
2 Ivan Ivanov 1 1 1901
3 Petr Petrov 2 2 1902
4 Alexander Sidorov 3 3 1903
5 3
6 name 1
7 date 3
8 mark 5
```

### Пример вывода

```
1 Ivan Ivanov
2 3.3.1903
3 bad request
```



## Задание по программированию: Класс Rational

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 5/5 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

### Общее описание

В этой задаче вам надо разработать класс для представления рациональных чисел и внедрить его в систему типов языка C++ так, чтобы им можно было пользоваться естественным образом. Задание состоит из нескольких частей. К каждой части приложен файл с заготовкой программы, который имеет следующую структуру

```
1 #include <iostream>
2 using namespace std;
3
4 // Комментарии, которые говорят, что именно нужно реализовать
5
6 int main() {
7     // Набор юнит-тестов для вашей реализации
8     cout << "OK" << endl;
9     return 0;
10 }
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

Вам нужно, **не меняя тело функции main**, реализовать то, что предложено в комментариях, так, чтобы ваша программа выводила на экран сообщение OK. Функция main содержит различные тесты, которые проверяют



Вам нужно, **не меняя тело функции main**, реализовать то, что предложено в комментариях, так, чтобы ваша программа выводила на экран сообщение OK. Функция main содержит ряд юнит-тестов, которые проверяют правильность вашей реализации. Если эти тесты находят ошибку в вашей реализации, то вместо OK программа выведет подсказку о том, что именно работает неправильно.

Когда ваша программа начнёт выводить OK, можете отправлять свой файл на проверку. Ваша реализация будет тестируться на другом, более полном наборе тестов. **То есть, если ваша программа выводит OK, то это не даёт гарантии, что ваша реализация будет принята тестирующей системой.** В случае неудачной посылки тестирующая система также выдаст вам подсказку о том, какая именно часть реализации работает неправильно.

Внимательно изучите набор юнит-тестов для **каждой** заготовки решения. Тесты описывают требования, предъявляемые к вашей реализации, которые могут быть не указаны в условии задач

На проверку можно отправлять весь файл с вашей реализацией: функцию main удалять из него не нужно.

## Часть 1

В первой части Вам надо реализовать класс Rational, который представляет собой рациональное число вида p/q, где p — целое, а q — натуральное и диапазоны возможных значений p, q таковы, что могут быть представлены типом int. При этом, класс Rational должен иметь следующий интерфейс:

```
1 class Rational {
2 public:
3     Rational();
4     Rational(int numerator, int denominator);
5 }
```



## Часть 1

В первой части Вам надо реализовать класс Rational, который представляет собой рациональное число вида  $p/q$ , где  $p$  — целое, а  $q$  — натуральное и диапазоны возможных значений  $p$ ,  $q$  таковы, что могут быть представлены типом int. При этом, класс Rational должен иметь следующий интерфейс:

```
1 class Rational {
2 public:
3     Rational();
4     Rational(int numerator, int denominator);
5
6     int Numerator() const;
7     int Denominator() const;
8 };
```

Класс Rational должен гарантировать, что  $p/q$  — это несократимая дробь. Например, код

```
1 Rational r(4, 6);
2 cout << r.Numerator() << '/' << r.Denominator();
```

должен выводить «2/3» — именно поэтому мы делаем Rational классом, а не структурой. Структура позволила бы нарушить этот инвариант:

```
1 struct Rational {
2     int numerator, denominator;
3 };
4
5 Rational r{2, 3};
```



должен выводить «2/3» — именно поэтому мы делаем Rational классом, а не структурой. Структура позволила бы нарушить этот инвариант:

```
1 struct Rational {
2     int numerator, denominator;
3 };
4
5 Rational r{2, 3};
6 r.numerator = 6; // Нарушен инвариант несократимости
```

Список требований, предъявляемых к реализации интерфейса класса Rational:

1. Конструктор по умолчанию должен создавать дробь с числителем 0 и знаменателем 1.
2. При конструировании объекта класса Rational с параметрами  $p$  и  $q$  должно выполняться сокращение дроби  $p/q$  (здесь вам может пригодиться решение задачи «Наибольший общий делитель»).
3. Если дробь  $p/q$  отрицательная, то объект  $Rational(p, q)$  должен иметь отрицательный числитель и положительный знаменатель.
4. Если дробь  $p/q$  положительная, то объект  $Rational(p, q)$  должен иметь положительные числитель и знаменатель (обратите внимание на случай  $Rational(-2, -3)$ ).
5. Если числитель дроби равен нулю, то знаменатель должен быть равен 1.

Гарантируется, что ни в одном из тестов, на которых будет тестироваться ваша реализация, не будет знаменателя, равного нулю.



Гарантируется, что ни в одном из тестов, на которых будет  
тестироваться ваша реализация, не будет знаменателя, равного нулю.

Замечание:

В стандарте языка C++17 для вычисления НОД имеется стандартная  
функция gcd. Допустимо использовать данную функцию для сокращения  
дробей. Подробнее с данной функцией и её ограничениями можно  
ознакомиться по ссылке: <https://en.cppreference.com/w/cpp/numeric/gcd>

Шаблон программы:

rational\_interface.cpp

## Часть 2

Реализуйте для класса Rational операторы ==, + и - так, чтобы операции с  
дробями можно было записывать естественным образом. Например,  
следующий код должен быть валидным:

```
1 const Rational r = Rational(1, 2) + Rational(1, 3) - Rational(1, 4);
2 if (r == Rational(7, 12)) {
3     cout << "equal";
4 }
```

Гарантируется, что во всех тестах, на которых будет тестироваться  
ваша реализация, числители и знаменатели ВСЕХ дробей (как  
исходных, так и получившихся в результате соответствующих  
арифметических операций) будут укладываться в диапазон типа int.



Гарантируется, что во всех тестах, на которых будет тестируться ваша реализация, числители и знаменатели ВСЕХ дробей (как исходных, так и получившихся в результате соответствующих арифметических операций) будут укладываться в диапазон типа int.

Шаблон программы:

```
rational_eq_add_sub.cpp
```

### Часть 3

Аналогично предыдущей части, реализуйте операторы \* и /. Например, следующий код должен быть валидным:

```
1 const Rational r = Rational(1, 2) * Rational(1, 3) / Rational(2, 3);
2 if (r == Rational(1, 6)) {
3     cout << "equal";
4 }
```

Как и в предыдущем случае, гарантируется, что во всех тестах, на которых будет тестируться ваша реализация, числители и знаменатели ВСЕХ дробей (как исходных, так и получившихся в результате соответствующих арифметических операций) будут укладываться в диапазон типа int.

Кроме того, гарантируется, что ни в одном из тестов, на которых будет тестируться ваша реализация, не будет выполняться деление на ноль.



Шаблон программы:

rational\_mult\_div.cpp

## Часть 4

В этой части вам нужно реализовать операторы ввода и вывода для класса Rational. В результате у вас должен работать, например, такой код

```
1 Rational r;
2 cin >> r;
3 cout << r;
```

Шаблон программы:

rational\_io.cpp

## Часть 5

Наконец, вам нужно реализовать возможность использовать объекты класса Rational в качестве элементов контейнера set и ключей в контейнере map. Пример:

```
1 set<Rational> rationals;
2 rationals.insert(Rational(1, 2));
```



rational\_io.cpp

## Часть 5

Наконец вам нужно реализовать возможность использовать объекты класса Rational в качестве элементов контейнера set и ключей в контейнере map. Пример:

```
1  set<Rational> rationals;
2  rationals.insert(Rational(1, 2));
3  rationals.insert(Rational(1, 3));
4
5  map<Rational, string> name;
6  name[Rational(1, 2)] = "одна вторая";
```

Шаблон программы:

rational\_map.cpp



## Тренировочное задание по программированию: Функция EnsureEqual

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

### Инструкции

Моя работа

Обсуждения

Напишите функцию

```
1 void EnsureEqual(const string& left, const string& right);
```

В случае, если строка **left** не равна строке **right**, функция должна выбрасывать исключение `runtime_error` с содержанием "`<l> != <r>`", где `<l>` и `<r>` - строки, которые хранятся в переменных **left** и **right** соответственно. Обратите внимание, что вокруг знака неравенства в строке, которая помещается в исключение, должно быть ровно по одному пробелу.

Если **left == right**, функция должна корректно завершаться.

Например, код

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.

[Основы разработки на C++: белый пояс](#) > [Неделя 4](#) > Функция EnsureEqual

&lt; Предыдущий Дальше &gt;

В случае, если строка **left** не равна строке **right**, функция должна выбрасывать исключение `runtime_error` с содержанием "`<l> != <r>`", где `<l>` и `<r>` - строки, которые хранятся в переменных **left** и **right** соответственно. Обратите внимание, что вокруг знака неравенства в строке, которая помещается в исключение, должно быть ровно по одному пробелу.

Если **left == right**, функция должна корректно завершаться.

Например, код

```
1 int main() {
2     try {
3         EnsureEqual("C++ White", "C++ White");
4         EnsureEqual("C++ White", "C++ Yellow");
5     } catch (runtime_error& e) {
6         cout << e.what() << endl;
7     }
8     return 0;
9 }
```

должен выводить

```
1 C++ White != C++ Yellow
```

вкладке 'Мои работы'.



## Задание по программированию: Исключения в классе Rational

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Вспомним класс Rational, который мы реализовали в задачах, посвящённых перегрузке операторов. Там специально говорилось, что в тестах объекты класса Rational никогда не создаются с нулевым знаменателем и никогда не выполняется деление на ноль. Настало время избавиться от этого ограничения и научиться обрабатывать нулевой знаменатель и деление на ноль. В этой задаче вам предстоит это сделать.

Поменяйте реализацию конструктора класса Rational так, чтобы он выбрасывал исключение [invalid\\_argument](#), если знаменатель равен нулю. Кроме того, переделайте реализацию оператора деления для класса Rational так, чтобы он выбрасывал исключение [domain\\_error](#), если делитель равен нулю.

Напомним, что вам надо реализовать интерфейс

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



нули.

Напомним, что вам надо реализовать интерфейс

```
1 class Rational {
2 public:
3     Rational();
4     Rational(int numerator, int denominator);
5
6     int Numerator() const;
7     int Denominator() const;
8 }
```

и оператор деления для класса Rational

```
1 Rational operator / (const Rational& lhs, const Rational& rh
```

Так же, как и в задачах на перегрузку операторов, вам будет предложен файл с заготовкой кода. Функция main в этом файле содержит ряд юнит-тестов, которые позволят вам проверить свою реализацию. При проверке вашего решения оно будет тестируться на более полном наборе тестов, так что успешное прохождение всех юнит-тестов не гарантирует, что ваша реализация будет зачтена.

Шаблон программы:

```
rational_throw.cpp
```



## Тренировочное задание по программированию: Калькулятор обыкновенных дробей

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Итак, теперь у нас с вами есть полноценный тип для представления обыкновенных дробей: класс Rational интегрирован в язык с помощью перегрузки операторов и не позволяет выполнять некорректные операции за счёт выбрасывания исключений. Давайте на основе класса Rational создадим простой калькулятор для обыкновенных дробей.

Вам надо написать программу, которая считывает из стандартного ввода одну строку в формате **ДРОБЬ\_1 операция ДРОБЬ\_2**. ДРОБЬ\_1 и ДРОБЬ\_2 имеют формат X/Y, где X — целое, а Y — целое неотрицательное число. операция — это один из символов '+', '-', '\*', '/'.

Если ДРОБЬ\_1 или ДРОБЬ\_2 является некорректной обыкновенной дробью, ваша программа должна вывести в стандартный вывод сообщение "Invalid argument". Если считанная операция — это деление на ноль, выведите в стандартный вывод сообщение "Division by zero". В противном случае выведите результат операции.

Пример

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



обыкновенных дробей: класс Rational интегрирован в язык с помощью перегрузки операторов и не позволяет выполнять некорректные операции за счёт выбрасывания исключений. Давайте на основе класса Rational создадим простой калькулятор для обыкновенных дробей.

Вам надо написать программу, которая считывает из стандартного ввода одну строку в формате **ДРОБЬ\_1 операция ДРОБЬ\_2**. ДРОБЬ\_1 и ДРОБЬ\_2 имеют формат X/Y, где X — целое, а Y — целое неотрицательное число. операция — это один из символов '+', '-', '\*', '/'.

Если ДРОБЬ\_1 или ДРОБЬ\_2 является некорректной обыкновенной дробью, ваша программа должна вывести в стандартный вывод сообщение "Invalid argument". Если считанная операция — это деление на ноль, выведите в стандартный вывод сообщение "Division by zero". В противном случае выведите результат операции.

### Пример

stdin	stdout
1/2 + 1/3	5/6
1/2 + 5/0	Invalid argument
4/5 / 0/8	Division by zero

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



## Задание по программированию: Работа с сервером времени

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

**Срок сдачи** Сдайте это задание до 2 авг. г., 12:59 +06

[Инструкции](#) [Моя работа](#)

Обсуждения

Вам дана функция string AskTimeServer(), про которую известно следующее:

- в процессе своей работы она обращается по сети к удалённому серверу, запрашивая текущее время;
- если обращение к серверу прошло успешно, функция возвращает текущее время в виде строки;
- если в процессе обращения к серверу возникла сетевая проблема, функция выбрасывает исключение system\_error;
- функция может выбрасывать другие исключения, чтобы сообщить о других проблемах.

Используя функцию AskTimeServer, напишите класс TimeServer со следующим интерфейсом:

```
1  class TimeServer {  
2  public:  
3      string GetcurrentTime();
```

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



Используя функцию AskTimeServer, напишите класс TimeServer со следующим интерфейсом:

```
1  class TimeServer {
2  public:
3  |  string GetcurrentTime();
4  private:
5  |  string last_fetched_time = "00:00:00";
6  };
```

Метод GetcurrentTime должен вести себя так:

- он должен вызывать функцию AskTimeServer, записать её результат в поле last\_fetched\_time и вернуть значение этого поля;
- если AskTimeServer выбросила исключение system\_error, метод GetcurrentTime должен его поймать и вернуть текущее значение поля last\_fetched\_time. Таким образом мы скрываем от пользователя сетевые проблемы, возвращая значение, которое было получено при последнем успешном обращении к серверу;
- если AskTimeServer выбросила другое исключение, метод GetcurrentTime должен пробросить его дальше, потому что в рамках класса TimeServer мы не знаем, как обрабатывать проблемы, не связанные со сбоями сети.

### Как выполнять задание

Вам дан файл, содержащий заготовку класса TimeServer. В нём вам надо реализовать метод GetcurrentTime так, как описано выше. Файл содержит пустую функцию AskTimeServer. Для тестирования своей реализации вы можете пробовать добавлять разные команды в её тело:



сетевые проблемы, возвращая значение, которое было получено при последнем успешном обращении к серверу;

- если AskTimeServer выбросила другое исключение, метод GetcurrentTime должен пробросить его дальше, потому что в рамках класса TimeServer мы не знаем, как обрабатывать проблемы, не связанные со сбоями сети.

### Как выполнять задание

Вам дан файл, содержащий заготовку класса TimeServer. В нём вам надо реализовать метод GetcurrentTime так, как описано выше. Файл содержит пустую функцию AskTimeServer. Для тестирования своей реализации вы можете пробовать добавлять разные команды в её тело:

- возврат строки;
- выброс исключения system\_error (класс system\_error принимает в конструкторе параметр типа error\_code, поэтому самый простой способ выбросить это исключение — throw system\_error(error\_code());, подробнее см. [https://en.cppreference.com/w/cpp/error/system\\_error](https://en.cppreference.com/w/cpp/error/system_error) );
- выброс других исключений.

time\_server.cpp



## Обзор

Неделя 1

Неделя 2

Неделя 3

Неделя 4

Неделя 5

## Отметки

Примечания

Форумы обсуждений

Сообщения

Информация о курсе

30 мин

## Задание по программированию

Исключения в классе Rational

20 мин

2 авг. г.

12:59 +06

## Задание по программированию

Работа с сервером времени

20 мин

2 авг. г.

12:59 +06

## Неделя 5

## Первое кумитэ

Вideo ГотовоМатериалы для  
самостоятельного изучения ГотовоДругое Готово

ОБЯЗАТЕЛЬНО

ОЦЕНКА

НЕОБХОДИМО  
СДАТЬ

## Задание по программированию

Финальная задача курса

44 ч

100 %

9 авг. г.

12:59 +06

Получать электронные письма от **Московский физико-технический институт?**Да  



## Задание по программированию: Финальная задача курса

Зачет · 1/1 баллов

**Срок сдачи** Сдайте это задание до 9 авг. г., 12:59 +06

**Инструкции** Моя работа

Обсуждения

Необходимо написать программу на C++, которая реализует работу с простой базой данных (сокращённо «БД»). Программа будет общаться с пользователем через стандартный ввод и вывод (потоки `stdin` и `stdout`).

Будем хранить в нашей БД пары вида: дата, событие. Определим дату как строку вида *Год-Месяц-День*, где *Год*, *Месяц* и *День* — целые числа.

Событие определим как строку из произвольных печатных символов без разделителей внутри (пробелов, табуляций и пр.). Событие не может быть пустой строкой. В одну и ту же дату может произойти много разных событий, БД должна суметь их все сохранить. Одинаковые события, произошедшие в один и тот же день, сохранять не нужно: достаточно сохранить только одно из них.

Наша БД должна понимать определённые команды, чтобы с ней можно было взаимодействовать:

### Как отправить

Когда работа будет готова, вы можете загрузить файлы для каждой части задания на вкладке 'Мои работы'.



Наша БД должна понимать определённые команды, чтобы с ней можно было взаимодействовать:

1	- добавление события:	Add Дата Событии
2	- удаление события:	Del Дата Событии
3	- удаление всех событий за конкретную дату:	Del Дата
4	- поиск событий за конкретную дату:	Find Дата
5	- печать всех событий за все даты:	Print
6		

Все команды, даты и события при вводе разделены пробелами. Командычитываются из стандартного ввода. В одной строке может быть ровно одна команда, но можно ввести несколько команд в несколько строк. На вход также могут поступать пустые строки — их следует игнорировать и продолжать обработку новых команд в последующих строках.

### Добавление события (Add Дата Событие)

При добавлении события БД должна его запомнить и затем показывать его при поиске (командой Find) или печати (командой Print). Если указанное событие для данной даты уже существует, повторное его добавление нужно игнорировать. В случае корректного ввода (см. раздел «Обработка ошибок ввода») программа ничего не должна выводить на экран.



## Удаление события (Del *Дата Событие*)

Команда должна удалить добавленное ранее событие с указанным именем в указанную дату, если оно существует. Если событие найдено и удалено, программа должна вывести строку «Deleted successfully» (без кавычек). Если событие в указанную дату не найдено, программа должна вывести строку «Event not found» (без кавычек).

## Удаление нескольких событий (Del *Дата*)

Команда удаляет все ранее добавленные события за указанную дату. Программа всегда должна выводить строку вида «Deleted *N* events», где *N* — это количество всех найденных и удалённых событий. *N* может быть равно нулю, если в указанную дату не было ни одного события.

## Поиск событий (Find *Дата*)

Найти и распечатать ранее добавленные события в указанную дату. Программа должна вывести на печать только сами события, по одному на строке. События должны быть отсортированы по возрастанию в порядке сравнения строк между собой (тип string).

## Печать всех событий (Print)

С помощью этой команды можно показать полное содержимое нашей БД. Программа должна вывести на печать все пары *Дата Событие* по одной на строке. Все пары должны быть отсортированы по дате, а события в рамках одной даты должны быть отсортированы по возрастанию в порядке сравнения строк между собой (тип string). Даты должны быть отформатированы специальным образом: ГГГГ-ММ-ДД, где Г, М, Д — это цифры чисел года, месяца и дня соответственно. Если какое-то число имеет



## Печать всех событий (Print)

С помощью этой команды можно показать полное содержимое нашей БД. Программа должна вывести на печать все пары *Дата Событие* по одной на строке. Все пары должны быть отсортированы по дате, а события в рамках одной даты должны быть отсортированы по возрастанию в порядке сравнения строк между собой (тип *string*). Даты должны быть отформатированы специальным образом: *ГГГГ-ММ-ДД*, где *Г*, *М*, *Д* — это цифры чисел года, месяца и дня соответственно. Если какое-то число имеет меньше разрядов, то оно должно дополняться нулями, например, 0001-01-01 — первое января первого года. Вам не понадобится выводить дату с отрицательным значением года.

### Обработка ошибок ввода

Ввод в этой задаче не всегда корректен: некоторые ошибки ввода ваша программа должна корректно обрабатывать.

#### Какие ошибки ввода **нужно** обрабатывать

Если пользователь ввёл неизвестную команду, то программа должна об этом сообщить, выведя строку «Unknown command: *COMMAND*», где *COMMAND* — это та команда, которую ввёл пользователь. Командой считается первое слово в строке (до пробела).

Если дата не соответствует формату *Год-Месяц-День*, где *Год*, *Месяц* и *День* — произвольные целые числа, то программа должна напечатать «Wrong date format: *DATE*», где *DATE* — это то, что пользователь ввёл вместо даты (до пробела). Обратите внимание, что части даты разделяются ровно одним дефисом, а сама дата не должна содержать лишних символов ни перед годом, ни после дня. Части даты не могут быть пустыми, но могут быть нулевыми и даже отрицательными.



Если формат даты верен, необходимо проверить валидность месяца и дня.

- Если номер месяца не является числом от 1 до 12, выведите «Month value is invalid: *MONTH*», где *MONTH* — это неверный номер месяца, например, 13 или -1.
- Если месяц корректен, а день не лежит в диапазоне от 1 до 31, выведите «Day value is invalid: *DAY*», где *DAY* — это неверный номер дня в месяце, например, 39 или 0.

Обратите внимание, что:

- Значение года проверять отдельно не нужно.
- Не нужно проверять календарную корректность даты: количество дней в каждом месяце считается равным 31, високосные годы учитывать не нужно.
- Если неверны как месяц, так и день, то нужно вывести одно сообщение об ошибке в месяце.

Примеры:

- 1-1-1 — корректная дата;
- -1-1-1 — корректная дата (год -1, месяц 1, день 1);
- 1--1-1 — дата в верном формате, но с некорректным месяцем -1;
- 1---1-1 — дата в неверном формате: месяц не может начинаться с двух дефисов.
- 1-+1-+1 — корректная дата, так как +1 это целое число

После обработки любой из описанных ошибок ввода и печати сообщения программа должна завершать своё выполнение.



### Какие ошибки ввода **не нужно** обрабатывать

Мы не ставим своей целью сломать вашу программу всеми возможными способами, поэтому, за исключением описанного в предыдущем пункте, вы можете полагаться на корректность ввода. В частности, мы гарантируем, что:

- Каждая команда занимает целиком ровно одну строку, хотя во вводе могут быть и пустые строки (их нужно игнорировать).
- События всегда содержат указанное количество аргументов: после команды Add всегда следуют дата и событие, после команды Find всегда следует дата, после команды Del всегда следует дата и, возможно, событие, а команда Print не содержит дополнительной информации.
- Все команды, даты и события являются непустыми строками и не содержат пробелов и прочих пробельных символов. (В частности, наши тесты не содержат команды «Add 2018-02-12», потому что в ней после даты отсутствует название события.) С другой стороны, команда Del может не содержать события после даты: в этом случае нужно удалить все события за указанную дату (см. раздел «Удаление нескольких событий»).
- Несмотря на то, что дата с отрицательным значением года считается корректной, тесты устроены так, что её не понадобится выводить в команде Print.



## Примеры

Корректный ввод

```
1 Add 0-1-2 event1
2 Add 1-2-3 event2
3 Find 0-1-2
4
5 Del 0-1-2
6 Print
7 Del 1-2-3 event2
8 Del 1-2-3 event2
```

Вывод:

```
1 event1
2 Deleted 1 events
3 0001-02-03 event2
4 Deleted successfully
5 Event not found
```

Неверный формат даты

```
1 Add 0-13-32 event1
```

Вывод:



## Неверный формат даты

```
1 Add 0-13-32 event1
```

Вывод:

```
1 Month value is invalid: 13
```

## Шаблон решения

Программа должна быть интерактивной и ожидать ввод до тех пор, пока в поток не попадёт символ конца ввода (эта часть уже реализована в шаблоне решения, которое приложено к этому условию).

Допишите в этот шаблон недостающие части решения и отправьте его на проверку:

```
project_template.cpp
```

## Примечания

## Преобразование числа к строке

Чтобы, имея число *MONTH*, составить строку «Month value is invalid: *MONTH*», можно использовать функцию `to_string`, преобразующую число к строке. Таким образом, составить необходимую строку можно следующим



## Примечания

### Преобразование числа к строке

Чтобы, имея число *MONTH*, составить строку «Month value is invalid: *MONTH*», можно использовать функцию `to_string`, преобразующую число к строке. Таким образом, составить необходимую строку можно следующим образом:

```
1 string error = "Month value is invalid: " + to_string(month)
```

### Поиск в константном словаре

При реализации данного шаблона вам может понадобится использовать поиск с помощью квадратных скобок для словаря, переданного в функцию по константной ссылке. Как было показано ранее, это сделать не удастся, так как обращение к несуществующему ключу с помощью квадратных скобок добавит его в словарь, что недопустимо для константного словаря.

В этом случае вместо квадратных скобок используйте метод `at`: в случае отсутствия ключа он выбросит исключение и потому может быть использован для константного объекта.

Например, вместо кода

```
1 void DoSomething(const map<int, int>& m) {
2     // ...
3     if (m.count(key) > 0) {
4         value = m[key]; // не компилируется
```



### Поиск в константном словаре

При реализации данного шаблона вам может понадобится использовать поиск с помощью квадратных скобок для словаря, переданного в функцию по константной ссылке. Как было показано ранее, это сделать не удастся, так как обращение к несуществующему ключу с помощью квадратных скобок добавит его в словарь, что недопустимо для константного словаря.

В этом случае вместо квадратных скобок используйте метод `at`: в случае отсутствия ключа он выбросит исключение и потому может быть использован для константного объекта.

Например, вместо кода

```
1 void DoSomething(const map<int, int>& m) {  
2     // ...  
3     if (m.count(key) > 0) {  
4         value = m[key]; // не компилируется  
5     }  
6     // ...  
7 }
```

используйте код

```
1 void DoSomething(const map<int, int>& m) {  
2     // ...  
3     if (m.count(key) > 0) {  
4         value = m.at(key); // теперь всё хорошо  
5     }  
6     // ...  
7 }
```



```
6 // ...
7 }
```

используйте код

```
1 void DoSomething(const map<int, int>& m) {
2     // ...
3     if (m.count(key) > 0) {
4         value = m.at(key); // теперь всё хорошо
5     }
6     // ...
7 }
```

### Непойманные исключения

Так как для решения задачи требуется программа, работающая верно на большом количестве разных входных данных, то неизбежно в ней могут обнаружиться ошибки, про некоторые из которых мы не рассказали в нашем курсе лекций. Одной из таких ошибок может быть непойманное исключение: ошибка заключается в том, что исключение, будучи выброшенным, не попадает ни под одно из выражений блоков catch вплоть до функции main. В этом случае программа тут же завершится аварийно, и в качестве ошибки в тесте вы увидите «Unknown signal 6».