

# Первая программа

- `#include <iostream>` и `using namespace std;` – для вывода текста
- `int main()` – точка входа; с неё начинается **любая** программа
- `cout << "текст";` – выводит текст на экран
- `endl;` – перевод строки

# Задачи

? Напишите программу, которая выводит на экран ваше имя.

? Выведите на экран звёздочки в виде прямоугольного треугольника.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

? Вычислите, используя арифметические операции и скобки:

$$25 + 17; \quad \frac{5}{4}; \quad 1 + \frac{1}{1 + \frac{1}{2}}.$$

# Основные типы данных

- `int` – целое число
- `float` – число с плавающей точкой
- `string` – строка
- `char` – символ
- `cin >> переменная;` – считывает значение в переменную

# Задачи

- ? Сложите два целых числа.
- ? Вычислите площадь квадрата по длине стороны.
- ? Переведите заданное количество метров в километры.
- ? Напечатайте последнюю цифру заданного натурального числа.
- ? Вычислите  $a^4$ , используя не более двух операций умножения.
- ? Вычислите  $a^{20}$ , используя не более пяти операций умножения.

# Материалы

**[savthe.com/edu](https://savthe.com/edu)**

- VimC++ (запускать ярлык GVim)
- Учебник по C++
- Шпаргалка по Vim

**[github.com/hant05080/lessons](https://github.com/hant05080/lessons)**

# Бисер

В шкатулке хранится разноцветный бисер или бусины). Все бусины имеют одинаковую форму, размер и вес. Бусины могут быть одного из  $N$  различных цветов. В шкатулке много бусин каждого цвета. Требуется определить минимальное число бусин, которые можно не глядя вытащить из шкатулки так, чтобы среди них гарантированно были две бусины одного цвета.

## Входные данные

На вход подаётся одно натуральное число  $N$  – количество цветов бусин ( $1 \leq N \leq 10^9$ ).

## Результат работы

Напечатайте одно целое число – минимальное количество бусин.

## Пример

Входные данные	Результат работы
3	4

# Следующее и предыдущее

Напишите программу, которая считывает целое число и выводит текст с упоминанием следующего и предыдущего для него чисел.

## Входные данные

На вход подаётся целое число, не превосходящее  $10^9$  по абсолютной величине.

## Результат работы

Напечатайте текст, аналогичный приведённому в примере.

## Пример

Входные данные	Результат работы
42	Следующее число после 42: 43 Предыдущее число перед 42: 41

# Магазин канцелярских товаров

Однажды, посетив магазин канцелярских товаров, Вася купил  $X$  карандашей,  $Y$  ручек и  $Z$  фломастеров. Известно, что цена ручки на 2 рубля больше цены карандаша и на 7 рублей меньше цены фломастера. Также известно, что стоимость карандаша составляет 3 рубля. Требуется определить общую стоимость покупки.

## Входные данные

На вход подаются 3 натуральных числа, не превосходящих  $10^9$

## Результат работы

Напечатайте одно натуральное число – стоимость покупки в рублях.

## Пример

Входные данные	Результат работы
1 1 1	20
1 2 3	49



# Сумма цифр

Найдите сумму цифр трёхзначного натурального числа.

## Входные данные

На вход подаётся трёхзначное натуральное число.

## Результат работы

Напечатайте одно натуральное число – сумму его цифр.

## Пример

Входные данные	Результат работы
100	1
123	6

# Инструкции ветвления

```
if (/* условие */)
{
    // некоторые действия
    // выполнятся, если условие верно
}
else
{
    // если условие неверно
}
```

# Операторы сравнения

- == равно (не путать с =)
- != не равно
- < меньше
- > больше
- <= меньше либо равно
- >= больше либо равно
- && логическое И
- || логическое ИЛИ

# Задачи

- ? Поменять местами значения двух переменных.
- ? Решить предыдущую задачу без дополнительной переменной.
- ? Вычислить модуль введённого числа.
- ? Определить, является ли введённое число чётным.
- ? Найти максимальное из двух чисел.
- ? Проверить, могут ли 3 заданных числа быть сторонами треугольника.

# Калькулятор

Напишите калькулятор, выполняющий одно из 4 арифметических действий над двумя заданными вещественными числами.

## Входные данные

На вход подаётся вещественное число  $a$ , символ  $s$  и вещественное число  $b$  ( $|a|, |b| \leq 1000, s \in \{ '+', '-', '*', '/' \}$ ).

## Результат работы

Напечатайте одно число – результат вычисления, либо сообщение об ошибке.

## Пример

Входные данные	Результат работы
2+3	5
3.14*2.72	8.5408
42/0	Ошибка

# Логические выражения

- `bool` – логический тип данных
- `!` – оператор отрицания
- `0 == false, !0 == true` // любое число, кроме нуля
- `a = !!a; // 0, если a была 0; иначе 1`
- Операторы сравнения возвращают результат логического типа

# МКАД

Длина Московской кольцевой автомобильной дороги – 109 километров. Байкер Вася стартует с первого километра МКАД и едет со скоростью  $v$  километров в час. На какой отметке он остановится через  $t$  часов?

## Входные данные

На вход подаются два целых числа  $t$  и  $v$  ( $0 \leq t, v \leq 40000$ ).

## Результат работы

Напечатайте единственное число от 1 до 109 – километр МКАД, на котором остановится Вася.

## Пример

Входные данные	Результат работы
60 2	12
109 42	1
0 146	1

# Счастливым билет

Вы пользуетесь общественным транспортом? Вероятно, вы расплачивались за проезд и получали билет с номером. Счастливым билетом называют такой билет с шестизначным номером, где сумма первых трех цифр равна сумме последних трех. Т.е. билет с номером 385916 – счастливый, т.к.  $3+8+5=9+1+6$ . Вам требуется написать программу, которая проверяет счастливость билета.

## Входные данные

На вход подаётся одно целое число  $N$  ( $0 \leq N < 10^6$ ).

## Результат работы

Напечатайте «YES», если билет с номером  $N$  счастливый и «NO» в противном случае.

## Пример

Входные данные	Результат работы
385916	YES
123456	NO



# Торт

На свой день рождения Петя купил красивый и вкусный торт, который имел идеально круглую форму. Петя не знал, сколько гостей придет на его день рождения, поэтому вынужден был разработать алгоритм, согласно которому он сможет быстро разрезать торт на  $N$  равных частей. Следует учесть, что разрезы торта можно производить как по радиусу, так и по диаметру. Помогите Пете решить эту задачу, определив наименьшее число разрезов торта по заданному числу гостей.

## Входные данные

На вход подаётся натуральное число  $N$  – число гостей, включая самого виновника торжества ( $N \leq 2 \cdot 10^9$ ).

## Результат работы

Напечатайте одно целое число – минимальное возможное число разрезов торта.

## Пример

Входные данные	Результат работы
2	1
3	3

# Ладья

Напомним, что в шахматах используется клеточная доска размером 8x8, где располагаются шахматные фигуры, которые могут перемещаться по определенным правилам. В частности, *ладья* может перемещаться на любое расстояние, как по вертикали, так и по горизонтали.

Требуется определить: может ли ладья выполнить ход из клетки с координатами  $(X_1, Y_1)$  в клетку с координатами  $(X_2, Y_2)$  на стандартной шахматной доске?

## Входные данные

На вход подаются 4 числа: начальная координата *ладьи*  $X_1, Y_1$  и конечная –  $X_2, Y_2$ . Гарантируется, что начальная и конечная координаты не совпадают.

## Результат работы

Напечатайте «YES», если ход допустим и «NO» в противном случае.

## Пример

Входные данные	Результат работы
4 3 7 3	YES
4 3 6 1	NO

## Короткая запись арифметических операций

- Вместо  $a = a @ b$ , где @ – знак арифметической операции, можно писать  $a @= b$ . Например,  $a += 2$ .
- Вместо  $a = a + 1$  и  $a = a - 1$  можно писать  $a++$  (или  $++a$ ) и  $a--$  (или  $--a$ ) соответственно.
- В логическую переменную можно записывать результат сравнения.

Например, вместо

```
if(a > 5) b = true;  
else b = false;
```

Можно писать

```
b = (a > 5);
```

# Инструкции циклов

```
while(/*условие*/)
{
    // тело цикла
}
```

Цикл `while` выполняется до тех пор, пока верно условие.

```
for(/*инициализация; условие; действие*/)
{
    // тело цикла
}
```

Цикл `for` удобно использовать, когда известно, сколько итераций нужно сделать.

Операторы прерывания цикла:

- `break`; – завершить выполнение цикла
- `continue`; – перейти к следующей итерации

# Задачи

- ? Напечатайте  $n$  звёздочек, не вводя дополнительную переменную.
- ? Напечатайте все целые числа (квадраты чисел; чётные числа) 0 до  $n$ .
- ? Напечатайте все делители числа  $n$ .
- ? Проверьте, является ли введенное натуральное число степенью тройки.
- ? Найдите максимальное из  $n$  натуральных чисел.
- ? Напечатайте квадраты целых чисел от 0 до  $n$ , не используя умножение и вложенные циклы.
- ? Проверьте, является ли натуральное число  $n$  простым.

# Демо ОГЭ 2018

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – минимальное число, оканчивающееся на 4.

## Пример

Входные данные	Результат работы
4 24 13 14 34	14

# Слон

Напомним, что в шахматах используется клеточная доска размером  $8 \times 8$ , где располагаются шахматные фигуры, которые могут перемещаться по определенным правилам. В частности, *слон* может перемещаться на любое расстояние по диагонали.

Требуется определить: может ли слон выполнить ход из клетки с координатами  $(X_1, Y_1)$  в клетку с координатами  $(X_2, Y_2)$  на стандартной шахматной доске?

## Входные данные

На вход подаются 4 числа: начальная координата *слона*  $X_1, Y_1$  и конечная –  $X_2, Y_2$ . Гарантируется, что начальная и конечная координаты не совпадают.

## Результат работы

Напечатайте «YES», если ход допустим и «NO» в противном случае.

## Пример

Входные данные	Результат работы
5 4 7 2	YES
5 4 4 6	NO

# Шоколадка

Требуется определить, можно ли от шоколадки размером  $n \times m$  долек отломить  $k$  долек, если разрешается сделать один разлом по прямой между дольками (то есть разломить шоколадку на два прямоугольника).

## Входные данные

На вход подаются 3 натуральных числа ( $n, m \leq 40000, k \leq 2 \cdot 10^9$ ).

## Результат работы

Напечатайте «YES», если возможно отломить указанное число долек и «NO» в противном случае.

## Пример

Входные данные	Результат работы
3 2 4	YES
3 2 1	NO
2 2 6	NO



# Монетки

На столе лежат  $n$  монеток. Некоторые из них лежат вверх решкой, а некоторые – гербом. Определите минимальное число монеток, которые нужно перевернуть, чтобы все монетки были повернуты вверх одной и той же стороной.

## Входные данные

На вход подаётся количество монеток  $n$ , а следом  $n$  чисел: 0, если монетка лежит решкой вверх, или 1, если гербом вверх.

## Результат работы

Напечатайте минимальное число монеток, которые нужно перевернуть.

## Пример

Входные данные	Результат работы
5 1 0 1 1 0	2
4 0 0 1 0	1

# Перепись

В доме живет  $n$  жильцов. Однажды решили провести перепись всех жильцов данного дома и составили список, в котором указали возраст и пол каждого жильца. Требуется найти номер самого старшего жителя мужского пола.

## Входные данные

На вход подаётся натуральное число  $n$ , а следом за ним –  $n$  строк с информацией о жильцах. Каждая строка содержит натуральное число  $a$  ( $1 \leq a \leq 100$ ) и символ  $s$  ('М' – мужчина или 'F' – женщина).

## Результат работы

Напечатайте порядковый номер самого старшего мужчины, либо -1, если жильцов мужского пола нет.

## Пример

Входные данные	Результат работы
3 25 М 100 F 70 М	3

# Автобусная экскурсия

Для обзорной экскурсии по городу был заказан двухэтажный автобус высотой **437** сантиметров. На экскурсионном маршруте встречаются  $N$  мостов. Организаторы обеспокоены тем, что автобус может не проехать под одним из них. Им удалось выяснить точную высоту каждого моста. Автобус может проехать под мостом, если высота моста превосходит высоту автобуса. Помогите организаторам узнать, закончится ли экскурсия благополучно, а если нет, то установить, где произойдет авария.

## Входные данные

На вход подаётся число  $n$ , а следом за ним –  $n$  натуральных чисел (высоты мостов).

## Результат работы

Напечатайте `No crash`, если экскурсия закончится благополучно, в противном случае – `Crash k`, где  $k$  – номер моста, где произойдёт авария.

## Пример

Входные данные	Результат работы
3 763 245 113	Crash 2

# Массивы

- Массив – последовательность элементов одного типа с общим именем
- Обращение к элементу массива осуществляется через его индекс
- Объявление массива в общем виде: `type name[size];`
- Нумерация элементов массива начинается **с нуля**
- Размером массива может быть только **константа**
- Для объявления константы используется спецификатор `const`

Пример:

```
int a[10]; // массив из 10 целых чисел
a[3] = 42; // инициализация четвёртого элемента массива
const int n = 8; // целочисленная константа
double b[n] // массив из 8 вещественных чисел
n = 12; // ошибка: изменение константы
a[10] = 5; // ошибка: выход за границу массива (UB)
int m = 17;
char c[m]; // ошибка: m – не константа (IDB)
```

# Случайные числа в C++

- `rand()`; – случайное целое неотрицательное число
- `rand()%n`; – случайное целое число от нуля до  $n-1$

Пример: заполним массив случайными числами от -25 до 25

```
const int n = 10; // размер массива – константа
int a[n];
for (int i = 0; i < n; ++i) // i = 0,1,...,9
    a[i] = rand()%51 - 25; // rand()%51 от 0 до 50
// выведем содержимое массива на экран
for (int i = 0; i < n; ++i)
    cout << a[i] << ' '; // выводим через пробел
cout << endl; // после вывода массива переходим на новую строку
```

# Задачи

В начале решения каждой задачи объявите массив из 20 целых чисел, заполните его случайными числами от -50 до 50 и выведите на экран содержимое этого массива.

? Вычислите сумму элементов массива.

? Вычислите сумму квадратов положительных элементов массива.

? Вычислите среднее арифметическое отрицательных элементов массива.

? Вычислите произведение элементов массива, значения которых попадают в отрезок  $[l, r]$  (где  $0 \leq l < r < 20$ ).

? Замените в массиве все нечетные элементы значениями их индексов.

? Определите значение максимального элемента в массиве и его положение.

? В массиве переставьте 1-й и 2-й элементы, 3-й и 4-й, 5-й и 6-й и т.д.

? Замените нулями все элементы массива, являющиеся делителями максимального элемента.

? Переставьте минимальный и максимальный элементы массива.

# Символы

```
char a = 'A'; // объявление символьной переменной
// Символы хранятся в виде кодов, с ними можно работать, как с
char b = a + 3; // D                                целыми числами
char c = 42; // *
// В таблице ASCII цифры и буквы латинского алфавита расположены
if ('X' < 'Z') //                                     по порядку
    cout << (int)'X' << ' ' << (int)'Z' << ' ' << endl; // 88 90
cout << '9' - '4' << endl; // 5
// Вместо символа можно явно выводить его код и наоборот
cout << 'A' << ' ' << (int)'A' << endl; // A 65
cout << 61 << ' ' << (char)61 << endl; // 61 =
```

# Задачи

- ? Определите, является ли введенный символ цифрой.
- ? Определите, сколько из 10 введенных символов являются буквами.
- ? Если введенный символ является строчной буквой латинского алфавита, переведите его в верхний регистр.



# Строки

```
string s = "hello"; // объявление строки
int n = s.size(); // размер строки (n == 5)
// Работа со строкой аналогична работе с массивом из char
cout << s[3] << s[4] << s[2] << endl; // lol
s[0] = 'H';
// Выводить строку можно целиком
cout << s << endl; // Hello
// Строки можно складывать (конкатенация) и сравнивать
string w = ", world!"; s += w;
cout << s << ' ' << s.size() << endl; // Hello, world! 13
// Удаление из строки
s.erase(5, 2); // (с какого символа, сколько символов)
cout << s << ' ' << s.size() << endl; // Helloworld! 11
// Вставка в строку
s.insert(5, ", new "); // (с какого символа, строка для вставки)
cout << s << ' ' << s.size() << endl; // Hello, new world! 17
// cin считывает строку до первого пробела
getline(cin, s); // так можно считать строку с пробелами
```

## Задачи

? Замените в строке все символы  $x$  на  $y$ .

? Определите, каких букв в строке больше: маленьких или больших.

? Проверьте, верно ли расставлены скобки в строке. Например, в строках: " $g(5)-2$ ", " $2 + (a-(b+5))$ ", " $((2-3) + (c-d))$ " скобки расставлены верно, а в строках " $(u-3($ ", " $15+a)$ ", " $2 - ) 3 + (7) ($ " – нет.

? Замените в строке все символы  $x$  на число 58.

## Динамические массивы

Обычные массивы иногда оказываются неудобными. Часто размер данных неизвестен, а создание массивов «с запасом» безосновательно увеличивает требуемый объем оперативной памяти. Стандартный контейнер `vector` работает, как обычный массив, но предоставляет удобные функции для работы с данными и не требует предварительного задания количества элементов.

```
#include<vector>
// ...
vector<int> v; // объявление вектора: vector<TYPE> NAME;
cout << v.size() << endl; // 0 (вектор пуст)
v.push_back(42); // добавление элемента
v.push_back(23);
v.push_back(-34);
cout << v.size() << endl; // 3
v[1] += 7;
// проход по элементам вектора
for (int i = 0; i < v.size(); ++i)
    cout << v[i] << ' '; // 42 30 -34
```

# Контроперация

Хакер Василий получил доступ к классному журналу и хочет заменить все свои минимальные оценки на максимальные. Напишите программу, которая заменяет оценки Василия, но наоборот: все максимальные – на минимальные.

## Входные данные

На вход подаются целые ненулевые числа – оценки Василия. Признак окончания последовательности – число 0.

## Результат работы

Напечатайте исправленные оценки, сохранив их порядок. Ноль выводить не надо.

## Пример

Входные данные	Результат работы
1 3 3 3 4 0	1 3 3 3 1
5 4 2 2 4 2 2 5 0	2 4 2 2 4 2 2 2
17 23 -5 7 23 0	17 -5 -5 7 -5

# Налоги

В государстве действуют несколько фирм. Прибыль  $i$ -ой фирмы равна  $V_i$  рублей в год. У царя есть любимые фирмы, а есть нелюбимые, поэтому налог для всех фирм разный и назначается царем в индивидуальном порядке. Налог на  $i$ -ую фирму равен  $p_i$  процентов. Собиратели статистики решили посчитать, с какой фирмы в казну идет наибольший доход. Помогите им в этой задаче.

## Входные данные

На вход подаются целые неотрицательные числа  $V_i$ , признак окончания последовательности – число -1. Далее идут числа  $p_i$ .

## Результат работы

Напечатайте одно число - номер фирмы, от которой государство получает наибольший налог. Если таких фирм несколько, выведите фирму с наименьшим номером.

## Пример

Входные данные	Результат работы
1 2 -1 3 2	2
100 1 50 -1 0 100 3	3

## ОГЭ-2

Ученики 4 класса вели дневники наблюдения за погодой и ежедневно записывали дневную температуру. Найдите самую низкую температуру за время наблюдения.

### Входные данные

Программа получает на вход количество дней, в течение которых проводилось измерение температуры  $N$  ( $1 \leq N \leq 31$ ), затем для каждого дня вводится температура.

### Результат работы

Выведите минимальную температуру. Если температура опускалась ниже  $-15$  градусов, выведите YES, иначе выведите NO.

### Пример

Входные данные	Результат работы
4 -5 12 -2 8	-5 NO

## ОГЭ-3

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму всех двузначных чисел, кратных 8.

### Входные данные

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

### Результат работы

Программа должна вывести одно число: сумму всех двузначных чисел, кратных 8.

### Пример

Входные данные	Результат работы
17 16 32 160 0	48

# Считалка

Для выбора водящего в детской игре  $N$  человек становятся в круг, после чего произносится считалка. На первом слове считалки указывается на первого человека в кругу, на втором слове – на второго человека и т. д. После  $N$ -го человека снова идёт первый человек (все люди в кругу пронумерованы числами от 1 до  $N$ , круг зацикливается, после человека с номером  $N$  идёт человек с номером 1). Всего в считалке  $M$  слов. Определите, на какого человека придётся последнее слово считалки.

## Входные данные

Программа получает на вход два целых положительных числа. Первое число  $N$  – количество людей в кругу. Второе число  $M$  – количество слов в считалке. Оба числа не превосходят  $10^9$ .

## Результат работы

Программа должна вывести одно целое число от 1 до  $N$  – номер человека в кругу на которого придётся последнее слово считалки.

## Пример

Входные данные	Результат работы
10 25	5



# Функции

Решение любой задачи естественно стараться свести к решению нескольких маленьких подзадач. Если в программе приходится несколько раз выполнять один и тот же алгоритм, то имеет смысл выделить его в подпрограмму и вызывать ее при необходимости. Такая подпрограмма будет называться **функцией**.

Функция имеет следующий вид:

```
тип имя(список аргументов)
{
    тело функции
}
```

Аргументы перечисляются через запятую с указанием типов. Если аргументов нет, ставятся пустые скобки.

# Функции: возвращаемое значение

Функции в языке C++ делятся по смыслу на 2 типа.

- Тип `void`: функция выполняет определённые действия (чтение данных, вывод на экран, и т.д.). Точка выхода из функции – её последняя команда или `return`;
- Любой другой тип (`int`, `bool`, `float`, ...): функция вычисляет некоторое значение и **возвращает** его. Точка выхода из функции – `return <значение>;`

Функции типа `void` вызываются по её имени со списком аргументов:

```
print("hello");
```

После имени функции круглые **скобки ставятся обязательно**, даже если функция не имеет аргументов. Функция должна быть объявлена до того места, где она будет вызвана.

При вызове функций других типов, как правило, сохраняется или используется как-то иначе их возвращаемое значение:

```
int m = max(2, 3);
```

## Функции: пример

```
void print(string text)
{
    cout << text << endl;
    // return; – не обязателен
}
int max(int a, int b) // тип нужно указывать у каждого аргумента
{
    // если a > b, функция завершит работу и вернёт значение a
    if (a > b) return a;
    // здесь можно не писать else, т.к. мы попадём сюда только
    // в том случае, если условие не выполнилось
    return b;
}
int main()
{
    print("hello");
    cout << max(4, 2) << endl; // выводим возвращаемое значение
}
```

# Унарная система счисления

В унарной системе счисления числа записываются как соответствующее количество "палочек". Например, число 3 записывается как `|||`, а число 7 – как `|||||||`. Напишите программу, которая переводит координаты точки в пространстве в унарную систему счисления. Перевод числа реализуйте в виде отдельной функции.

## Входные данные

На вход программе подаются 3 целых числа  $x, y, z \geq 0$ .

## Результат работы

Напечатайте значения чисел в унарной системе, каждое с новой строки.

## Пример

Входные данные	Результат работы
2 3 4	x:    y:     z:

\*Измените функцию перевода так, чтобы она работала и для отрицательных чисел.

# Палиндромы

Палиндромами называются такие числа, десятичная запись которых читается одинаково слева направо и справа налево. Выясните, какие из заданных чисел являются палиндромами. (Подсказка: реализуйте вспомогательную функцию разворота числа в десятичной записи.)

## Входные данные

На вход программе подаётся натуральное число  $n$ , а следом за ним –  $n$  чисел  $x_i \geq 0$ .

## Результат работы

Для каждого числа в отдельной строке напечатайте информацию о том, является ли оно палиндромом.

## Пример

Входные данные	Результат работы
3 9 146 123321	9: Yes 146: No 123321: Yes

# Суммы цифр

Учительница записала на доске целое число  $N$ . Вовочка подсчитал сумму цифр этого числа и записал ее ниже. С полученным числом он проделал то же самое, и продолжал выписывать числа до тех пор, пока два последних записанных числа не совпали. Ваша задача – найти сумму  $S$  всех выписанных на доску чисел. Реализуйте подсчёт суммы цифр в числе в виде отдельной функции.

## Входные данные

На вход подаётся натуральное число  $N \leq 2 \cdot 10^9$ .

## Результат работы

Напечатайте натуральное число  $S$ .

## Пример

Входные данные	Результат работы
34	48
1234	1246
987654	987711