



Урок 24. C++: частное, остаток и цифры числа

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 24 из 34 · 45 минут · Модуль «Числа: деление нацело»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Таймер хранит 583 секунды. Если написать 583/60, секунды исчезают. Как сохранить и минуты, и остаток?

Результат урока: Разложить целое число с помощью / и %, восстановить исходное значение и проверить нулевой случай.

01. Проблемная ситуация

За 5 минут разложите 583 секунды на полные минуты и оставшиеся секунды. Запишите равенство проверки. Ученик получил 9 минут 58 секунд: найдите, какое равенство нарушено. Затем предположите ответ для 59 и 60 секунд.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Целое деление и остаток дополняют друг друга

Для неотрицательного n и положительного b целочисленное деление даёт $q=n/b$ и $r=n\%b$. Выполняются $n=q*b+r$ и $0\leq r<b$. Делитель ноль недопустим, поэтому его проверяют до вычисления. Операторы имеют одинаковый приоритет и вычисляются слева направо; скобки выделяют нужный этап.

Последняя десятичная цифра равна $n\%10$; $n/10$ удаляет её у неотрицательного числа. Для сотен $n/100$, для десятков $(n/10)\%10$. В нашем контракте 0...999 выводятся именно три отдельных разряда, поэтому 7 даёт 0 0 7. Это не восстановление исходной записи 007: ведущие нули целочисленный ввод не хранит.

В C++ отрицательное целочисленное частное усекается к нулю; правила Python для отрицательных аргументов иные. Здесь отрицательные входы отвергаются, поэтому доказательство остатка относится только к указанному контракту. Все программы работают с локальными целыми учебными данными.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

$583/60=9$, $583\%60=43$; проверка $9\cdot60+43=583$. Для 60 ответ 1 0, для 0 — 0 0. Программа принимает одно целое 0...9999, иначе печатает error.

```
#include <iostream>
```

```
int main() {
    long long s;
    if (!(std::cin >> s) || s < 0 || s > 9999) {
        std::cout << "error\n"; return 0;
    }
    std::cout << s / 60 << " " << s % 60 << "\n";
}
```



C++: частное, остаток и цифры числа

Делимое	583 секунды
Частное	9 полных минут
Остаток	43 секунды; меньше 60
Проверка	$9 \cdot 60 + 43 = 583$

$583 = 9 \cdot 60 + 43$; для 407 разряды: 4 сотни, 0 десятков, 7 единиц.

03. Практика 1. Проверяем материал

Три разряда без потери нулей

14–25 минут. Напишите программу трёх разрядов для целого n от 0 до 999; вне диапазона error.

n	Сотни, десятки, единицы
0	0 0 0
7	0 0 7
100	1 0 0
407	4 0 7
999	9 9 9

2 минуты — прогноз пяти строк; 4 — код; 3 — запуск всех строк, а также -1 и 1000; 2 — обратная проверка $100 \cdot a + 10 \cdot b + c$. Продукт: исходник и журнал «ввод, прогноз, вывод». Критерии: три числа, сохранён нулевой разряд, границы проверены, каждое исходное n восстановлено.

04. Практика 2. Перенос способа

Полные коробки, остаток и число упаковок

25–33 минуты. Нужно упаковать n предметов в коробки вместимостью b . Ввод: целые $0 \leq n \leq 1000000$ и $1 \leq b \leq 1000$. Выведите число полных коробок q , остаток r и общее число использованных коробок; неполную считают одной. За 2 минуты составьте формулу, за 4 реализуйте, за 2 проверьте 25 6, 24 6, 0 6 и 5 0. Критерии: делитель проверен заранее, лишняя коробка при точном делении не добавлена.

05. Обсуждение и границы вывода

33–40 минут. Достаточно ли проверить разбор только на 999? Сравните тесты 407 и 478, предложите пример, где промежуточный ноль обязателен. Обоснуйте, какие две проверки нужно сделать с остатком.

06. Индивидуальная проверка

40–45 минут. Разложите 905 на разряды. Для 125 предметов и вместимости 12 найдите q, r и число коробок. Назовите условие на делитель. Четыре критерия: разряды, пара q/r , коробки, условие.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ожидаемые ответы даны в таблице; -1 и 1000 дают error. Ошибка $n\%100$ вместо десятков даёт для 407 число 7 , но для 478 число 78 : один удачный пример её скрывает. Полный вариант ниже.

```
#include <iostream>
```

```
int main() {
    int n;
    if (!(std::cin >> n) || n < 0 || n > 999) {
        std::cout << "error\n"; return 0;
    }
    std::cout << n / 100 << " " << (n / 10) % 10
        << " " << n % 10 << "\n";
}
```

Практика 2: разбор и критерии

Ответы: $25\ 6 \rightarrow 4\ 1\ 5$; $24\ 6 \rightarrow 4\ 0\ 4$; $0\ 6 \rightarrow 0\ 0\ 0$; $5\ 0 \rightarrow \text{error}$. Сравнение $r!=0$ имеет значение true/false, при сложении превращается в $1/0$. Формула $q+1$ без условия неверна для 24 и 0 .

```
#include <iostream>
```

```
int main() {
    long long n, b;
    if (!(std::cin >> n >> b) || n < 0 || n > 1000000
        || b < 1 || b > 1000) {
        std::cout << "error\n"; return 0;
    }
    long long q = n / b, r = n % b;
    long long boxes = q + (r != 0);
    std::cout << q << " " << r << " " << boxes << "\n";
}
```

Обсуждение: возможный вывод

Нужны ноль, границы и разные позиции нулей: 7 , 100 , 407 . Проверяют восстановление $n=q*b+r$ и диапазон остатка. Только первое равенство недостаточно: $583=8*60+103$, но 103 не может быть остатком при делении на 60 .

Индивидуальная проверка: ответ и критерии

$9\ 0\ 5$; $q=10$, $r=5$; коробок 11 ; $b>0$. Четыре пункта по одному баллу. Проверка $10*12+5=125$.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Локальный компилятор G++ с C++17 или настроенная учителем IDE; сборка `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. В Windows задайте `lesson.exe`. Запуск `./lesson` в Linux/macOS, `.\lesson.exe` в Windows. Для урока 31 дополнительно обычный браузер: открывать локальный SVG, не C++-исходник. Все входы и ожидаемые результаты даны. Установку и проверку запуска учитель выполняет до занятия; детские аккаунты и интернет не нужны.

Уровень — начинающий программист, 7 класс. Предпосылки: целые типы, ввод и арифметические выражения. В паре автор вводит и объясняет, проверяющий сопоставляет с прогнозом, затем роли меняются. Без компьютера трассируйте полный код на бумаге, отмечайте координаты по схеме и проверяйте готовые результаты; учитель демонстрирует заранее собранный пример. Бумажная трассировка не выдаётся за реальную компиляцию или запуск ученика. План 45 минут включает чтение, ввод, сборку и обсуждение. В графических задачах сначала прогноз, затем сравнение. Матрица программы предназначена учителю.



Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Деление нацело. Остаток от деления	Разложить целое число с помощью / и %, восстановить исходное значение и проверить нулевой случай.	Самостоятельный продукт и тесты, описанные в практиках
Составлять и отлаживать простые диалоговые программы	Сверяет прогноз с фактическим запуском; объясняет ошибочный вариант	Журнал тестов и индивидуальный ответ по критериям

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

C++: частное и остаток

<https://eel.is/c++draft/expr.mul>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 17.10.2026.