

Числовые типы данных

Целочисленные типы

До сих пор рассматривались только один тип целочисленных переменных — `int`. На самом деле существует несколько основных целочисленных типов, тип `int` — лишь один (но наиболее часто используемый) из них.

Таблица основных целочисленных типов.

Название	Размер	Знаковый	Синонимы
<code>short</code>	2 байта	Знаковый	<code>short int</code> , <code>signed short</code> , <code>signed short int</code>
<code>unsigned short</code>	2 байта	Беззнаковый	<code>unsigned short int</code>
<code>int</code>	4 байта	Знаковый	<code>signed int</code>
<code>unsigned</code>	4 байта	Беззнаковый	<code>unsigned int</code>
<code>long</code>	4 байта	Знаковый	<code>long int</code> , <code>signed long</code> , <code>signed long int</code>
<code>unsigned long</code>	4 байта	Беззнаковый	<code>unsigned long int</code>
<code>long long</code>	8 байт	Знаковый	<code>long long int</code> , <code>signed long long</code> , <code>signed long long int</code>
<code>unsigned long long</code>	8 байт	Беззнаковый	<code>unsigned long long int</code>

То есть типы бывают “короткими” (`short`), обычными, длинными (`long`) и очень длинными (`long long`). Последний тип является расширением компилятора GNU C++ и не является стандартным типом для языка C++, поэтому он может отсутствовать в других реализациях языка или называться по-другому (например, в компиляторе Microsoft Visual C++ аналогичный тип называется `int64`). Чем “длиннее” тип, тем большее число различных значений он может принимать, тем больше памяти он занимает. Также типы бывают знаковыми (`signed`), которые могут принимать как положительные, так и отрицательные значения и беззнаковые (`unsigned`), которые принимают только неотрицательные значения.

Таблица значений, которые могут принимать различные типы:

Название	Размер	Минимальное значение	Максимальное значение
<code>short</code>	16 бит	$-2^{15} = -32768$	$2^{15}-1 = 32767$
<code>unsigned short</code>	16 бит	0	$2^{16}-1 = 65535$
<code>int</code> , <code>long</code>	32 бита	$-2^{31} = -2147483648$	$2^{31}-1 = 2147483647$
<code>unsigned</code> , <code>unsigned long</code>	32 бита	0	$2^{32}-1 = 4294967295$

long long	64 бита	$-2^{63} = -$ 9223372036854775808	$2^{63}-1 = 9223372036854775807$
unsigned long long	64 бита	0	$2^{64}-1 = 18446744073709551615$

На самом деле в стандарте языка C++ не оговорены конкретные значения размеров типов. Оговорено только то, что одинаковые знаковые и беззнаковые типы имеют одинаковые размеры, и размер меньшего типа всегда не превосходит размера большего типа. Вот какие размеры могут быть у этих типов в зависимости от разрядности процессора компьютера:

Тип	16-битный процессор	32-битный процессор	64-битный процессор
short	2 байта	2 байта	2 байта
int	2 байта	4 байта	4 байта
long	4 байта	4 байта	8 байт
long long	—	8 байт	8 байт

Действительные типы

Действительные (вещественные) числа представляются в виде чисел с десятичной точкой (а не запятой, как принято при записи десятичных дробей в русский текстах). Для записи очень больших или очень маленьких по модулю чисел используется так называемая запись “с плавающей точкой” (также называемая “научная” запись). В этом случае число представляется в виде некоторой десятичной дроби, называемой *мантиссой*, умноженной на целочисленную степень десяти (порядок). Например, расстояние от Земли до Солнца равно $1.496 \cdot 10^{11}$, а масса молекулы воды $2.99 \cdot 10^{-23}$.

Числа с плавающей точкой в программах на языке C++, а также при вводе и выводе записываются в виде мантиссы, затем пишется буква e, затем пишется порядок. Пробелы внутри этой записи не ставятся. Например, указанные выше константы можно записать в виде $1.496e11$ и $2.99e-23$. Перед самим числом также может стоять знак минус.

Для представления в памяти ЭВМ действительных чисел существует три типа:

Тип	Точность	Размер	Количество знаков мантиссы	Минимальное положительное значение	Максимальное значение
float	Одинарная	4 байта	7	$1.4e-45$	$3.4e38$
double	Двойная	8 байт	15	$5.0e-324$	$1.7e308$
long double	Расширенная	10 байт	19	$1.9e-4951$	$1.1e4932$

Операция деления

Для действительных чисел определены операции сложения, вычитания, умножения и деления.

При этом операция деления выполняется по-разному для переменных и констант целочисленного типа и для переменных и констант действительных типов. В первом случае деление производится нацело с отбрасыванием дробной части, во втором случае — деление производится точно и результатом является действительное число. Более точно, если делимое и делитель одновременно являются целочисленными константами или переменными целочисленных типов, то деление будет целочисленным, а если хотя бы одно из них действительное, то деление будет действительным. Например:

```
cout << 10 / 3 << endl;
cout << 10. / 3 << endl;
cout << 10 / 3. << endl;
cout << 10. / 3. << endl;
```

выведет 3 в первой строке и 3.33333 в остальных строках.

Результат выполнения деления не зависит от того, какой переменной будет присвоен результат. Если написать `double a = 10 / 3;`, то переменная `a` будет равна 3, так как деление $10/3$ будет целочисленным, независимо от того, чему будет присвоен результат.

Упражнение

Определите, чему будут равны следующие переменные

```
int a = 13 / 5;
int b = 13 % 5;
int c = 13.0 / 5;
double d = 13 / 5;
double e = 13 % 5;
double f = 13.0 / 5;
double g = 13 / 5 + 2 / 5;
double h = 13.0 / 5 + 2.0 / 5;
int i = 13.0 / 5 + 2.0 / 5;
```

Список функций

В стандартную математическую библиотеку языка Си (а, значит, и C++) входит множество специальных математических функций, которые нужно знать и уметь использовать. Для того, чтобы использовать эти функции в своей программе, необходимо подключить *заголовочный* файл, содержащий описания этих функций, что делается строчкой в начале программы:

```
#include <cmath>
```

Функция от одного аргумента вызывается, например, так: `sin(x)`. Вместо числа `x` может быть любое число, переменная или выражение. Функция возвращает значение, которое можно вывести на экран, присвоить другой переменной или использовать в выражении:

```
y = sin(x);
cout << sqrt(2) << endl;
```

Функция	Описание
---------	----------

Округление	
round	Округляет число по правилам арифметики, то есть $\text{round}(1.5) == 2$, $\text{round}(-1.5) == -2$
floor	Округляет число вниз (“пол”), при этом $\text{floor}(1.5) == 1$, $\text{floor}(-1.5) == -2$
ceil	Округляет число вверх (“потолок”), при этом $\text{ceil}(1.5) == 2$, $\text{ceil}(-1.5) == -1$
trunc	Округление в сторону нуля (отбрасывание дробной части), при этом $\text{trunc}(1.5) == 1$, $\text{trunc}(-1.5) == -1$
fabs	Модуль (абсолютная величина)
Корни, степени, логарифмы	
sqrt	Квадратный корень. Использование: $\text{sqrt}(x)$
cbrt	Кубический корень. Использование: $\text{cbrt}(x)$
pow	Возведение в степень, возвращает a^b . Использование: $\text{pow}(a, b)$
exp	Экспонента, возвращает e^x . Использование: $\text{exp}(x)$
log	Натуральный логарифм
log10	Десятичный логарифм
Тригонометрия	
sin	Синус угла, задаваемого в радианах
cos	Косинус угла, задаваемого в радианах
tan	Тангенс угла, задаваемого в радианах
asin	Арксинус, возвращает значение в радианах
acos	Арккосинус, возвращает значение в радианах
atan	Арктангенс, возвращает значение в радианах

Все перечисленные в таблице функции работают с типом `double`, этот тип считается основным типом для работы с действительными числами.

В: Целая часть

Дано положительное действительное число X . Выведите его целую часть.

Ввод	Вывод
3.14	3

С: Дробная часть

Дано положительное действительное число X . Выведите его дробную часть.

Ввод	Вывод
23.9	0.9

D: Первая цифра после точки

Дано положительное действительное число X . Выведите его первую цифру после десятичной точки. При решении этой задачи нельзя пользоваться условной инструкцией и циклом.

Ввод	Вывод
0.75	7

E: Гипотенуза

Даны длины катетов прямоугольного треугольника. Выведите длину его гипотенузы.

Ввод	Вывод
3 4	5

F: Площадь треугольника (формула Герона)

Даны длины сторон треугольника. Вычислите площадь треугольника.

Ввод	Вывод
3 4 5	6
1 1 1	0.433013

G: Часы - 1

С начала суток прошло N часов, M минут, S секунд ($0 \leq N < 12$, $0 \leq M < 60$, $0 \leq S < 60$). По данным числам N , M , S определите угол (в градусах), на который повернулась часовая стрелка с начала суток и выведите его в виде действительного числа.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Ввод	Вывод
1 2 6	31.05

H: Часы - 2

С начала суток часовая стрелка повернулась на угол в α градусов. Определите, на какой угол повернулась минутная стрелка с начала последнего часа. Входные и выходные данные — действительные числа.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Пример

Ввод	Вывод
190	120

I: Часы - 3

С начала суток часовая стрелка повернулась на угол в α градусов. Определите сколько полных часов, минут и секунд прошло с начала суток, то есть решите задачу, обратную задаче F. Запишите ответ в три целочисленные переменные и выведите их на экран.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Ввод	Вывод
31.05	1 2 6

J: Проценты

Процентная ставка по вкладу составляет P процентов годовых, которые прибавляются к сумме вклада. Вклад составляет X рублей Y копеек. Определите размер вклада через год.

Программа получает на вход числа целые P , X , Y и должна вывести два числа: величину вклада через год в рублях и копейках. Копейки необходимо округлить до целого числа по правилам арифметики.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Ввод	Вывод
12 179 0	200 48

K: Сложные проценты

Процентная ставка по вкладу составляет P процентов годовых, которые прибавляются к сумме вклада через год. Вклад составляет X рублей Y копеек. Определите размер вклада через K лет.

Программа получает на вход целые числа P , X , Y , K и должна вывести два числа: величину вклада через год в рублях и копейках. Копейки необходимо округлить до целого числа по правилам арифметики. Перерасчет суммы вклада (с округлением копеек) происходит ежегодно.

Ввод	Вывод
12 179 0 5	315 46

L*: Цена товара

Цена товара обозначена в рублях с точностью до копеек, то есть действительным числом с двумя цифрами после десятичной точки. **Запишите в две целочисленные переменные** стоимость товара в виде целого числа рублей и целого числа копеек и выведите их на экран.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Ввод	Вывод
10.35	10 35

М: $\pi^2/6$

По данному числу n вычислите сумму $1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots + \frac{1}{n^2}$

Решая эту задачу не забывайте, что $1/4 \neq 0$, $1/9 \neq 0$ и т.д.

Ввод	Вывод
3	1.36111

Знаете ли вы, что этот ряд сходится к $\pi^2/6$?

Н: Просто π

По данному числу n вычислите сумму $4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots\right)$

Операцией возведения в степень пользоваться нельзя.

Ввод	Вывод
2	3.46667

Этот ряд сходится к числу π .

О*: Система линейных уравнений - 1

Даны числа a, b, c, d, e, f . Известно, что система линейных уравнений

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

имеет ровно одно решение. Выведите два числа x и y , являющиеся решением этой системы.

Ввод	Вывод
1 0 0 1 3 3	3 3

R*: Система линейных уравнений - 2

Даны числа a, b, c, d, e, f . Решите систему линейных уравнений

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

Вывод программы зависит от вида решения этой системы.

Если система не имеет решений, то программа должна вывести единственное число 0.

Если система имеет бесконечно много решений, каждое из которых имеет вид $y=kx+b$, то программа должна вывести число 1, а затем значения k и b .

Если система имеет единственное решение (x_0, y_0) , то программа должна вывести число 2, а затем значения x_0 и y_0 .

Если система имеет бесконечно много решений вида $x=x_0, y$ — любое, то программа должна вывести число 3, а затем значение x_0 .

Если система имеет бесконечно много решений вида $y=y_0, x$ — любое, то программа должна вывести число 4, а затем значение y_0 .

Если любая пара чисел (x, y) является решением, то программа должна вывести число 5.

Ввод	Вывод
1 0 0 1 3 3	2 3 3
1 1 2 2 1 2	1 -1 1
0 2 0 4 1 2	4 0.5

Q*: Первая цифра после точки

Попробуйте еще раз решить следующую задачу.

Дано положительное действительное число X . Выведите его первую цифру после десятичной точки. При решении этой задачи нельзя пользоваться условной инструкцией и циклом.

Если у вас не получается решить эту задачу, прочтите примечание в конце листка

Ввод	Вывод
2.1	1

R*: Часы - 3

Вспомните задачу I и попробуйте еще раз решить следующую задачу.

С начала суток часовая стрелка повернулась на угол в α градусов. Определите сколько полных часов, минут и секунд прошло с начала суток, то есть решите задачу, обратную задаче F. Запишите ответ в три целочисленные переменные и выведите их на экран.

При решении этой задачи нельзя пользоваться условными инструкциями и циклами.

Ввод	Вывод
31.025	1 2 3

S*: Округление

Мнение конструкторов ЭВМ и математиков о том, что такое деление с остатком для отрицательных чисел расходится. Напомним, что при делении числа a на число b с остатком находятся такие числа q (частное) и r (остаток), что $a = qb + r$. При этом $|r| < |b|$. Если числа a и b — положительные, то выполняется более сильное неравенство: $0 \leq r < b$ и тогда деление с остатком определяется однозначно. Но если числа — отрицательные то деление с остатком на компьютере выполняется не так, как это кажется правильным математикам.

Вот как выполняется деление с остатком на компьютере для отрицательных чисел:

a	b	a / b	a % b
26	10	2	6
-26	10	-2	-6
26	-10	-2	6
-26	-10	2	-6

Таким образом, в компьютере целочисленное частное является результатом округления действительного частного в сторону нуля, как это делает функция `trunc`. Между тем, математик считает, что целочисленное частное — это целая часть от частного ($q = \lfloor a/b \rfloor$), то есть результат использования функции `floor`. Это даст такие результаты с точки зрения математики:

a	b	$\lfloor a/b \rfloor$	$A \bmod b$
26	10	2	6
-26	10	-3	4
26	-10	-3	-4
-26	-10	2	-6

Напишите программу, которая по данным числам a и b вычисляет их целочисленное частное и остаток от деления так, как это принято в математике. Программа не должна использовать действительные числа и циклы.

Если у вас не получаются задачи...

Если вам не удастся решить некоторые задачи, то посмотрите на следующую программу:

```
#include<iostream>
using namespace std;

int main()
{
    double x, y;
    x = 4.1;
    y = 4.2;
    cout << x << endl;
    cout << y << endl;
    cout.precision(20);
    cout << x << endl;
    cout << y << endl;
    return 0;
}
```

Инструкция `cout.precision(20)` устанавливает количество значащих цифр при выводе действительных чисел. Откомпилируйте и запустите эту программу, подумайте над результатом ее работы. Подумайте, как это может сказаться на решении заданий P, Q и S