Skillbox Python – Часть 1

Рекомендованные книги для параллельного чтения с курсом:

Пол Бэрри — Изучаем программирование на Python

Рейтц Кеннет, Шлюссер Таня — Автостопом по Python

Марк Лутц — Learning Python / Изучаем Python. 5-е издание, 2019. Книга-справочник

Лутц Марк — Python. Карманный справочник

# Модуль 1

## 1.3 Знакомство с онлайн-редактором. Первая программа

**Задание 1**

Повторите действия из урока и создайте программу, которая выводит в консоль текст Hello World!:

Зайдите в repl.it.

Создайте новый проект, язык Python.

В examples выберите пример Hello World.

Запустите программу, посмотрите, что она выводит в консоль.

Теперь измените программу, чтобы она выводила фразу ‘Привет Мир!’:

Замените Hello World! на нужный текст.

Запустите программу ещё раз, убедитесь, что всё работает.

**Задание 2**

Дополните программу из предыдущего задания так, чтобы она вывела в консоль приветствие, ваше имя и любимый цвет в отдельных строчках. Затем запустите программу и проверьте результат.

Теперь попробуйте поменять местами строчки программы, например первую и третью. В результате выведется сначала цвет, затем имя и в конце — приветствие.

## 1.4 Первые ошибки

Найдите и исправьте ошибки в коде. Для этого скопируйте код в repl.it, посмотрите, какие ошибки там есть, и исправьте их. Убедитесь, что программа работает.

1.

Print('Hello World')

2.

print 'Hello World'

3.

print('Hello World")

4.

print (Hello)

5.

print('Привет',')

print ('Меня зовут Вася'')

print 'и я умнее всех')

## 1.7 Практическая работа

**Задача 1. «Доска»**

Исправьте программу так, чтобы в результате её выполнения на экран вывелась «доска».

print('-'---------')

print(' |00000000| ')

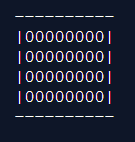
print('''|00000000|'')

print(' |00000000| ')

print('|00000000| ')

print('-'---------")

Вывод:



**Что оценивается**

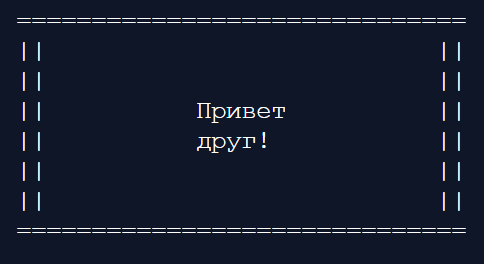
Результат вывода корректен и соответствует картинке.

Отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно.

Нет пробелов и внутри скобок.

**Задача 2. Приветствие**

Напишите программу, которая выводит такое «окно» на экран. Размер окна может быть любым, вместо слова «друг» можете написать своё имя. Постарайтесь добиться результата из третьего варианта.



**Что оценивается**

Результат вывода корректен и соответствует картинке.

Отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно.

Нет пробелов и внутри скобок.

**Советы и рекомендации**

Обратите внимание на правильное использование repl.it.

Как отправить задание на проверку

Как результат работы отправьте преподавателю ссылку на ваш реплит module1 через форму для сдачи домашнего задания.

# Модуль 2

## 2.1 Переменные

**Задача 1. Моя первая переменная**

Создайте переменную, присвойте ей текст «Моя первая переменная!» и выведите значение переменной на экран. Для этого:

1. Назовите переменную, используйте только буквы латинского алфавита. Пример имени: first.

2. После этого сделайте присваивание (знак = ).

3. Вставьте нужный текст.

4. Вставьте команду вывода переменной на экран в следующей строке программы. Кавычки здесь не нужны.

5. Запустите программу и убедитесь, что в консоль выводится сообщение: «Моя первая переменная!»

**Задача 2. Две переменные**

Продолжая предыдущую программу, создайте ещё одну переменную с другим именем (например second), присвойте ей значение «И не последняя!» и выведете её на экран. В результате работы программы в консоли должно отобразиться две строки: «Моя первая переменная!» и «И не последняя!»

**Задача 3. Ошибки в коде**

Дана программа:

my color@ = 'Зелёный'

\*sport& = 'Бег'

print('Меня зовут')

print(name)

name = 'Иван'

print('Мой любимый цвет:')

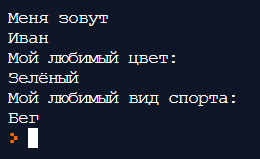
print(my\_color)

print('Мой любимый вид спорта:')

print(спорт)

Скопируйте программу в редактор и исправьте ошибки в названиях и выводе переменных. При необходимости переставляйте строчки местами.

Проверьте правильность работы программы.



## 2.2 Оператор ввода input

**Задача 1. Приветствие пользователя**

Повторите действия из урока и создайте программу, которая запрашивает у пользователя имя и затем выводит для него приветствие. Для этого:

Создайте переменную и присвойте ей команду input(). Внутри скобок команды напишите приглашение, например: «Введите имя».

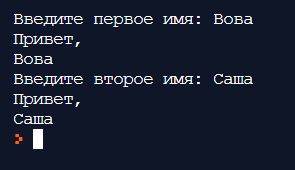
В следующих двух строках напишите вывод приветствия и переменной на экран.

Запустите программу, введите в консоль имя и нажмите Enter.

Убедитесь, что программа работает верно.

**Задача 2. Приветствие для двух пользователей**

Продолжая предыдущую программу, запросите у пользователя второе имя и выведите второе приветствие на экран. Пример результата программы:



**Задача 3. Неверная программа**

Ниже приведена программа, которая должна запрашивать фамилию пользователя и выводить её на экран. Но её написали неправильно. Скопируйте программу в редактор и исправьте все ошибки. Убедитесь, что программа работает.

sur name = inpu('Введите фамилию: ')

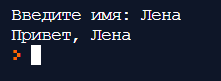
print('Ваша фамилия - ')

print('sur name')

## 2.3 Строки и конкатенация

**Задача 1. Вывод в одну строку**

Напишите программу, которая запрашивает у пользователя его имя и выводит для него приветствие в одну строку, как в примере ниже. Не забывайте ставить запятую при перечислении объектов внутри print!



**Задача 2. Конкатенация**

Используя программу из предыдущего задания, добавьте вывод восклицательного знака в конец приветственной строки. Проверьте программу.

Затем попробуйте использовать конкатенацию для вывода строки. Для этого в команде print уберите запятые между объектами и вместо них поставьте знак «+». Снова проверьте программу. Результат должен быть такой же.

**Задача 3. Склейка**

Напишите программу, которая запрашивает на вход два слова, записывает их в разные переменные, затем склеивает их и выводит на экран полученную строку.

**Результат будет такой:**

Введите первое слово: Всем

Введите второе слово: привет

Всем привет

## 2.5 Итоги пройденных тем. Проверьте себя

**Задача 1. Пропавшая переменная**

Что нужно сделать

Найдите в программе необъявленную переменную и объявите её, присвоив ей значение ‘Кот’.

client = ‘Петя’

print(client)

print(‘ и ’)

print(pet)

Что оценивается

Результат вывода корректен, нет ошибок. Присвоено значение переменной pet.

Отсутствуют пробелы после print и перед скобками: print () — неверно, print() — верно.

Нет пробелов и внутри скобок.

Знак равенства выделен пробелами с двух сторон: client = ‘Петя’ — верно, client=‘Петя’ — неверно.

**Задача 2. Цвета**

Что нужно сделать

Исправьте программу так, чтобы в результате её выполнения на экран в одну строку выводился текст: Red Blue Green RedGreenBlue Blue GreenBlue.

r = 'Red'

g = 'Green'

b = 'Blue'

print(b, r, g, b, g + b, b + b + g, b)



**Что оценивается**

Результат вывода соответствует картинке (тексту).

После запятой стоит пробел, перед запятой пробела нет.

Знак конкатенации (+) выделен пробелами с двух сторон.

**Задача 3. Животные**

Создайте две переменные с именами «Первое животное» и «Второе животное» на английском языке. Запишите в первую переменную слово «Заяц», а во вторую — «Черепаха». Используя эти переменные, выведите на экран текст «Заяц спит, Черепаха идёт» в одну строку.

**Что оценивается**

Input содержит корректное приглашение для ввода: input() — неверно.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

После запятой стоит пробел, перед запятой пробела нет.

Отсутствие пробелов после print и перед скобками: print () — неверно, print() — верно.

**Задача 4. Вход в систему**

Исправьте программу и допишите необходимые команды для получения нужного результата. Будьте внимательны при исправлении и помните о правилах названия переменных.

Программа:

first name = input('Введите имя пользователя: ')

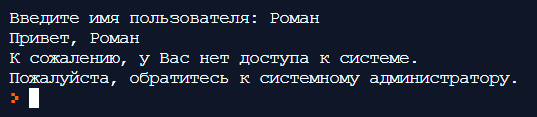
grииting = 'Утро доброе'

print(greeting, first name)

intro = "К сожалению, у Вас нет доступа к системе."

info == "Пожалуйста, обратитесь к системному администратору."

Ожидаемый результат:



**Что оценивается**

Результат вывода соответствует картинке.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Нет смешения символов из разных алфавитов в именах переменных.

Пробелы после запятых, пробелы при бинарных операциях.

После запятой стоит пробел, перед запятой пробела нет.

Отсутствуют пробелы после print и перед скобками: print () — неверно, print() — верно.

Нет пробелов и внутри скобок.

**Задача 5. Полёт**

Напишите программу для сервиса заказа билетов, которая запрашивает у пользователя город вылета и город прилёта. Затем выведите их в одну строку через тире. Обратите внимание на свои переменные: названия должны отражать содержимое.

**Что оценивается**

Input содержит корректное приглашение для ввода: input() — неверно.

Результат вывода соответствует заданию.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях (конкатенация в данной задаче).

После запятой стоит пробел, перед запятой пробела нет.

Отсутствуют пробелы после print и перед скобками: print () — неверно, print() — верно.

Нет пробелов и внутри скобок.

**Задача 6. Повышенная сложность. Обмен значений двух переменных**

Дана программа, которая запрашивает у пользователя два слова, а затем выводит их на экран два раза. Скопируйте эту программу в редактор и проверьте.

a = input('Введите первое слово: ')

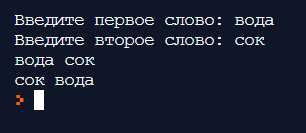
b = input('Введите второе слово: ')

print(a, b)

# стереть эту строчку и вставить свой код здесь

print(a, b)

Задача: поменять значения переменных a и b местами. Изменять, удалять, менять местами первую, вторую, третью и последнюю строчки нельзя. Но в четвёртую строку можно вставлять сколько угодно кода, не трогая последний принт. Пример результата работы программы:



**Что оценивается**

Результат вычислений и вывода корректен. Строки с print остались неизменными, не изменён порядок вывода переменных.

## 2.6 Практическая работа

Домашнее задание принимается только через Fork или корректно оформленный repl.it. Для выполнения домашнего задания используйте подготовленный нами шаблон.

**Задача 1. Пропавшая переменная**

Найдите в программе необъявленную переменную и объявите её, присвоив ей значение ‘Кот’.

client = ‘Петя’

print(client)

print(‘ и ’)

print(pet)

**Что оценивается**

Результат вывода корректен, нет ошибок. Присвоено значение переменной pet.

Отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно.

Нет пробелов и внутри скобок.

Знак равенства выделен пробелами с двух сторон: “client = ‘Петя’” — верно, “client=‘Петя’” — неверно.

**Задача 2. Цвета**

Исправьте программу так, чтобы в результате её выполнения на экран в одну строку выводился текст: Red Blue Green RedGreenBlue Blue GreenBlue.

r = 'Red'

g = 'Green'

b = 'Blue'

print(b, r, g, b, g + b, b + b + g, b)



**Что оценивается**

Результат вывода соответствует картинке (тексту).

После запятой стоит пробел, перед запятой пробела нет.

Знак конкатенации (+) выделен пробелами с двух сторон.

**Задача 3. Животные**

Создайте две переменные с именами «Первое животное» и «Второе животное» на английском языке. Запишите в первую переменную слово «Заяц», а во вторую — «Черепаха». Используя эти переменные, выведите на экран текст «Заяц спит, Черепаха идёт» в одну строку.

**Что оценивается**

input содержит корректное приглашение для ввода: “input()” — неверно;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3);

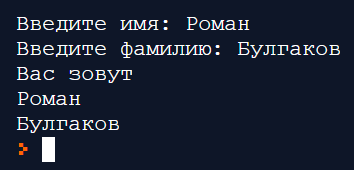
после запятой стоит пробел, перед запятой пробела нет;

отсутствие пробелов после print и перед скобками: “print ()” — неверно, “print()” — верно.

**Задача 4. Информация о пользователе**

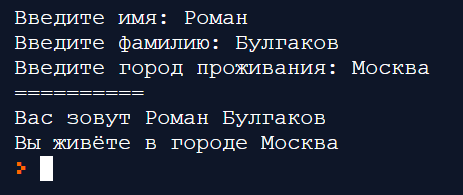
Напишите программу, которая запрашивает некоторые данные у пользователя, затем выведите их на экран, как показано ниже. Данные должны лежать в разных переменных.

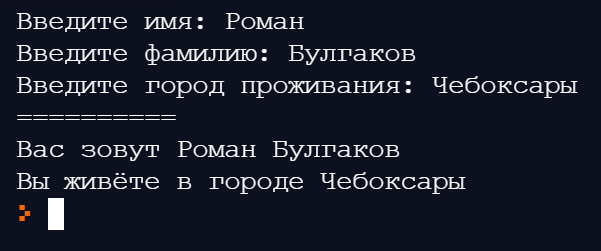
Вариант 1. Запросите имя и фамилию и выведите их на экран построчно. Результат должен быть таким (с вашим именем и фамилией, конечно):



Вариант 2. Запросите имя, фамилию и город проживания, затем выведите их на экран в две строки: первая — «Вас зовут» и далее имя и фамилия, вторая — «Вы живёте в городе» и далее город. Для красоты отделите в консоли ввод и вывод данных,

как в примере:





**Что оценивается**

input содержит корректное приглашение для ввода: “input()” — неверно;

результат вывода соответствует примеру;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3);

пробелы после запятых, пробелы при бинарных операциях;

после запятой стоит пробел, перед запятой пробела нет;

отсутствие пробелов после print и перед скобками: “print ()” — неверно, “print()” — верно;

отсутствуют пробелы внутри скобок.

**Задача 5. Вход в систему**

Исправьте программу и допишите необходимые команды для получения нужного результата. Будьте внимательны при исправлении и помните о правилах названия переменных.

Программа:

first name = input('Введите имя пользователя: ')

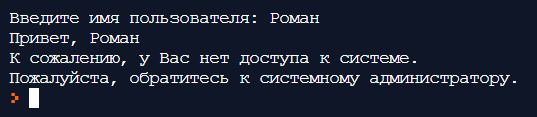
grииting = 'Утро доброе'

print(greeting, first name)

intro = "К сожалению, у Вас нет доступа к системе."

info == "Пожалуйста, обратитесь к системному администратору."

Ожидаемый результат:



**Что оценивается**

Результат вывода соответствует картинке.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Нет смешения символов из разных алфавитов в именах переменных.

Пробелы после запятых, пробелы при бинарных операциях.

После запятой стоит пробел, перед запятой пробела нет.

Отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно.

Нет пробелов и внутри скобок.

**Задача 6. Полёт**

Напишите программу для сервиса заказа билетов, которая запрашивает

у пользователя город вылета и город прилёта. Затем выведите их в одну строку

через тире. Обратите внимание на свои переменные: названия должны отражать содержимое.

Что оценивается

input содержит корректное приглашение для ввода: “input()” — неверно;

результат вывода соответствует заданию;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3);

пробелы после запятых, пробелы при бинарных операциях (конкатенация в данной задаче);

после запятой стоит пробел, перед запятой пробела нет;

отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно;

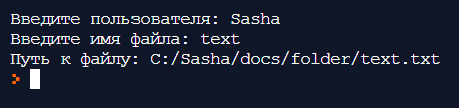
нет пробелов и внутри скобок.

**Задача 7. Путь к файлу**

К каждому файлу на компьютере можно узнать путь. Выглядит он примерно так:

'C:/user/docs/folder/new\_file.txt'

Напишите программу, которая запрашивает у пользователя его имя и имя файла (переменные user и new\_file соответственно). Используя операцию конкатенации, выведите путь к файлу на экран. Пример результата:



**Что оценивается**

результат вывода соответствует картинке;

input содержит корректное приглашение для ввода: “input()” — неверно;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3);

пробелы после запятых, пробелы при бинарных операциях (конкатенация в данной задаче);

после запятой стоит пробел, перед запятой пробела нет;

отсутствуют пробелы после print и перед скобками: “print ()” — неверно, “print()” — верно;

нет пробелов и внутри скобок.

**Задача 8. По желанию. Обмен значений двух переменных**

Дана программа, которая запрашивает у пользователя два слова, а затем выводит их на экран два раза. Скопируйте эту программу в редактор и проверьте.

a = input('Введите первое слово: ')

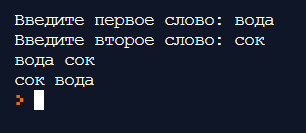
b = input('Введите второе слово: ')

print(a, b)

# стереть эту строчку и вставить свой код здесь

print(a, b)

Задача: поменять значения переменных a и b местами. Изменять, удалять, менять местами 1-ю, 2-ю, 3-ю и последнюю строчки нельзя. Но в 4-ю строку можно вставлять сколько угодно кода, не трогая последний принт. Пример результата работы программы:



**Что оценивается**

Результат вычислений и вывода корректен. Строки с print остались неизменными,

не изменён порядок вывода переменных.

Что оценивается (общее)

Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками:

“print ()”,“input ()” — неверно, “print()” — верно.

**Советы и рекомендации**

Учитывайте успеваемость — если вы легко справляетесь с заданиями, можете потратить больше времени на работу над стилем.

Ознакомьтесь со стилем оформления программ на языке Python [PEP8](https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html).

Оригинальный документ [PEP8](https://peps.python.org/pep-0008/) (на английском).

# Модуль 3

## 3.2 Числа и арифметические операции с ними

**Задача 1. Выражения**

Третьекласснику Вове задали домашнее задание по математике, и папа-программист сказал, что вечером обязательно проверит его. Но Вове лень считать, поэтому он пишет программы на Python.

Посчитайте и последовательно выведите на экран значения математических выражений, по одному в каждой строчке, чтобы Вове не досталось от папы:

6 умножить на 39;

3 в степени 5;

Минус 8 разделить на минус 4;

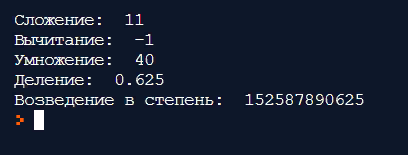
10 разделить на 2 и прибавить 6.

**Задача 2. Эти замечательные астрологи**

Венера, преподаватель Максима по астрологии и нумерологии, не верит, что математические операции работают так, как описывает математика. Напишите для Максима программу, которая демонстрировала бы работу основных операций, чтобы убедить Венеру.

Создайте три переменных, каждой из них присвойте любые числа, например 5, 4, 2 (кстати, не забывайте про множественное присваивание).

Затем выведите результат сложения, вычитания, умножения, деления и возведения в степень всех трёх переменных в отдельных строчках. Результат с 5, 4 и 2 должен выглядеть так:



**Задача 3. Деление на ноль**

Проверим школьное правило «На ноль делить нельзя». Напишите программу, которая выводит результат деления числа (например 65) на ноль. Какой будет ответ?

**Тизер к заданию**

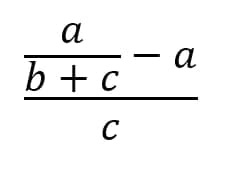
Правильный ответ: деление на ноль в питоне порождает ошибку ZeroDivisionError, это означает, что на ноль делить нельзя. Однако в будущем мы узнаем, что делить на ноль всё-таки можно — и даже получать бесконечность, но это уже совсем другая история :)

## 3.3 Приоритет арифметических операций

**Задача 1. Фруктовый сад**

Маша проводит исследования корреляции размеров фруктов в зависимости от их расположения на участке. Она взяла яблоко, грушу и персик диаметрами 8, 5 и 3 сантиметра соответственно и вычислила коэффициент зависимости по своей формуле.

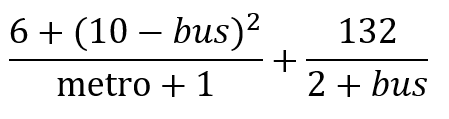
Реализуйте программу, которая вычисляет значение выражения



где a, b и c — переменные, в которых лежат числа 8, 5 и 3 соответственно. Выведите результат на экран.

**Задача 2. Автобус или метро?**

Для расчёта эффективности потраченных денег на метро и автобус Никита использует следующую формулу:



В общем, не спрашивайте. Он написал программу для подсчёта формулы, но она почему-то не работает.

bus = 5

metro = 3

result = (6 + 10 - bus \* 2) / (bus + 1) + 132 / 2 + metro

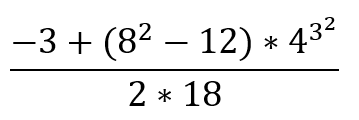
print(result)

Скопируйте программу в редактор, исправьте выражение и убедитесь в правильности её работы. Правильный ответ: 26.607142857142858

**Задача 3. Сложные степени**

Говорят, если в 22:31:49 посчитать на Python значение определённого выражения, то спустя год станешь очень хорошим программистом. Давайте проверим это утверждение.

1. Создайте переменную и запишите в неё следующее математическое выражение:



2. Затем выведите значение переменной на экран. Ответ должен быть равен 378652.3611111111.

3. Будьте внимательны со скобками, особенно с делением числителя на знаменатель.

**Подсказка**

Приоритет операций в python не отличается от приоритета, принятого в алгебре.

Операции выполняются в следующем порядке

Операции в скобках (a + b)

Операции возведения в степень a\*\*b

Операции умножения и деления a \* b

Операции сложения и вычитания a + b - c

Операции унарных плюсов и минусов -4

Приоритеты всех операций вы сможете найти по ссылке [Приоритет операций](https://docs.python.org/3/reference/expressions.html#operator-precedence)

При равенстве приоритетов операции выполняются слева направо.

При вычислении выражений с числителем и знаменателем, записанных в виде дробей будьте внимательны к тому, что весь числитель нужно разделить на весь знаменатель.

Для этого можно отдельно вычислить числитель и отдельно знаменатель, а потом разделить числитель на знаменатель, либо поместить числитель и знаменатель в скобки.

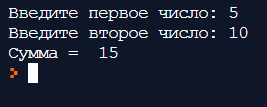
## 3.4 Ввод числа с клавиатуры

**Задача 1. Координаты точки**

Лена пишет программу для построения графиков функций, чтобы можно было анализировать статистику посещений её сайта. Перед тем как построить график функции, Лена сначала расставляет на нём точки. У каждой точки есть координаты: X и Y. Лене также нужно найти сумму этих координат, чтобы убедиться в их правильном расположении.

Реализуйте программу, которую мы разбирали в уроке: она запрашивает у пользователя два числа и выводит их сумму на экран.

Используйте функцию int() для преобразования входящих данных из текста в число. Для этого внутри скобок функции int вставьте команду input. Результат должен быть таким:

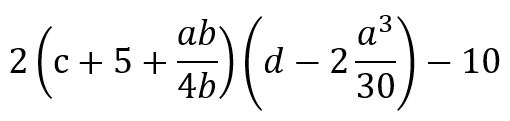


**Задача 2. Отрезок**

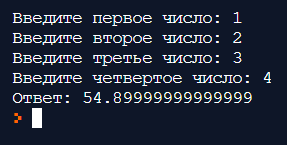
Доверчивой Лене необходимо построить отрезок, соединяющий две точки на графике, и найти его длину по довольно необычной формуле, которую она нашла в интернете. Для нахождения длины отрезка нужны координаты как первой, так и второй точки.

Продолжая программу прошлой задачи, добавьте ввод третьего и четвёртого числа, вычислите значение выражения и выведите результат на экран.

Выражение:



Результат:



**Задача 3. Исправление ошибки**

Секретарь заполняла данные по категории сотрудников для расчёта множителя к зарплате и, не проверив правильность записи, случайно нажала «Подтвердить», после чего данные со значениями заблокировались. Однако осталась возможность редактировать формулу.

Дана программа:

a = '2'

b = '5'

c = '3'

num = 6 \*\* a + (7 - b) \* c

print(num)

Скопируйте её в редактор и исправьте в ней четвёртую строку, используя только функцию int().

Запустите и проверьте работу программы. Результат должен быть равен 42.

## 3.5 Деление нацело и с остатком

**Задача 1. Яблоки**

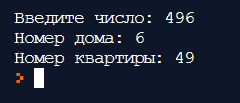
Транспортная компания ООО «ФруктыТрансСервис» занимается логистикой и грузоперевозками фруктов. Программисту Владимиру дали задачу написать программное обеспечение, которое позволит понять, сколько фруктов можно загрузить и сколько останется на складе.

Напишите программу, которая решает задачу из урока: у транспортной компании есть 41 тонна яблок, которые нужно разместить по ящикам. Каждый ящик вмещает в себя три тонны. Необходимо выяснить, сколько ящиков мы сможем заполнить и сколько яблок останется. Оба ответа нужно вывести на экран.

**Задача 2. Последняя цифра**

Избирательный участок включает в свою базу данных номера квартир и домов и, чтобы не хранить два числа отдельно, оба номера записывает одним числом вместе, где последняя цифра — это номер дома, а всё, что перед ней, — номер квартиры.

Реализуйте программу, которая запрашивает число у пользователя, а затем выводит на экран номер дома и номер квартиры. Пример:



## 3.7 Практическая работа

**Задача 1. Язык математики**

Что нужно сделать

В первый же день на сайте отвалилась формула по расчёту рекламной метрики, и только Вася может её поправить. Часть программы с вводными данными представлена ниже, отдельно записана формула на математическом языке.

Дана программа:

a = 8

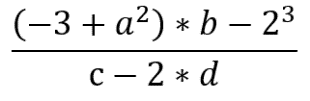
b = 10

c = 12

d = 18

Продолжите программу: переведите выражение с математического языка на язык Python, запишите его в переменную res и выведите результат.

Выражение:



**Советы и рекомендации**

Порядок действий не всегда очевиден, но при помощи скобок его можно изменять и упрощать.

**Что оценивается**

Результат вычисления корректен.

Правильно употреблены пробелы после запятых и при бинарных операциях.

**Задача 2. Финансовый отчёт**

Васе пришло очередное задание — автоматизация финансовой отчётности. Звучит сложно, а на деле нужно просто написать код, который будет считать нужные для отчёта вычисления автоматически. Вычисления, которые нужно реализовать в программе: сумму дохода первых двух кварталов поделить на сумму последних двух кварталов, чтобы понять динамику роста или падения дохода.

**Алгоритм действий в программе:**

Запросить у пользователя четыре числа.

Отдельно сложить два первых и два вторых.

Разделить первую сумму на вторую.

Вывести результат на экран.

**Что оценивается**

В input содержится корректное приглашение для ввода.

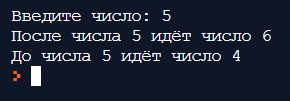
Результат вычислений корректен.

Переменные имеют значащие имена: не только a, b, c, d (видео 2.3).

Есть пробелы после запятых и при бинарных операциях.

**Задача 3. Следующее и предыдущее числа**

Напишите программу, которая получает от пользователя число и выводит на экран два ответа — следующее и предыдущее числа. Результат:



**Что оценивается**

В input содержится корректное приглашение для ввода.

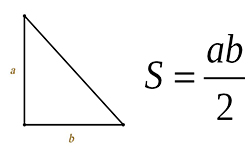
Вывод соответствует заданию.

Нет простых print(a), print(a − 1).

Есть пробелы после запятых в print.

**Задача 4. Площадь треугольника**

Напишите программу, которая запрашивает у пользователя длины двух катетов в прямоугольном треугольнике и выводит его площадь.



**Советы и рекомендации**

Не стоит применять целочисленное деление, это не совсем корректно.

Обратите внимание на написание переменной S: её нужно писать как s, со строчной.

**Что оценивается**

Вычисления корректны, применены правильные операции.

В input содержится корректное приглашение для ввода.

Есть пробелы после запятых и при бинарных операциях.

**Задача 5. Часы**

Напишите программу, которая получает на вход число n (количество минут), затем считает, сколько это будет в часах и сколько минут останется, и выводит на экран эти два результата.

**Советы и рекомендации**

В реальной практике приходится вникать в контекст задачи (даже если это совсем не ваша область знаний), чтобы понять, как решить её при помощи кода. Ответы на следующие вопросы станут подсказками к шагам решения:

Сколько градусов в полном круге? Сколько часов в полном обороте часовой стрелки?

Сколько оборотов делает минутная стрелка за час? Сколько градусов проходит минутная стрелка за полный час? За половину часа?

Какую часть от полного количества часов на циферблате проходит часовая стрелка за час?

Какую часть круга проходит часовая стрелка за час?

**Что оценивается**

Результат вычислений корректен.

В input содержится корректное приглашение для ввода.

Переменные имеют значащие названия.

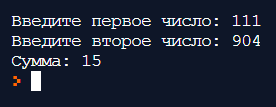
Есть пробелы после запятых и при бинарных операциях.

**Задача 6. Проверяем бухгалтера**

Невнимательный бухгалтер Антон складывает числа быстро, но иногда забывает о двух последних разрядах. Чтобы помочь Антону, напишите программу, которая бы складывала только два последних разряда.

Реализуйте программу, которая запрашивает два числа у пользователя. После этого у каждого числа возьмите две последние цифры. Получившиеся два числа сложите и выведите на экран.

Пример:



**Что оценивается**

Результат вычислений корректен.

В input содержится корректное приглашение для ввода.

Переменные имеют значащие названия.

Есть пробелы после запятых и при бинарных операциях.

**Задача 7. Режем число на части**

Реализуйте программу, которая получает на вход четырёхзначное число и выводит на экран каждую его цифру отдельно (в одну строчку либо каждую цифру в новой строчке). Само число при этом изменять нельзя, то есть нужно обойтись без переприсваивания. Однако можно использовать сколько угодно переменных.

**Что оценивается**

Результат вычислений корректен.

В input содержится корректное приглашение для ввода.

Есть пробелы после запятых и при бинарных операциях.

**Задача 8. Поменять местами: не всё так просто! (необязательная, повышенной сложности)**

Вы уже умеете менять местами строковые переменные и знаете, что в переменных кроме строк можно хранить и числа. Напишите программу, которая меняла бы значения двух переменных местами, но без использования третьей переменной и синтаксического сахара, который мы разбирали, а именно: без конструкции a, b = b, a. В переменные будут вводиться только числа.

a = int(input('Введите первое число: '))

b = int(input('Введите второе число: '))

print(a, b)

# стереть эту строчку и вставить свой код здесь

print(a, b)

Изменять, удалять, менять местами первую, вторую, третью и последнюю строчки нельзя. В четвёртую строку можно вставлять сколько угодно кода, не трогая последний print.

**Что оценивается**

Результат вычислений корректен.

Не используются swap и третья переменная.

Не изменён вывод.

В input содержится корректное приглашение для ввода.

**Что оценивается (общее)**

Нет пустых или неинформативных input (видео 2.3).

Нет неинформативных имён переменных.

Есть пробелы после запятых и при бинарных операциях.

**Советы и рекомендации**

Обратите внимание на приоритет арифметических операций в [PEP8](https://docs.python.org/3/reference/expressions.html#operator-precedence).

Если после решения задач у вас остаются силы, можете заняться совершенствованием стиля кода. Стиль не влияет на эффективность выполнения кода, но помогает другим программистам лучше понимать ваш код, что делает вас более ценным сотрудником.

# Модуль 4

## 4.2 Условный оператор

| **Операции сравнения** | | **Примеры использования** |
| --- | --- | --- |
| > | Больше | if a > b: |
| < | Меньше | if a < b: |
| >= | Больше либо равно | if a >= b: |
| <= | Меньше либо равно | if a <= b: |
| == | Равно | if a == b: |
| != | Не равно | if a != b: |

**Задача 1. Сбой**

На предприятии произошло ЧП: что-то случилось с программным обеспечением, и производство встало. Когда стали разбираться, нашли битый кусок программы:

a = 6

b = ?

c = 0

if ? < ?

c = a \* b

print(c)

По задумке, здесь проверялись переменные а и b и в конце должен был выводиться ответ 12.

Скопируйте программу в редактор, исправьте ошибки и вставьте необходимые значения вместо знаков вопроса. Проверьте работу программы.

**Задача 2. Курс от Skillbox**

Повторим задачу, разобранную в уроке, и дополним её одной строчкой.

Пользователь хочет купить курс и вводит состояние своего банковского счёта для проверки.

Курс стоит 75 000 рублей. Если на счету хватает денег (больше или равно), то со счёта списываются 75 000 рублей, и программа выводит сообщение об успешной покупке.

В конце обязательно выводится сообщение «Хорошего дня!», вне зависимости от результата проверки.

Пример:

Сколько денег на счету? 90 000

Курс успешно приобретён.

Хорошего дня!

**Задача 3. Угадай число**

Отец пообещал сыну отпустить его гулять, если тот с первого раза угадает число от 1 до 10, которое загадал папа. В противном случае сыну придётся весь день делать уроки.

Напишите программу из урока, которая запрашивает число у пользователя (у сына), затем сравнивает его с другим числом (которое отец загадал), и если они равны, то выводит сообщение «Угадал». В конце вывести фразу: «Конец игры».

Пример 1:

Какое число я загадал? 5

Угадал!

Конец игры

Пример 2:

Какое число я загадал? 6

Конец игры

Попробуйте решить задачу сначала с помощью одного знака сравнения (==), а затем с помощью другого (!=). Во втором случае сообщение измените на «Не угадал».

## 4.3 Полная форма условного оператора if

**Задача 1. Курс от Skillbox-2**

Напишите программу для примера, разобранного в уроке.

Пользователь покупает курс стоимостью 75 000 рублей.

Если денег на счету достаточно, нужно списать деньги и вывести сообщение: «Курс успешно приобретён», — а иначе вывести: «Не хватает денег на счёте».

Не забудьте пожелать «Хорошего дня!» в любом случае. Мы же вежливые продавцы.

Пример:

Сколько денег на счету? 5000

Не хватает денег на счету.

Хорошего дня!

**Задача 2. Разминка для мозгов**

Напишите программу, которая проверяет то, как вы умеете складывать два числа в уме. Программа получает на вход два числа и предполагаемую сумму и должна выводить два разных сообщения — на верный и неверный ответ пользователя. В последнем случае надо показывать правильный результат.

Пример:

Введите первое число: 3

Введите второе число: 10

Сумма этих чисел: 13

Ответ верный!

Пример 2:

Введите первое число: 3

Введите второе число: 10

Сумма этих чисел: 14

Ответ неверный!

**Задача 3. Угадай число 2**

На удивление, отец и сын частенько стали играть в игру «Угадай число», и поэтому папа захотел немного усовершенствовать свою программу, чтобы на экран всегда выводилось нужное сообщение.

Напишите программу, которая запрашивает число у пользователя, сравнивает его с другим числом и выводит соответствующее сообщение: «Угадал», — если они равны, и: «Не угадал», — если не равны. В конце выводите фразу: «Конец игры».

Пример 1:

Какое число я загадал? 5

Угадал!

Конец игры

Пример 2:

Какое число я загадал? 6

Не угадал!

Конец игры

Попробуйте решить задачу сначала с помощью одного знака сравнения (==), а затем с помощью другого (!=).

## 4.5 Практическая работа

**Задача 1. Датчик погоды**

За окном квартиры стоит датчик погоды, который определяет, идёт ли дождь. Если идёт, датчик оповещает сообщением: «Пошёл дождь. Возьмите зонтик!»

Напишите программу, которая получает на вход число 0 или 1. Единица означает, что дождь идёт. В таком случае выводите на экран сообщение: «Пошёл дождь. Возьмите зонтик!». Ноль означает, что дождя нет, в этом случае надо вывести сообщение «Дождя нет!»

Пример 1:

На улице идёт дождь? 1

Пошёл дождь. Возьмите зонтик!

Пример 2:

На улице идёт дождь? 0

Дождя нет!

**Что оценивается**

результат вывода корректен;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 2. Поступление**

Вася работает разработчиком, и его компания создаёт сайт для образовательной компании. Заказчики попросили реализовать калькулятор баллов ЕГЭ в помощь поступающим. Эту задачу отдали Васе.

Напишите программу, которая запрашивает у пользователя результаты ЕГЭ по трём экзаменам, и проверяет, поступил ли он на бюджет. Проходной балл равняется 270. Выведите соответствующее сообщение.

Пример 1:

Введите количество баллов по русскому языку: 90

Введите количество баллов по математике: 90

Введите количество баллов по информатике: 90

Поздравляю, ты поступил на бюджет!

Пример 2:

Введите количество баллов по русскому языку: 100

Введите количество баллов по математике: 50

Введите количество баллов по информатике: 70

К сожалению, ты не прошёл на бюджет.

**Что оценивается**

input содержит корректное приглашение для ввода;

результат вывода корректен;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 3. Следим за расписанием**

После выполненной задачи Вася устал и понял, что весь следующий день ему придётся восстанавливать силы. Вася решил, что работать он будет только в чётные дни и написал небольшую программу, которая поможет ему не пропустить рабочий день.

Напишите программу, которая проверяет, чётное ли число ввёл пользователь, и выводит соответствующее сообщение.

Подсказка: для проверки чётности используйте оператор %.

**Что оценивается**

input содержит корректное приглашение для ввода;

результат вывода корректен;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 4. Калькулятор скидки**

Васин друг переехал в новую квартиру, и ему нужно купить три стула в разные комнаты. Цены на стулья разные, а в некоторых магазинах есть скидки. Друг хочет заказать у Васи калькулятор скидки, чтобы проще ориентироваться в ценах.

Напишите программу, которая запрашивает три стоимости товара и вычисляет сумму чека. Если сумма чека превышает 10 000 руб., нужно вычесть из этой суммы скидку 10% (умножить на 10, разделить на 100). Итоговая сумма должна выводиться на экран.

**Что оценивается**

корректность вычислений и применения операций;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 5. Модуль числа**

Создайте программу, которая найдёт модуль очередного числа x. Если число x отрицательное, то она должна перевести его в положительное, а в конце на экране должен быть модуль введённого числа.

Пример:

Ввели 5, ответ 5

Ввели −7, ответ 7

Подсказка: в некоторых случаях достаточно переприсвоить переменную со знаком минус.

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру (не выведено одно число без описания);

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 6. Игра в кубики**

Вася понимает, как важен отдых после тяжёлого рабочего дня, поэтому часто ходит в местный бар с друзьями. Владелец бара любит азартные игры и иногда предлагает посетителям с ним сыграть. Вася избегает азартные игры, но предложил владельцу купить у него программу для расчёта результатов игры, которую можно выводить на экраны бара.

Напишите программу: на вход в неё подаётся два числа. Если первое число больше или равно второму, то нужно вывести на экран их разность и отдельной строкой: «Игрок платит». В противном случае, вывести их сумму и отдельной строкой: «Владелец платит». Последней строкой на экран должна быть выведена фраза: «Игра окончена».

Пример:

Кубик Кости: 3

Кубик владельца: 4

Сумма: 7

Владелец платит

Игра окончена

**Рекомендация**

Обратите внимание на разный выигрыш, который получит игрок при разных исходах.

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

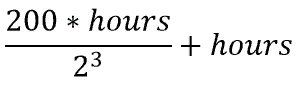
формат вывода соответствует примеру (не выведено одно число без описания);

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 7. Хватит ли зарплаты**

Вася набрался опыта и решил поискать вакансию с большей зарплатой. Его привлекла вакансия со странной формулой для расчёта заработной платы:



Он хочет понять, сколько часов нужно отработать, чтобы хватило на погашение кредита и еду.

Напишите программу, которая запрашивает у пользователя три числа: количество отработанных часов, остаток по кредиту и количество денег на еду. После этого рассчитывается зарплата по формуле. Если зарплата больше или равна сумме, которая требуется на кредит и еду, то выводится сообщение: «Часов хватает. Можно отдохнуть». В противном случае: «Часов не хватает. Придётся работать больше!»

Пример:

Введите отработанные часы: 80

Введите остаток по кредиту: 1000

Введите траты на еду: 5000

Часов не хватает. Придётся работать больше!

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру (не выведено одно число без описания);

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

пробелы после запятых, пробелы при бинарных операциях.

**Задача 8. Максимальное число (по желанию)**

Пользователь вводит три числа. Напишите программу, которая выводит на экран максимальное из этих трёх чисел (все числа разные). Используйте дополнительные переменные, если нужно.

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

правильное употребление пробелов после запятых и при бинарных операциях.

**Что оценивается во всей практической работе**

избегание пустых или неинформативных input (видео 2.3);

избегание неинформативных имён переменных;

избегание использования в качестве имён встроенных функций [список встроенных функций](https://docs.python.org/3.7/library/functions.html) — официальная документация);

правильное употребление пробелов после запятых и при бинарных операциях.

**Советы и рекомендации**

Обратите внимание на:

приоритет арифметических операций [PEP8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence) (мы вводим and, or);

расширенное описание сложения чисел [PEP 3141](https://www.python.org/dev/peps/pep-3141/#implementing-the-arithmetic-operations)

успеваемость: если вы легко справляетесь с заданиями, можете потратить время, чтобы улучшить стиль.

# Модуль 5

## 5.2 Вложенные условия

**Задача 1. Координаты**

Мы тестируем 2D-игру, где нужно управлять подводной лодкой. У лодки есть координаты в пространстве — X (икс) и Y (игрек). X — это движение вперёд-назад, а Y — вверх-вниз. Соответственно, во время движения лодки меняются и её координаты. Во время тестирования игры нам необходимо сравнивать эти координаты и выводить на экран нужное сообщение, в том числе если они равны.

Вводятся две координаты — X и Y. С помощью трёх последовательных проверок сравните обе координаты и выведите соответствующее сообщение.

Пример:

Введите икс: 5

Введите игрек: 6

X меньше Y

Пример 2:

Введите икс: 3

Введите игрек: 3

X равен Y

**Задача 2. Скидки!**

Напишите программу для примера, разобранного в уроке. Пользователь покупает курс стоимостью 75 000 рублей. Если денег на счету достаточно, то нужно:

Списать со счёта деньги.

Проверить баланс счёта. Если там меньше 5000 рублей, то зачислить на счёт 1000 рублей и вывести сообщение: «Сделана скидка».

Вывести сообщение: «Курс успешно приобретён».

А иначе вывести: «Не хватает денег на счету». Также в конце вывести остаток счёта и сообщение: «Хорошего дня!»

Пример:

Сколько денег на счету? 78500

Курс успешно приобретён

Сделана скидка

Остаток на счету: 4500

Хорошего дня!

**Задача 3**

Мама дала Маше денег и отправила её в магазин за сыром. А ещё сказала: «Если останутся деньги, то можешь купить себе мороженое. Если денег на сыр не хватит, то денег маловато — а значит, и мороженого не будет».

Сделайте программу, которая получает на вход количество денег. Сыр стоит 60 рублей, мороженое — 20 рублей. Если денег на сыр хватает (больше либо равно), то:

Выводите сообщение: «На сыр денег хватило», — и вычитайте стоимость сыра из кошелька.

Если оставшихся денег хватает на мороженое, то выводите: «И на мороженое тоже!». Иначе выводите: «Денег маловато».

## 5.3 Цепочки условий if-elif-else

**Задача 1. Координаты**

Вернёмся к задаче про подводную лодку и её координаты. Теперь мы знаем, что вместо трёх отдельных проверок можно использовать оператор elif — «»иначе-если». То есть если одна координата больше другой, то будет одно сообщение, иначе если одна меньше другой, то…

Напишите программу, которая сравнивает координаты X (икс) и Y (игрек) и выводит соответствующий результат. Для этого используйте оператор elif.

**Задача 2. Прогрессивный налог**

В некоторых странах действует так называемая прогрессивная шкала налогообложения: чем больше ты зарабатываешь, тем больший налог платишь. Нужно написать программу, которая будет рассчитывать сумму налога исходя из прибыли. Если прибыль до 10 000 — ставка налога равна 13%, начиная с 10 000 и до 50 000 — 20%. А начиная с 50 000 — 30%. А также нужно добавить «проверку на дурака»: если ввели число меньше нуля, то вывести сообщение: «Ошибка: доход не может быть отрицательным».

**Задача 3. Фальшивомонетчики**

Пете дали три монетки и сказали, что если он сможет на весах без гирь определить, какая из них фальшивая (более лёгкая), то сможет забрать их все. А Петя в одной из книжек прочитал, что для этого достаточно одного взвешивания: если две монеты весят одинаково, то фальшивая третья, а иначе фальшивая та, которая легче на весах. Напишите программу, которая принимает на вход вес трёх монет (две одинаковые, третья меньше) и определяет, какая из них легче.

## 5.4 Логические операторы and и or

**Задача 1. Покупка велосипеда**

Напишите программу, которую мы разбирали в рамках теории. Нашему ребёнку нужен новый хороший велосипед. Правда, никто из нас в них не разбирается, всё что нам нужно — чтобы велосипед не был устаревшим и чтоб скоростей на нём было побольше, а сколько он стоит — пока неважно. Чтобы не искать велосипед на сайте вручную, мы хотим написать программу, которая будет проверять каждый велосипед на нужный нам год выпуска и на количество скоростей.

Используя один из логических операторов (and, or), напишите программу из урока, которая запрашивает год выпуска велосипеда и количество скоростей на нём и выводит на экран сообщение о том, подходит этот велик или нет. Год выпуска — не старше 2018-го, количество скоростей — не менее 24.

**Задача 2. Как поступить?**

Илья хочет в лучший вуз страны, а для этого нужно не только хорошо сдать экзамены (балл должен быть больше 280), но и иметь золотую медаль.

Напишите программу, которая запрашивает у пользователя два числа: результат экзаменов и наличие золотой медали (0 — нет медали, 1 — медаль есть), а затем проверяет, поступил ли Илья в вуз. Выведите соответствующее сообщение.

Пример:

Сколько баллов набрал? 290

Есть медаль? 1

Поздравляем! Ты поступил!

Пример 2:

Сколько баллов набрал? 269

Есть медаль? 1

К сожалению, ты не прошёл в наш университет.

**Задача 3. Бактерии живут комфортно**

Биолог Арсений изучает микробы и их поведение при разных температурных нагрузках. Он помещает их в специальную среду, где температура скачет в промежутке от 0 до 100 градусов. Если же температура в среде выходит за рамки промежутка, то выводится предупреждение.

Напишите программу, которая запрашивает у пользователя температуру, и, если она меньше нуля или больше 100, то выводится сообщение об опасности.

## 5.6 Практическая работа

**Задача 1. Калькулятор опыта**

В свободное время Вася любит играть в компьютерные игры. Однажды у него появилась классная идея для сюжета игры. Чтобы воплотить её в жизнь, он начал изучать геймдизайн. Создавать игру он начал с главного героя и его системы прокачки.

Напишите программу, которая определяет уровень персонажа в компьютерной игре. Пользователь вводит число «очков опыта», а программа вычисляет уровень. Новый уровень даётся при достижении 1000, 2500 и 5000 «очков опыта». Начальный уровень равен единице.

Пример:

Введите количество опыта: 6000

Ваш уровень: 4

Пример 2:

Введите количество опыта: 2000

Ваш уровень: 2

**Советы и рекомендации**

По возможности уделите внимание сокращению кода. Постарайтесь не проверять условия, которые уже проверены: если вы проверили условие condition, то не следует проверять not condition.

**Что оценивается**

результат вывода корректен, особое внимание уделено границам диапазонов;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях;

правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Задача 2. Функция**

Учитель математики придумывает каждому ученику отдельные функции, которые нужно отобразить на графике и посчитать. Ещё учитель разбирается в программировании, и чтобы не считать вручную эти функции, он написал программу, которая делает работу за него.

Напишите программу, которая получает от пользователя число X и вычисляет значение функции Y по следующей схеме:

y= {x −12, x>0, 5, x=0 x2, x<0

Напомним, как это работает:

для X > 0, Y = X − 12

для X = 0, Y = 5

для X < 0, Y = X2

Пример:

Введите икс: 0

Игрек равен 5

Пример 2:

Введите икс: −6

Игрек равен 36

**Советы и рекомендации**

По возможности уделите внимание сокращению кода. Избегайте проверки условий, которые уже проверены: если вы проверили условие condition, то не следует проверять not condition.

**Что оценивается**

результат вывода корректен;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых при бинарных и логических операциях;

правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Задача 3. Поступление**

В университете на факультет кибернетики очень серьёзный конкурс — поступают только сильнейшие, первые десять человек из списка. Потом среди поступивших определяется, кто будет получать стипендию. Для стипендии общий балл при поступлении должен быть не менее 290.

Напишите программу, которая получает на вход место студента в списке и его балл, а затем выводит соответствующие сообщения о поступлении и получении стипендии.

Пример 1:

Введите место в списке поступающих: 3

Введите количество баллов за экзамены: 295

Поздравляем, вы поступили!

Бонусом вам будет начисляться стипендия.

Пример 2:

Введите место в списке поступающих: 3

Введите количество баллов за экзамены: 270

Поздравляем, вы поступили!

Но вам не хватило баллов для стипендии.

Пример 3:

Введите место в списке поступающих: 11

К сожалению, вы не поступили.

**Что оценивается**

результат вывода корректен и соответствует примеру;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых при бинарных и логических операциях;

правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Задача 4. Опять двойка**

Папа-программист настолько обленился, что вместо того, чтобы спросить у сына, какую оценку тот получил в школе, он написал программу:

rating = int(input('Что получил по математике? '))

if rating == 2:

print('Плохо. Марш учиться!')

if rating == 3:

print('Плохо. Марш учиться!')

if rating == 4:

print('Молодец! Можешь отдохнуть.')

if rating == 5:

print('Молодец! Можешь отдохнуть.')

Сын посмотрел на код программы и понял, что её можно улучшить. Он даже рассказал папе, как это сделать, за что получил безграничное уважение отца.

Скопируйте программу в редактор и оптимизируйте:

При плохой оценке (2 или 3) выводится сообщение: «Плохо. Марш учиться!»

При хорошей оценке (4 или 5) выводится сообщение: «Молодец! Можешь отдохнуть».

В программе не должно быть повторяющихся строк.

**Что оценивается**

В программе отсутствуют повторяющиеся строки.

**Задача 5. Вася хочет выигрывать**

Вася вдохновился фильмом «Двадцать одно» и решил изучить математику игровых автоматов. Для анализа данных ему нужна информация о том, как часто в автомате выпадает три или две одинаковых картинки. Для сбора данных нужна программа, проверяющая это равенство.

Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 3 (если все совпадают), 2 (если два совпадают) или 0 (если все числа разные).

**Советы и рекомендации**

По возможности уделите внимание сокращению кода. Избегайте проверки условий, которые уже проверены: если вы проверили условие condition, то не следует проверять not condition.

**Что оценивается**

input содержит корректное приглашение для ввода;

в выводе может быть указано только число, но хорошим стилем считается описание вывода;

правильное употребление пробелов после запятых при бинарных и логических операциях;

правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Задача 6. Новоселье**

Семья из трёх человек устала тесниться в однушке и наконец решила переехать. При обсуждении, какую купить квартиру, исходя из предпочтений и семейного бюджета, они остановились на двух вариантах:

Выбрать квартиру попросторнее (не менее 100 м2), но стоимостью не более 10 млн.

Немного расширить круг поиска, то есть выбрать квартиру поменьше (от 80 м2), но и стоимостью не более 7 млн.

Напишите программу, которая получает на вход стоимость квартиры и её площадь и выводит сообщение, подходит ли она.

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

в выводе программы указано, подходит ли данный параметр;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 7. Почта**

Почтовое отделение открывается в 08:00 и закрывается в 22:00. С 14:00 до 15:00 все сотрудники уходят на обед, а в 10:00 и 18:00 приезжают машины с посылками, и все сотрудники на два часа заняты их разгрузкой. Во время обеда и разгрузки машин посылки никто не выдаёт.

Напишите программу, которая получает на вход время в часах — число от 0 до 23 — и пишет, можно ли в этот час получить посылку. Используйте только один условный оператор if-else, без elif и прочих. Решите задание двумя способами:

При выполнении условия выводится сообщение: «Можно получить посылку».

При выполнении условия выводится сообщение: «Посылку получить нельзя».

Советы и рекомендации

Обратите внимание на количество условий и постарайтесь сократить их. Не используйте перечисление конкретных часов вида a == 1 and a == 2...

**Что оценивается**

результат вычислений корректен;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях.

**Советы и рекомендации**

Обратите внимание:

стиль оформления программ на языке Python [PEP8](https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html);  
оригинальный документ [PEP8](https://www.python.org/dev/peps/pep-0008/) (на английском);

приоритет арифметических операций [PEP8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence) (вводим and, or); [список встроенных функций](https://docs.python.org/3.7/library/functions.html) (официальная документация).

Что оценивается во всей практической работе

правильно оформленный input, без пустого приветствия для ввода (видео 2.3);

переменные имеют значащие имена, а не только a, b, c, d (видео 2.3);

правильное употребление пробелов после запятых и при бинарных операциях;

отсутствие пробелов после имён функций и перед скобками: “print ()”,“input ()” — неверно, “print()” — верно;

правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня;

переменные имеют корректные названия и в качестве имён не используются имена

встроенных функций ([список встроенных функций](https://docs.python.org/3.7/library/functions.html) — официальная документация).

# Модуль 6

## 6.2 Оператор while

**Задача 1. Циклы — это сложно?**

Даша начала проходить циклы. Она написала программу, которая просто должна считать сумму чисел до тех пор, пока мы не ввели 0, но что-то пошло не так.

number = int(input(‘Введите число: ’))

While number <= 0:

summ += number

number = int(input(‘Введите число: ’))

print(Summ)

Скопируйте программу в редактор, исправьте её и убедитесь в правильности работы программы.

**Задача 2. Введите пароль**

Когда мы включаем ноутбук, он просит нас ввести пароль. Если пароль оказался неверным, нам сообщают об этом и снова просят ввести пароль. И так до тех пор, пока мы не введём правильный пароль, — тогда ноутбук даст доступ в учётную запись, и программа завершится.

Напишите программу, которая запрашивает у пользователя пароль до тех пор, пока он не введёт верный (верным будет считаться пароль 235).

Пример:

Введите пароль: 100000

Неверный пароль!

Попробуйте ещё раз: 500

Неверный пароль!

Попробуйте ещё раз: 235

Пароль верный! Добро пожаловать.

**Задача 3. Накопления**

Кирилл копит себе на хорошую новенькую машину и для этого он каждый раз откладывает в копилку определённую сумму с зарплаты.

Напишите программу, которая спрашивает у пользователя «Сколько отложить денег?» до тех пор, пока сумма в копилке не превысит или не станет равна 500 000 рублей.

## 6.3 Прерывание цикла, оператор break

**Задача 1. Бегать — это полезно**

Представим, что у нас далёкое будущее и мы можем заниматься спортом на планетах со странными перепадами температур. Спортсмен бегает по стадиону до тех пор, пока температура на улице больше 15 градусов. Он пробегает 20 метров, затем температура падает на два градуса, и, если уже в этот момент она стала меньше либо равна 15 градусам, спорт сразу заканчивается. Если же всё в порядке, то он проходит пешком ещё 10 метров. Затем всё это повторяется.

Напишите программу, которая спрашивает у пользователя, сколько на улице градусов, и, исходя из погоды, считает количество пройденных по стадиону метров и выводит ответ на экран.

**Задача 2. Расшифровка кода**

Нам нужно расшифровать определённый код в виде одного большого числа. Для этого нужно посчитать сумму цифр справа налево. Если мы встречаем в числе цифру 5, то выводим сообщение «Обнаружен разрыв» и заканчиваем считать сумму. В конце программы на экран выводится сумма тех цифр, которые мы взяли.

**Задача 3. Ставки приняты, ставок больше нет**

Костя опять за старое! Только теперь он играет в кубики более надёжно, то есть на какую-то фиксированную сумму. И при этом пока что постоянно выигрывает! Однако по правилам это не мешает ему проиграть сразу всё.

Напишите программу, которая запрашивает у пользователя начальное количество денег и, пока оно меньше 10 000, запрашивает число, которое выпало на кубике (от 1 до 6). Если на кубике выпало 3, то выводится сообщение «Вы проиграли всё!», и деньги обнуляются. Если выпало другое число, к сумме прибавляется 500.

Пример:

Введите стартовую сумму: 5000

Сколько выпало на кубике? 4

Выиграли 500 рублей!

Сколько выпало на кубике? 5

Выиграли 500 рублей!

Сколько выпало на кубике? 3

Вы проиграли всё!

Игра закончена.

Итого осталось: 0 рублей

Пример 2:

Введите стартовую сумму: 9700

Сколько выпало на кубике? 4

Выиграли 500 рублей!

Игра закончена.

Итого осталось: 10200 рублей

## 6.4 Бесконечный цикл. Логический тип данных

**Задача 1. Неправильный таймер**

Петя писал таймер для телефона, но у него что-то пошло не так.

count = 0

While count <= 10

if count == 0:

print('Время вышло!')

else:

print(count)

count -= 1

Скопируйте программу в редактор, исправьте ошибки и убедитесь, что на экран выводятся числа с 10 до 0 и сообщение «Время вышло!».

**Задача 2. Тестируем приложение**

Напишите программу, которая имитирует работу с приложением: программа спрашивает у пользователя «Продолжаем работать? 1/0: » до тех пор, пока пользователь не введёт 0, — после этого выводится сообщение: «Приложение закрывается…». В конце программы также выводится сообщение: «Работа завершена». Для создания бесконечного цикла используйте while True.

**Задача 3. Вирус**

Дима написал программу-вирус специально для того, чтобы проучить своего друга-должника, который никак не хочет возвращать скейтборд. Программа не даёт работать за компьютером и постоянно выводит на экран сообщение «Компьютер заблокирован. Вернёшь скейт — скажу код разблокировки!». Как только вводится правильный код, вирус удаляется. Напишите такую же программу, которую написал Дима.

Пример:

Компьютер заблокирован. Вернёшь скейт — скажу код разблокировки!

Введите код: 1005

Компьютер заблокирован. Вернёшь скейт — скажу код разблокировки!

Введите код: 7777

Компьютер заблокирован. Вернёшь скейт — скажу код разблокировки!

Введите код: 0550

Код верный, завершаю работу…

## 6.5 Цикл while со счётчиком

**Задача 1. Надоедливый заказчик**

Нашему заказчику нужно, чтобы фраза «Я — программист!» выводилась на экран столько раз, сколько он сам этого захочет.

Напишите программу, которая запрашивает число — количество строчек с фразой «Я — программист!» — и столько же раз выводит на экран эту фразу. Для решения задачи используйте переменную-счётчик, которая увеличивается на единицу до тех пор, пока не будет введено нужное количество строчек.

**Задача 2. Напоминалка**

У Евгения много работы, а ещё он очень забывчивый. Иногда он забывает о какой-нибудь важной встрече, и ему приходится выслушивать критику от начальства. Напишите для него программу-напоминалку. В самом начале программа спрашивает, сколько раз ему напомнить, а затем нужное количество раз выводит: «Вы хотели не забыть о чём-то».

**Задача 3. Рыбалка**

Наши прекрасные родственники удачно сходили на рыбалку. Настолько, что ходили мешком перетаскивать рыбу с берега в машину целых n раз. Каждый мешок они взвешивали на электронных весах (все мешки весили по-разному). Напишите программу для весов, которая считает суммарный вес мешков и выводит его на экран.

## 6.7 Практическая работа

**Задача 1. Кубы чисел**

В один из вечеров к Васе домой пришёл племянник и пожаловался на сложности с уроками математики: у него никак не получалось разобраться со степенями чисел. Вася решил помочь племяннику и написать программу, которая позволит наглядно увидеть возведение чисел в третью степень.

Напишите программу, которая возводит в третью степень каждое число от 1 до N и выводит результат на экран.

Что оценивается

Результат вывода корректен, решение использует цикл while.

Input содержит корректное приглашение для ввода.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 2. Коллекторы**

Напишите робота для коллекторов. В самом начале он спрашивает имя должника и сумму долга, а затем начинает требовать у него погашения до тех пор, пока должник не введёт нужную сумму (равную сумме долга или больше). После погашения долга робот сообщает об этом пользователю и благодарит его.

Пример

Василий, ваша задолженность составляет 100 рублей.

Сколько рублей вы внесёте прямо сейчас, чтобы её погасить? 50

Маловато, Василий. Давайте ещё раз.

Сколько рублей вы внесёте прямо сейчас, чтобы её погасить? 90

Маловато, Василий. Давайте ещё раз.

Сколько рублей вы внесёте прямо сейчас, чтобы её погасить? 110

Отлично, Василий! Вы погасили долг. Спасибо!

**Рекомендация**

Обратите внимание: считать сумму внесённых средств не нужно, это не соответствует условию задачи.

**Что оценивается**

Input содержит корректное приглашение для ввода.

Результат вывода корректен.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 3. Слишком большие числа**

У неудачливого бухгалтера всё опять идёт наперекосяк: ему приносят такие большие счета, что числа не помещаются на бумаге.

Напишите программу, которая считала бы для него, сколько цифр во вводимом числе. Обратите внимание, что число 0 имеет одну цифру.

**Что оценивается**

Input содержит корректное приглашение для ввода.

Результат вывода корректен, правильно подсчитано количество цифр в числе.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

Используется цикл while.

Решение не использует работу со строками и операции над ними.

**Задача 4. Поставьте оценку!**

Вася выложил своё новое приложение на платформу для начинающих разработчиков, на ней пользователи могут ставить оценку приложению: от −100 до 100. Ему захотелось сравнить количество положительных и отрицательных отзывов, для этого он заранее отфильтровал все отзывы так, чтобы в конце были те, которые равны нулю.

Напишите программу, которая находит количество положительных и количество отрицательных чисел в последовательности. Последовательность заканчивается на числе 0.

Пример

Введите число: −4

Введите число: −90

Введите число: 6

Введите число: 0

Кол-во положительных чисел: 1

Кол-во отрицательных чисел: 2

**Что оценивается:**

Результат вычислений корректен.

Формат вывода соответствует примеру.

Input содержит корректное приглашение для ввода.

Ввод чисел осуществляется внутри цикла.

При вводе 0 происходит выход из цикла и производится вывод подсчитанных положительных и отрицательных чисел.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 5. Обычный день на работе**

Максим программирует целый день на работе и вечером идёт домой. Каждый час начальство кидает ему несколько задач, которые нужно решить до следующего рабочего часа. Вдобавок каждый час Максиму звонит супруга. Он знает, что если он возьмёт трубку, то жена попросит зайти вечером в магазин.

Напишите программу, в которой считается, сколько задач выполнил Максим за день (восемь часов). Если он хотя бы раз взял трубку, то в конце дополнительно выводите сообщение: «Нужно зайти в магазин».

Пример

Начался восьмичасовой рабочий день.

1-й час

Сколько задач решит Максим? 1

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

2-й час

Сколько задач решит Максим? 2

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

3-й час

Сколько задач решит Максим? 3

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

4-й час

Сколько задач решит Максим? 4

Звонит жена. Взять трубку? (1 — да, 0 — нет): 1

5-й час

Сколько задач решит Максим? 5

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

6-й час

Сколько задач решит Максим? 1

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

7-й час

Сколько задач решит Максим? 2

Звонит жена. Взять трубку? (1 — да, 0 — нет): 1

8-й час

Сколько задач решит Максим? 3

Звонит жена. Взять трубку? (1 — да, 0 — нет): 0

Рабочий день закончился. Всего выполнено задач: 21

Нужно зайти в магазин.

**Что оценивается**

Input содержит корректное приглашение для ввода.

Формат вывода соответствует примеру.

Число итераций не превосходит число рабочих часов.

Используется флаг для учёта факта звонка жены.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 6. Вклады**

Вклад в банке составляет X рублей. Ежегодно он увеличивается на P процентов, после чего дробная часть копеек отбрасывается. Определите, через сколько лет вклад составит не менее Y рублей.

Напишите программу, которая по данным числам X, Y, P определяет, сколько лет пройдёт, прежде чем сумма достигнет значения Y.

**Что оценивается:**

Результат вычислений корректен.

Input содержит корректное приглашение для ввода.

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 7. Игра «Угадай число»**

В одной из практических работ мы делали задачу, где папа-программист написал для сына программу, которая просит его угадать число. Недостаток программы был в том, что бедному сыну приходилось её каждый раз перезапускать, чтобы угадывать число. Теперь, когда мы знаем гораздо больше, пришло время исправить этот недостаток и заодно немного улучшить саму игру.

Напишите программу-игру, которая запрашивает у пользователя число до тех пор, пока он его не отгадает. Выводите сообщения в соответствии с примером.

Пример (загадали число 7)

Введите число: 3

Число меньше, чем нужно. Попробуйте ещё раз!

Введите число: 10

Число больше, чем нужно. Попробуйте ещё раз!

Введите число: 8

Число больше, чем нужно. Попробуйте ещё раз!

Введите число: 7

Вы угадали! Число попыток: 4

**Что оценивается**

Формат вывода соответствует заданию.

Input содержит корректное приглашение для ввода.

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

**Задача 8. Игра «Компьютер угадывает число»**

Поменяйте мальчика и компьютер из прошлой задачи местами. Теперь мальчик загадывает число между 1 и 100 (включительно). Компьютер может спросить у мальчика: «Твоё число равно, меньше или больше, чем число N?», где N — число, которое хочет проверить компьютер. Мальчик отвечает одним из трёх чисел: 1 — равно, 2 — больше, 3 — меньше.

Напишите программу, которая с помощью цепочки таких вопросов и ответов мальчика угадывает число.

Дополнительно: сделайте так, чтобы можно было гарантированно угадать число за семь попыток.

Подсказка

При желании найдите описание алгоритма бинарного поиска и попробуйте применить в этой задаче. Разбор подобного решения будет в следующем модуле.

**Что оценивается**

Результат вычислений корректен.

Input содержит корректное приглашение для ввода.

Правильное употребление пробелов после запятых и при бинарных операциях.

**Что оценивается (общее)**

Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Правильное употребление пробелов после запятых и при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками: print (), input () — неверно, print() — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные имеют корректные названия, в качестве имён не используются имена встроенных функций ([список встроенных функций](https://docs.python.org/3.7/library/functions.html) — официальная документация).

**Советы и рекомендации**

Стиль оформления программ на языке Python [PEP 8](https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html).

Приоритет арифметических операций [PEP 8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence) (мы вводим and, or).

[Список встроенных функций](https://docs.python.org/3.7/library/functions.html) (официальная документация).

# Модуль 7

## 7.2 Цикл for. Работа со списком чисел

**Задание 1. Дом для семьи**

Максим написал программу, которая должна определять, подходит ли земельный участок для его семьи или нет. Живут они втроем, вот и условие будет таким же: если количество квадратных метров делится на 3, то участок подходит.

For in meters 100,90,95,87,102:

if meters % 3 == 1:

print(meters, 'Подходит')

else:

print(meters, 'Не подходит')

Скопируйте программу в редактор и исправьте её. Убедитесь, что она работает правильно и решает задачу Максима.

**Задание 2. Таблица степеней**

Аркадию для выступления с докладом нужно выучить таблицу степеней для определённых чисел. Правда, память у него работает довольно необычно, и ему проще учить их в нужном ему порядке.

Напишите программу, которая выводит вторую, третью и четвёртую степень для каждого числа в отдельной строке (первая строка - степени для числа 3, вторая строчка - степени для числа 7 и т.д.). Числа: 3,7,5,6,4.

Результат:

9 27 81

49 343 2401

25 125 625

36 216 1296

16 64 256

**Задача 3. Лотерея 2**

Напишите программу для немного усложнённой версии задачи про выигрышные билеты. Есть заранее известные номера билетов: 345, 19, 87, 1020 и 421 (можете брать свои номера, не стесняйтесь). Теперь, билет считается выигрышным, если номер билета - трёхзначное число и оно делится на 5. Выведете в консоль сообщение для каждого выигрышного билета и количество победителей.

## 7.3 Функция range

**Задача 1. Квадраты чисел**

Напишите программу, которая выводит квадраты чисел от 0 до 10. Для этого используйте цикл for и функцию range.

**Задача 2. Кукушка**

Напишите программу, которая имитировала бы часы с кукушкой. В начале работы она спрашивает, который час, а затем нужное количество раз пишет “Ку-ку!”.

**Задача 3. Любовь с первой цифры (цикл for)**

Перепишите программу из прошлого модуля, используя цикл for.

Паша очень любит математику. Настолько сильно, что у него по всей комнате висят всякие таблицы умножения, сложения, какие-то графики, формулы. И теперь он захотел распечатать таблицу степеней двойки, у них как раз началась новая тема по информатике.

Используя цикл for, выведите на экран для числа 2 его степени от 0 до 20.

## 7.4 Функция range с началом отсчета

**Задача 1. Квадраты превратились в кубы**

Напишите программу, которая выводит кубы чисел (число в третьей степени), лежащих в диапазоне от 1 до 10.

Результат:

1

8

27

64

125

216

343

512

729

1000

**Задача 2. Сумма чисел**

Напишите программу, где пользователь вводит любые два целых положительных числа. А программа суммирует все числа в диапазоне от первого до второго. Гарантируется, что первое введённое число всегда меньше второго.

Пример:

Введите первое число: 5

Введите второе число: 10

Сумма чисел от 5 до 10 равна 45

**Задача 3. Поел — можно и поспать, поспал — можно и поесть**

Напишите программу, разобранную в уроке.

У Саши интересный режим сна: он может проснуться когда угодно, хоть ночью, но засыпает всегда до того, как наступит полночь, обычно в 23 часа. А ещё он очень любит поесть. Он ест каждый час и если съедает больше 2000 калорий, то он просто падает спать. Напишите программу, которая поможет Саше понять, всё ли с ним хорошо. Для этого нужно посчитать, сколько он всего съест калорий и сколько часов будет бодрым.

## 7.6 Практическая работа

**Задача 1. Должники**

В базе банка хранятся данные и должников, и законопослушных граждан. Каждому человеку система присваивает свой номер. У нас есть случайная выборка из десяти номеров. Правда, иногда база глючит и выдаёт номер с отрицательным значением. А ещё по статистике, которую собрал наш «МирПрогБанк», каждый второй пользователь брал кредит и не выплатил его вовремя, то есть является должником.

Напишите программу, которая при вводе десяти чисел определяет, сколько из них являются одновременно чётными и положительными.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, не только a, b, c, d;

задача решена с помощью конструкции for.

**Задача 2. Посчитай чужую зарплату...**

Бухгалтер устала постоянно считать вручную среднегодовую зарплату сотрудников компании и, чтобы облегчить себе жизнь, обратилась к программисту.

Напишите программу, которая принимает от пользователя зарплату сотрудника за каждый из 12 месяцев и выводит на экран среднюю зарплату за год.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

вывод содержит описание результата (не выведено одно число без описания);

переменные имеют значащие имена, не только a, b, c, d;

задача решена с помощью конструкции for.

**Задача 3. Факториал**

Мы всё ближе и ближе подбираемся к серьёзной математике. Одна из классических задач — задача на нахождение факториала числа. И в будущем мы с ней ещё встретимся.

Дано натуральное число n. Напишите программу, которая находит n! (n-факториал).

Запись n! означает следующее:

n! = 1 \* 2 \* 3 \* 4 \* 5 \* … \* n

Пример:

Введите число: 5

Факториал числа 5 равен 120

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру;

вывод содержит описание, не только результат в виде числа;

для решения используется цикл for, а не встроенные функции или рекурсия;

переменные имеют значащие имена, не только a, b, c, d.

**Задача 4. Успеваемость в классе**

В классе N человек. Каждый из них получил за урок по информатике оценку: 3, 4 или 5, двоек сегодня не было. Напишите программу, которая получает список оценок — N чисел — и выводит на экран сообщение о том, кого сегодня больше: отличников, хорошистов или троечников.

Замечание: можно предположить, что количество отличников, хорошистов, троечников различно.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

в выводе присутствует сообщение о том, кого больше;

для решения используется цикл for, а не встроенные функции или рекурсия;

переменные имеют значащие имена, не только a, b, c, d.

**Задача 5. Отрезок**

Напишите программу, которая считывает с клавиатуры два числа: a и b, — считает и выводит в консоль среднее арифметическое всех чисел из отрезка [a; b], кратных числу 3.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

в выводе присутствует описание результата;

для решения используется конструкция for.

**Задача 6. Замечательные числа**

Напишите программу, которая находит и выводит все двузначные числа, равные утроенному произведению своих цифр. К таким относятся, например, 15 и 24.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

допускается вывод без описания, достаточно вывести числа;

переменные имеют значащие имена, не только a, b, c, d.

**Задача 7. Пропавшая карточка**

Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Напишите программу, которая поможет найти потерянную карточку, если номера оставшихся известны. Найдите её, зная номера оставшихся карточек.

Введите число карточек — N.

Затем введите N − 1 номера оставшихся карточек. Они могут быть введены в любом порядке.

Пример:

Введите количество карточек: 5

Введите номер оставшейся карточки: 1

Введите номер оставшейся карточки: 4

Введите номер оставшейся карточки: 5

Введите номер оставшейся карточки: 3

Номер пропавшей карточки: 2

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержат корректные приглашения для ввода;

переменные имеют значащие имена, не только a, b, c, d.

**Что оценивается (общее)**

Использование именованных индексов, не просто i (видео 7.2).

Ввод правильных чисел, без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).

Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками: print (), input () — неверно, print() — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

# Модуль 8

## 8.2 Алгоритмические задачи со счётными циклами

**Задача 1. Таблица кубов**

Паше для проекта по математике нужна таблица кубов (третья степень числа) только чётных чисел от 1 до N. Напишите программу, которая выведет ему эту таблицу на экран. Не используйте условные операторы, выведите формулу, как мы сделали это в уроке.

**Задача 2. Деление клетки**

Реализуйте программу, разобранную в уроке.

В одной лаборатории наблюдают за одноклеточной амёбой. Мы знаем, что каждые три часа она делится на 2 клетки. Нам нужно для этой лаборатории написать программу, которая будет выводить сколько прошло часов и сколько получилось клеток. Также нас попросили писать на каждом этапе деления сколько осталось часов до завершения наблюдения, чтобы ученым было проще формулировать выводы на определённом этапе наблюдения.

Пример сообщений:

Прошло часов: 3.

Клеток: 2.

Часов до конца эксперимента: 12

Прошло часов: 6...

**Задача 3. Квадраты нечётных чисел**

Вводится число N. Напишите программу, которая выводит квадраты нечетных чисел от 1 до N. Нельзя использовать условные операторы. Можно использовать цикл while, но постарайтесь всё-таки решить с помощью конструкции for in range. Не нужно искать решение в интернете, попробуйте подумать сами, в следующем уроке мы обязательно разберём эту задачу.

## 8.3 Функция range: start, stop, step

**Задача 1. Степень нечётного числа**

Выведите третью степень каждого нечётного числа в диапазоне от единицы до указанного пользователем числа включительно. Для этого используйте шаг внутри функции range.

**Задача 2. Театр**

Ваню заставили пойти в театр на балет. Ему стало там настолько скучно, что он придумал себе очень странное развлечение: считать сумму номеров каждого пятого стула в рядах.

Напишите программу для вычисления суммы каждого пятого числа, лежащего в диапазоне от единицы до N. Использовать условный оператор нельзя.

Пример:

Введите число: 21

Номер стула: 1

Номер стула: 6

Номер стула: 11

Номер стула: 16

Номер стула: 21

Сумма: 55

**Задача 3. Диета**

Саша просыпается когда угодно, но в 23 часа уже точно идёт спать. Питается Саша следующим образом: каждые 3 часа он выпивает литр воды и съедает N калорий. Пить и есть он, кстати, начинает сразу как только проснётся. Напишите программу, которая считает сколько он выпьет литров воды и сколько калорий он съест перед тем как пойдёт спать.

## 8.4 Отрицательный шаг в функции range

**Задача 1. Прятки**

Наверное, все мы в детстве играли в прятки. И конечно же всегда был “голя” - тот, кто всех ищет. И пока все прячутся, он с закрытыми глазами ведёт обратный отсчёт секунд.

Напишите программу, которая получает на вход количество секунд N и выводит все числа от N до 1 в порядке убывания. В конце выведите сообщение “Я иду искать!”

Пример:

Введите кол-во секунд: 5

5

4

3

2

1

Я иду искать!

**Задача 2. Армия**

У офицера сегодня очень плохое настроение: он выстроил всех солдат в шеренгу и, начиная с конца, начал спрашивать у каждого четвёртого солдата сколько правил прописано в воинском уставе. Количество правил с каждым разом меняется. Если солдат ответил неверно, то применяется закон под названием “упал-отжался”. Количество отжиманий считается как 10 умножить на порядковый номер солдата в шеренге. Напишите программу, которая посчитает сколько в сумме получится таких отжиманий.

Усложнение: напишите программу так, чтобы офицер начинал спрашивать не с последнего солдата в шеренге, а с четвертого с конца. Если солдат в шеренге слишком мало, офицер не спрашивает никого

**Задача 3. Прятки 2**

Пока все прятались, “голя” решил схитрить и считать секунды быстрее, чем нужно.

Напишите программу, которая выводит только чётные числа в порядке убывания от N до 1 включительно, используя цикл for. Нельзя использовать условный оператор.

## 8.6 Практическая работа

**Задача 1. Космическая еда**

Ваш космический корабль потерпел крушение на пустынной планете. Еда здесь не растёт, но вы спасли из обломков 100-килограммовый мешок гречки. Из прошлого опыта вы знаете, что если будете экономно питаться, то у вас будет уходить по четыре килограмма гречки в месяц.

Чтобы прикинуть гречневый бюджет, вы решили написать программу, которая выведет информацию о том, сколько килограммов гречки у вас должно быть в запасе через месяц, два и так далее, пока она не закончится. Используйте цикл for.

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

для решения использован цикл for и range c шагом;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 2. Долги**

«МирПрогБанк» наконец-то разделил законопослушных граждан и должников и поместил их в разные базы. Но банк не торопится сильно давить на неплательщиков. Операторам банка дали задание позвонить каждому пятому должнику из списка (нумерация начинается с нуля) и уточнить, какую сумму каждый из них задолжал банку.

Напишите программу, которая получает данные о количестве должников, а затем спрашивает у каждого пятого (начиная с нуля) его долг. В конце выводится общая сумма долгов.

Пример 1

Введите количество должников: 13

Должник с номером 0

Сколько должны? 1000

Должник с номером 5

Сколько должны? 5000

Должник с номером 10

Сколько должны? 2000

Общая сумма долга: 8000

Пример 2

Введите количество должников: 10

Должник с номером 0

Сколько должны? 1000

Должник с номером 5

Сколько должны? 5000

Общая сумма долга: 6000

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

в выводе присутствует общая сумма долга;

формат вывода соответствует примеру (не выведены числа без описания);

для решения использован цикл for и range c шагом;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 3. Микроволновка**

Мы разрабатываем микропрограмму — таймер обратного отсчета для микроволновых печей. Некоторым пользователям не нравится пищащий звук.

Есть задача убрать звук по готовности и заменить его сообщением на LED-экране. В нашем случае будем выводить в консоль сообщение с обратным отсчетом в секундах от “reverse\_timer” до момента готовности, то есть «0» секунд, и спрашивать пользователя, готов ли он забрать еду.

Пользователь в любой момент может прервать режим разогрева, введя «1» (то есть ответить «Да, еда готова»), тогда программа выводит на экран сообщение «Ваша еда готова, можете забрать» и показывает, на какой секунде был прерван таймер.

Если пользователь отвечает «0», что равноценно «Нет», то таймер уменьшается. Когда он достигнет «0» секунд, выводим сообщение «Ваша еда готова, осторожно горячo!»

В данном задании используем цикл for.

“reverse\_timer” – переменная счетчик, значение которой запрашиваем у пользователя через функцию ввода input.

Задайте время до обнуления таймера.

Используйте цикл for.

На каждой итерации задавайте персонажу вопрос, хочет ли он сейчас остановить разогрев или нет.

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

вывод содержит обратный отсчёт до самого конца или до того момента, когда пользователь решит остановить разогрев;

задача решена с помощью цикла for, возможно, с range и отрицательным шагом;

input содержит корректное приглашение для ввода;

последней секундой считается 1;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 4. Среднее на отрезке**

Напишите программу, которая считывает с клавиатуры три числа a, b и c, считает и выводит на консоль среднее арифметическое всех чисел из отрезка [a; b], кратных числу c.

**Рекомендации**

Функция range(start, stop) не включает границу stop, останавливается, не доходя до неё.

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 5. Функция**

Перед изучением функций из программирования Саша решил оживить свои знания о функциях математики. Помогите Саше написать программу, которая будет считать значение функции в каждой точке отрезка с нужным шагом, начиная с конца).

Напишите программу, которая получает на вход начало и конец отрезка, а также шаг (отрицательный), а затем высчитывает функцию y в каждой точке отрезка и выводит ответ на экран с нужным шагом, начиная с конца.

Сама функция выглядит так:‌

Пример

Введите начало отрезка: −2

Введите конец отрезка: 2

Введите шаг: −1

В точке 2 функция равна 9

В точке 1 функция равна 0

В точке 0 функция равна 1

В точке −1 функция равна 6

В точке −2 функция равна 9

**Рекомендации**

Функция range(start, stop) не включает границу stop, останавливается, не доходя до неё.

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

input содержит корректное приглашение для ввода;

в программе происходит проверка границ перед созданием range и, возможно, их разворот, как и изменение знака шага;

формат вывода соответствует примеру;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 6. Стипендия**

Ежемесячная стипендия студента составляет educational\_grant рублей, а расходы на проживание превышают стипендию и составляют expenses рублей в месяц.

Рост цен ежемесячно увеличивает расходы на 3%, кроме первого месяца. Обратите внимание, что каждый месяц цены увеличиваются на 3% относительного прошлого месяца.

Составьте программу расчёта суммы денег, которую необходимо получить у родителей один раз в начале обучения, чтобы можно было прожить учебный год (десять месяцев), используя только эти деньги и стипендию.

Пример

Введите стипендию: 10000

Введите расходы на проживание: 12000

месяц траты 12000 не хватает 2000

месяц траты 12360.0 не хватает 4360.0

месяц траты 12730.8 не хватает 7090.8

месяц траты 13112.7 не хватает 10203.52

месяц траты 13506.1 не хватает 13709.63

месяц траты 13911.2 не хватает 17620.92

месяц траты 14328.6не хватает 21949.55

месяц траты 14758.4 не хватает 26708.03

месяц траты 15201.2 не хватает 31909.27

месяц траты 15657.2 не хватает 37566.55

Нужно попросить у родителей 37566.55 рублей

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задача 7. Сумма ряда**

Дано натуральное число N. Напишите программу для вычисления суммы N элементов последовательности по формуле — порядковый номер элемента (расчёт начинается с нуля).

Примеры расчётов:

При N = 4 элементы выражения будут равны:

n = 0

elem = (−1) \*\* 0 \* (½) \*\* 0 = 1

n = 1

elem = (−1) \*\* 1 \* (½) \*\* 1 = (−½)

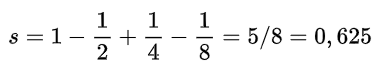
n = 2

elem = (−1) \*\* 2 \* (½) \*\* 2 = ¼

n = 3

elem = (−1) \*\* 3 \* (½) \*\* 3 = (−⅛)

Сумма равна:



При N = 6 элементы выражения будут равны:

n = 0

elem = (−1) \*\* 0 \* (½) \*\* 0 = 1

n = 1

elem = (−1) \*\* 1 \* (½) \*\* 1 = (−½)

n = 2

elem = (−1) \*\* 2 \* (½) \*\* 2 = ¼

n = 3

elem = (−1) \*\* 3 \* (½) \*\* 3 = (−⅛)

n = 4

elem = (−1) \*\* 4 \* (½) \*\* 4 = (1/16)

n = 5

elem = (−1) \*\* 5 \* (½) \*\* 5 = (−1/32)

Сумма равна:

s = 1 − ½ + ¼ − ⅛ + 1/16 − 1/32 = 21/32 = 0,65625

P. S. Не стоит выполнять расчёты каждого элемента вручную, используйте цикл.

Примеры вывода в консоль:

Введите N: 4

Ответ: 0,625

Введите N: 6

Ответ: 0,65625

**Рекомендации**

Так как n изменяется от нуля до N, эти изменения можно легко задать при помощи цикла и функции range.

Подставляя переменную такого цикла в формулу, вы на каждой итерации будете получать один элемент последовательности, останется лишь добавить их суммирование.

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3);

программа работает корректно при вводе разных значений числа N.

Допускается вывод без описания, достаточно вывести сумму ряда.

**Задача 8. Кинотеатр**

X мальчиков и Y девочек пошли в кинотеатр и купили билеты на идущие подряд места в одном ряду. Напишите программу, которая выдаст, как нужно сесть мальчикам и девочкам, чтобы рядом с каждым мальчиком сидела хотя бы одна девочка, а рядом с каждой девочкой — хотя бы один мальчик.

На вход подаются два числа: количество мальчиков X и количество девочек Y. В ответе выведите какую-нибудь строку, в которой будет ровно X символов B, обозначающих мальчиков, и Y символов G, обозначающих девочек, удовлетворяющую условию задачи. Пробелы между символами выводить не нужно. Если рассадить мальчиков и девочек согласно условию задачи невозможно, выведите строку «Нет решения».

Пример 1

Введите количество мальчиков: 5

Введите количество девочек: 5

Ответ: BGBGBGBGBG

Пример 2

Введите количество мальчиков: 5

Введите количество девочек: 3

Ответ: BGBGBBGB

Пример 3

Введите количество мальчиков: 100

Введите количество девочек: 1

Ответ: Нет решения

**Что оценивается**

Задача решена успешно, если:

результат вывода соответствует условию;

формат вывода соответствует примеру (выведенные числа сопровождаются текстовым описанием);

input содержит корректное приглашение для ввода;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Общие рекомендации**

Выводите значение счётчика при отладке программы (видео 8.4).

Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).

Помните о приоритете арифметических операций [PEP 8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence)

Если в одной строке пайтон встречает несколько операций, то он будет выполнять их не одновременно а по порядку, указанному в этой таблице.

Пример:

a, b = 1, 1

if a or b > 5:

внутри if пайтон первым делом начнёт выполнять действие “or”

Он проверит отдельно то, что слева от него и, если получит ответ False - проверит то, что справа (если ответ будет True, то то, что справа проверяться не будет вовсе)

Справо находится переменная ‘a’, без каких-то дополнительных операторов

Пайтон проверит - ‘a’ является 0? если да, то он заменит переменную на False, иначе он заменит её на True.

И т.к. на этом этапе он получит ответ True, то b > 5 он даже выполнять не начнёт.

Почитайте руководство по написанию кода на Python:

[PEP 8](https://www.python.org/dev/peps/pep-0008/) (на английском языке),

[PEP 8](https://letpy.com/python-guide/pep8/) (на русском языке).

Обратите внимание на [список встроенных функций](https://docs.python.org/3.7/library/functions.html) (официальная документация).

Прочитайте описание конструкции [for-else](https://docs.python.org/3/tutorial/controlflow.html#break-and-continue-statements-and-else-clauses-on-loops).

Прочитайте описание [enumerate](https://docs.python.org/3/library/functions.html#enumerate) для исключения конструкций вида for i in range(len(array)).

**Что оценивается**

Использованы именованные индексы, не просто i (видео 7.2).

Правильные числа введены без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).

Практическая работа сдана через Fork или корректно оформленный repl.it.

Правильно оформлен input, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Есть пробелы после запятых, пробелы при бинарных операциях.

Нет пробелов после имён функций и перед скобками: “print ()”,“input ()” — неверно, “print()” — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные имеют корректные названия, и в качестве имён не используются имена встроенных функций (список встроенных функций — официальная документация).

# Модуль9

## 9.2 Сравнение строк

**Задача 1. Урок литературы**

Выполните задание, разобранное в уроке.

К уроку литературы нужно было прочитать “Евгения Онегина”. Учитель задаёт вопрос пяти случайным детям. Она задаёт вопрос “Кто написал произведение?” и если ученик не угадывает, то учитель ставит двойку и спрашивает следующего. Если хоть кто-то угадает, то вопросы больше не задаются. Напишите программу, которая посчитает количество двоечников из этих пяти.

**Задача 2. Начальник**

Напишите программу для робота-начальника. Он спрашивает у пользователя, выполнил ли он задание, которое выдавал вчера, и продолжает это делать до тех пор, пока тот не ответит ему “Да, конечно, сделал”.

**Задача 3. Дразнилка**

Напишите программу-дразнилку “Купи слона”. Она спрашивает у пользователя, как его зовут, затем отвечает “%username%, купи слона”. Дальше, что бы он ни говорил, она передразнивает: Все говорят “...”, а ты купи слона!

Пример:

Как тебя зовут? Дима

Дима, купи слона!

Хорошо, давай куплю

Все говорят Хорошо, давай куплю, а ты купи слона!

...

## 9.3 Цикл for: итерирование по строке

**Задача 1. Python!**

Напишите программу, которая выводит в отдельную строчку каждый символ фразы “Python!”.

**Задача 2. Вирус**

Ваня экспериментирует с различного рода компьютерными вирусами, которые портят жизнь людям. На просторах Интернета он нашёл код довольно необычного вируса, который “поворачивает” весь текст в документе и повторяет каждый символ 3 раза.

Пользователь вводит текст. Напишите программу, которая выводит каждый символ текста в отдельной строке и три раза.

Пример:

Введите текст: Привет!

ППП

ррр

иии

ввв

еее

ттт

!!!

**Задача 3.**

Мы входим в команду разработки нового текстового редактора и нам поручили разработать для него подсчёт нужного символа в тексте, а именно - буквы Ы. Причём отдельно с верхним регистром и отдельно с нижним.

Напишите программу, которая считает количество больших и количество маленьких букв Ы в тексте и выводит ответ на экран.

Пример:

Введите текст: Прыг скок

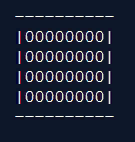
Больших букв Ы: 0

Маленьких букв Ы: 1

## 9.4 Дополнительные возможности функции print

**Задача 1. Доска**

Используя новые знания с циклами и оператором end, выведете на экран “доску”, которую вы делали в первом модуле



**Задача 2. Локальная сеть**

У Никиты дома есть много компьютеров, которые он хочет подключить к одной локальной сети. Для этого ему нужно сгенерировать айпи адрес для каждого компьютера. Айпи адрес записывается как 4 числа, которые отделяются точкой. Не долго думая, для генерации Никита решил использовать арифметическую прогрессию, причём первые 3 числа в адресе - это члены прогрессии, а последнее число - это её сумма.

Напишите программу, где пользователь вводит первый член прогрессии и разность и в результате получает айпи-адрес.

**Задача 3. Табло**

Петя пишет компьютерную спортивную игру, где каждый “гол” это десять очков. Он хочет сделать красивое табло с градацией этих очков.

Пользователь вводит число N (N >= 0). Реализуйте программу, которая выводит в одну строчку каждое десятое число, разделяя их символами “-=-”. Эти же символы стоят перед первым числом и после последнего.

Пример:

Введите число: 50

-=- 0 -=- 10 -=- 20 -=- 30 -=- 40 -=- 50 -=-

## 9.6 Практическая работа

**Задание 1. «Я стал новым пиратом!»**

Старому капитану нужно пополнить команду, но на корабль попадут только достойные! Он отобрал десять человек. Те, кто крикнет слово «Карамба», попадут на борт.

Пользователь вводит десять слов. Напишите программу, которая определяет, сколько из них совпадают со словом «Карамба».

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

программа игнорирует регистр первой буквы К;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задание 2. Кривой мессенджер**

Существует мессенджер, в котором иногда возникают неполадки при передаче сообщений: в них попадает лишний символ — звёздочка. Пользователям это надоело, поэтому они стали уходить в другие сервисы. Но один из них заинтересовался, на каких позициях обычно появляется звёздочка. Чтобы выяснить это, пользователю необходимо подготовить строки, в которых символ «\*» встречается ровно один раз.

Напишите программу, которая определяет порядковый номер звёздочки в строке.

Пример:

Введите текст: «Пр\*ивет как дела».

Символ «\*» стоит на позиции 3.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

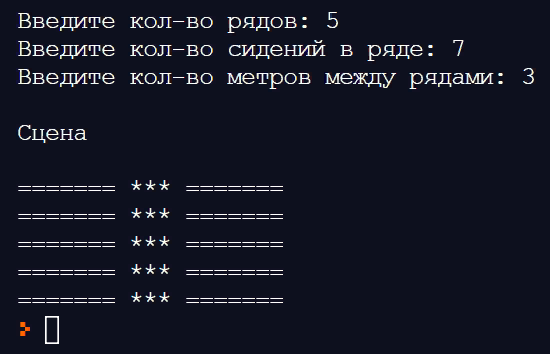
вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задание 3. Театр**

В городе планируют построить театр под открытым небом, но для начала нужно оценить, сколько сделать рядов, сидений в них и каким должно быть расстояние между рядами.

Напишите программу, которая получает на вход количество рядов, сидений и свободных метров между рядами, а затем выводит примерный макет театра на экран.



**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Рекомендации по выполнению

Можно воспользоваться синтаксическим сахаром: умножением строки на число и конкатенацией (объединением строк). Для начала получите одну строку.

**Задание 4. Марсоход-2**

К роботу Валли отправили «коллегу» Билли. Это его первая высадка на Марс, поэтому его тестируют в прямоугольном помещении размером 15 × 20 м. Марсоход высаживается в центре комнаты (в точке 8, 10), затем управление им передаётся оператору, то есть пользователю вашей программы.

Программа спрашивает, в какую сторону оператор хочет направить робота: север (клавиша W), юг (клавиша S), запад (клавиша A) или восток (клавиша D). Оператор делает выбор, марсоход перемещается в эту сторону на один метр, а программа сообщает новую позицию робота. Если марсоход упёрся в стену, он не должен пытаться переместиться в сторону стены — в этом случае его позиция не меняется.

Создайте программу для управления роботом Билли.

Пример:

[Программа]: Марсоход находится на позиции 6, 19, введите команду:

[Оператор]: A

[Программа]: Марсоход находится на позиции 5, 19, введите команду:

[Оператор]: W

[Программа]: Марсоход находится на позиции 5, 20, введите команду:

[Оператор]: W

[Программа]: Марсоход находится на позиции 5, 20, введите команду:

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру;

марсоход не двигается, если достигает границы участка;

программа игнорирует регистр вводимых команд;

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Рекомендации по выполнению**

Обращайте внимание на границы.

Попробуйте изменять положение марсохода только после проверки координат.

Ввод команд происходит только в верхнем регистре, учитывать нижний не нужно.

Старайтесь максимально уменьшать количество дублирований кода. Если действие выполняется независимо от условий - его не стоит дублировать в каждом условии.

**Задание 5. Великий и могучий**

Паоло изучает русский язык: занимается по учебникам, читает книги, слушает музыку. Ему особенно понравилась книга «Преступление и наказание». Паоло решил найти самое длинное слово в этой книге, чтобы сравнить его с аналогом на своём языке.

Напишите программу, которая получает на вход текст и находит длину самого длинного слова в нём. Слова в тексте разделяются одним пробелом.

Пример:

Введите текст: Меня зовут Пётр.

Самое длинное слово, букв: 5.

Введите текст: Меня зовут Василий

Самое длинное слово, 7 букв

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Рекомендации по выполнению

При помощи функции print убедитесь, что счётчики обнуляются в нужный момент.

Не забывайте, что не все условия можно собирать в один условный блок. Некоторые из них должны срабатывать независимо друг от друга.

**Задание 6. Коровы**

Для коров есть 10 стойл. В каждом из них условия для животных разные, поэтому и молока они дают по-разному. В первом стойле производят 2 литра в день, во втором — 4, в третьем — 6, далее — 8, 10, 12, 14, 16, 18 и 20. При этом коровы находятся не во всех стойлах. Свободные и занятые обозначаются строкой из букв a и b, где a — свободное стойло, b — занятое.

Напишите программу для подсчёта получаемого молока в коровнике. Важно учитывать следующее взаимодействие: пользователь вводит строку из десяти символов a и b. Необходимо определить, сколько в итоге будет произведено молока за день.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

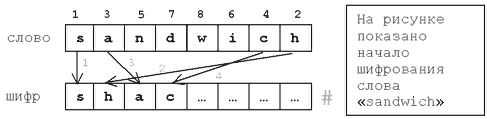
input содержит корректное приглашение для ввода;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

**Задание 7. Метод бутерброда**

В секретном агентстве Super-Secret-no решили использовать «метод бутерброда» для шифрования переписки своих сотрудников. Сначала буквы слова нумеруются в таком порядке: первая буква получает номер 1, последняя буква — номер 2, вторая — номер 3, предпоследняя — номер 4, потом третья… и так для всех букв (см. рисунок). Затем все буквы записываются в шифр в порядке своих номеров.



Например, слово sandwich зашифруется в shacnidw.

Программист Super-Secret-no написал программу шифрования и уволился. Теперь агенты не могут понять, что они написали друг другу.

Напишите программу-дешифратор, которая расшифровала бы введённые сообщения.

Пример:

Введите зашифрованное сообщение: shacnidw.

Расшифрованное сообщение: sandwich.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

как и в сложении чисел - в сложении строк можно менять слагаемыми местами. только в отличии от чисел - такая перемена изменит результат:

строка += “а” – так вы добавите символ в конец строки

строка = “а” + строка – так вы добавите символ в начало строки (что позволит сложить строку задом наперед)

**Что оценивается (общее)**

Используются именованные индексы, не просто i (видео 7.2).

Вводятся правильные числа, без дополнительных действий со стороны пользователя и без использования +1 (видео 7.4).

Input оформлен правильно, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Используются пробелы после запятых и при бинарных операциях.

Отсутствуют пробелы после имён функций и перед скобками: print (), input () — неверно, print() — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные имеют корректные названия, в качестве имён не используются имена встроенных функций. Подробнее о них вы можете узнать в списке встроенных функций.

**Рекомендации по выполнению**

Выводите значение счётчика при отладке программы (видео 8.4).

Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).

Помните о приоритете арифметических операций PEP 8 (мы вводим and, or).

Почитайте руководство по написанию кода на Python:

PEP 8 (на английском языке),

PEP 8 (на русском языке).

Обратите внимание на список встроенных функций (официальная документация).

Прочитайте описание конструкции for-else.

Прочитайте описание enumerate для исключения конструкций вида for i in range(len(array)).

# Модуль 10

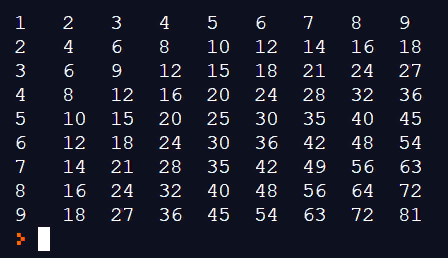
## 10.1 Работа со вложенными циклами

**Задача 1. Таблица умножения**

Математик Паша недавно заметил, что у него уже есть куча разных таблиц степеней, но нет самого основного — таблицы умножения. Пора бы это исправить.

Напишите программу, которая выводит таблицу умножения для чисел от 1 до 9. Для этого используйте конструкцию вложенного цикла: внешний отвечает за первый множитель, а внутренний — за второй.

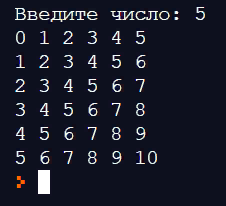
Дополнение: выведите настоящую таблицу умножения, без всяких знаков, только числа. Чтобы она получилась красивой и ровной, используйте литерал \t внутри оператора end. \t — это литерал табуляции, благодаря ему все числа выстраиваются в виде таблицы. Результат должен получиться таким:



**Задача 2. Таблица суммы**

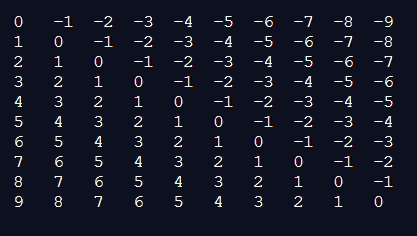
Напишите программу, которая запрашивает у пользователя число N и выводит таблицу суммы для чисел от 1 до N.

Пример:



**Задача 3. Анализ данных**

Отдел анализа данных сегодня получил вот такую интересную штуку:



Вам, как работнику этого отдела, дали задание понять, как и почему такое произошло. Напишите программу, которая выводит на экран такую таблицу.

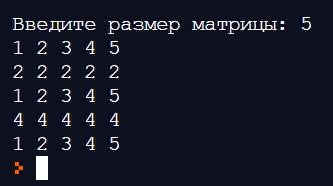
💡 Замечание: не забудьте про литерал табуляции \t.

## 10.2 Использование if во вложенных циклах

**Задача 1. Матрица**

Напишите программу, которая выводит квадратную матрицу размера N на N. В каждой нечётной строке матрицы идут числа от 1 до N, а в каждой чётной — просто числа, равные номеру этой строки.

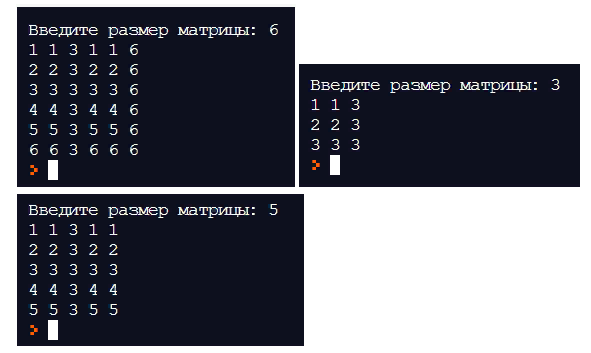
Пример:



**Задача 2. Чёрный ящик**

Преподаватель показал сту

дентам несколько результатов программы и сказал: «Кто догадается, что делает программа и как она это делает, получит зачёт автоматом». Студент Миша намерен получить этот зачёт. Он заметил, что в результатах программы есть определённая закономерность. Вот некоторые результаты:



Напишите программу, которая запрашивает у пользователя размер матрицы и выводит соответствующие результаты.

Подсказка: столбцы.

**Задача 3. Координатные оси**

Напишите программу, которая рисует координатные оси на поле 20×50. Результат должен получиться таким:

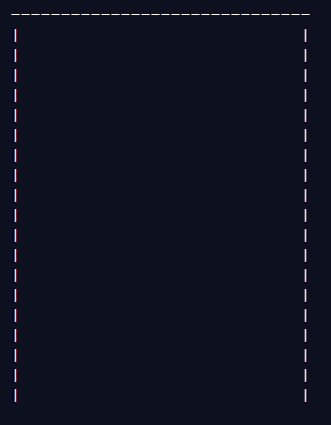


Что нужно поменять в коде, что в середине был не дефис, а вертикальная палочка?

## 10.3 Работа с двумя счетчиками в условном операторе

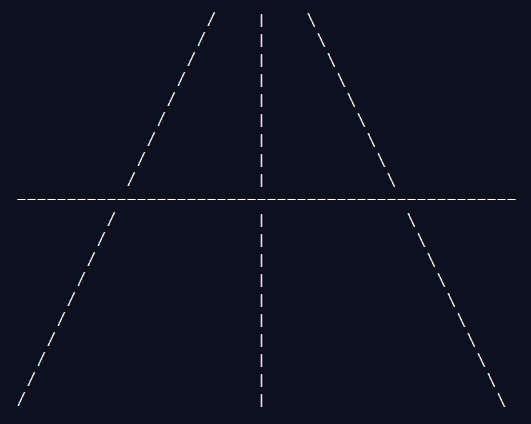
**Задача 1. Врата**

Напишите программу, которая выводит в консоль «врата», состоящие из тире, вертикальных линий и пробелов. Поле состоит из 20 строк и 30 столбцов (но не стесняйтесь пробовать и другие размеры).



**Задача 2. Дорога**

Мы делаем текстовую игру — гонку, и нам нужно вывести на экран что-то вроде трассы, где будут соревноваться наши машины. Напишите программу, которая выводит такую дорогу на экран (поле 20×50).

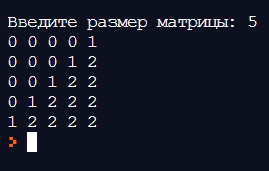


Что нужно сделать, чтобы обочины рисовались поверх горизонтальной линии?

**Задача 3. Диагональная матрица**

Напишите программу, которая получает на вход размер квадратной матрицы и выводит на экран по такому принципу диагональ из единиц с левого нижнего угла до верхнего правого, ниже диагонали — двойки, выше — нули.

Пример:



## 10.4 Решение задач с помощью вложенных циклов

**Задача 1. Электронная очередь**

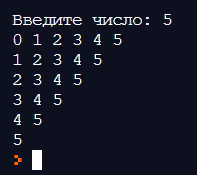
Нам дали заказ написать программу для электронной очереди. У каждого человека в очереди есть номер: нулевой, первый, второй, третий и так далее. Каждый час очередь уменьшается на одного человека, то есть уходит сначала нулевой номер, затем первый, второй и так далее. Наша программа получает на вход одно число — количество людей в очереди — и выводит на экран историю обслуживания очереди: какие номера в какой час оставались.

**Задача 2. Цифры больше пяти**

Пользователь вводит последовательность из N чисел. Напишите программу, которая подсчитывает общее количество цифр больше пяти во всей последовательности.

**Задача 3. Лестница чисел**

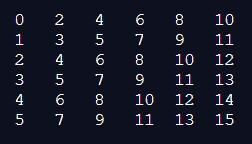
Пользователь вводит число N. Напишите программу, которая по этому числу выводит вот такую лестницу из чисел:



## 10.6 Практическая работа

**Задание 1. Тестовое задание**

Степан устраивается на работу и должен выполнить тестовое задание: проанализировать таблицу, понять, как она строится, и написать программу для её вывода на экран.



Помогите Степану реализовать программу.

Подсказка: обращайте внимание на номера столбцов и помните о литерале \t для табуляции.

**Что оценивается**

Задание считается успешно выполненным, если:

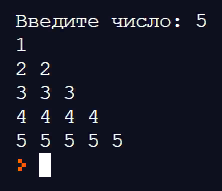
результат вывода соответствует условию;

задача решена с помощь циклов for или while;

формат вывода соответствует примеру.

**Задание 2. Лестница**

Напишите программу, которая выводит «лестницу» из чисел, когда пользователь вводит число N:



**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

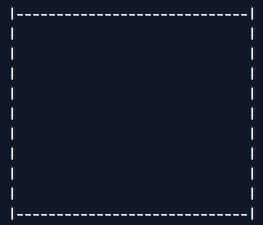
input содержит корректное приглашение для ввода;

может быть использована конструкция print и пустой input (видео 10.4);

формат вывода соответствует примеру.

**Задание 3. Рамка**

Напишите программу, которая рисует прямоугольную рамку с помощью символьной графики. Для вертикальных линий используйте символ вертикального штриха (|), а для горизонтальных — дефис (-). Пусть ширину и высоту рамки определяет пользователь.



**Что оценивается**

Задание считается успешно выполненным, если:

input содержит корректное приглашение для ввода;

формат вывода соответствует примеру;

в качестве элементов горизонтальной линии можно использовать дефисы (-) или нижние подчёркивания (\_).

**Задание 4. Простые числа**

Напишите программу, которая считает количество простых чисел в заданной последовательности и выводит ответ на экран.

[Простое число](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) делится только на себя и на единицу. Последовательность задаётся при помощи вызова ввода (input) на каждой итерации цикла. Одна итерация — одно число.

[Алгоритм нахождения простых чисел](https://habr.com/ru/articles/122538/)

[Решето эратосфена](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%AD%D1%80%D0%B0%D1%82%D0%BE%D1%81%D1%84%D0%B5%D0%BD%D0%B0)

[Тест простоты](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D1%82%D1%8B)

[Факторизация больших чисел, криптографический алгоритм RSA](https://ru.wikipedia.org/wiki/RSA).

Пример:

Введите количество чисел: 6.

Введите число: 4.

Введите число: 7.

Введите число: 20.

Введите число: 3.

Введите число: 11.

Введите число: 37.

Количество простых чисел в последовательности: 4.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Рекомендации по выполнению**

Простое число имеет только два делителя, поэтому для проверки нужно перебрать все числа от одного до него. Если загаданное число делится без остатка на что-то, кроме единицы или себя, то это уже не простое число.

**Задание 5. Наибольшая сумма цифр**

Пользователь вводит N чисел. Среди натуральных чисел, которые он указал, найдите наибольшее по сумме цифр. Выведите на экран это число и сумму его цифр.

**Что оценивается**

Задание считается успешно выполненным, если:

результат вывода соответствует условию;

input содержит корректное приглашение для ввода;

вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задание 6. Пирамидка**

Что нужно сделать

Напишите программу, которая выводит на экран равнобедренный треугольник (пирамидку), заполненный символами хештега (#). Пусть высоту пирамиды определяет пользователь.



**Что оценивается**

Задание считается успешно выполненным, если:

формат вывода соответствует примеру;

input содержит корректное приглашение для ввода.

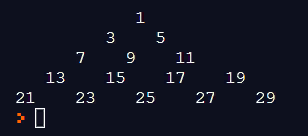
Советы и рекомендации

Вспомните, как выводился колонтитул вида -----!!!!!!-----.

Не забывайте, что для нас пробел — это пустое место, а для Python — это такой же символ, как и любой другой. Если нужно добавить отступ, необходимо использовать пробел или символ табуляции (\t).

**Задание 7. Пирамидка-2**

Напишите программу, которая получает на вход количество уровней пирамиды и выводит их на экран, заполняя нечётными числами:



**Что оценивается**

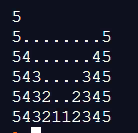
Задание считается успешно выполненным, если:

формат вывода соответствует примеру;

input содержит корректное приглашение для ввода.

**Задание 8. Яма**

Представьте, что вы разрабатываете компьютерную игру с текстовой графикой. Вам поручили создать генератор ландшафта. Напишите программу, которая получает на вход число N и выводит на экран числа в виде ямы:



**Что оценивается**

Задание считается успешно выполненным, если:

формат вывода соответствует примеру;

input содержит корректное приглашение для ввода.

**Что оценивается (общее)**

Используются именованные индексы, а не только i (видео 7.2).

Правильные числа вводятся без дополнительных действий со стороны пользователя и без использования +1 (видео 7.4).

Input оформлен правильно: нет пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Используются пробелы после запятых и при бинарных операциях.

Нет пробелов после имён функций и перед скобками: print (), input () — неверно, print() — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные названы корректно. Не используются имена встроенных функций. Подробнее о них вы можете узнать в списке встроенных функций.

**Советы и рекомендации**

При отладке программы выводите значение счётчика (видео 8.4).

Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).

Помните о приоритете арифметических операций PEP 8 (мы вводим and, or).

Почитайте руководство по написанию кода на Python:

PEP 8 (на английском языке),

PEP 8 (на русском языке).

Обратите внимание на список встроенных функций (официальная документация).

Прочитайте описание конструкции for-else.

Прочитайте описание enumerate для исключения конструкций вида for i in range(len(array)).

# Модуль 11

## 11.2 Ввод вещественного числа. Функции float и round

**Задача 1. Ставки на спорт**

Нас наняла букмекерская контора, где проводятся ставки на конный спорт. Напишите программу расчёта потенциального выигрыша игрока. Для этого вводится его ставка в рублях и коэффициент (вещественное число), а выводится их произведение в качестве потенциального выигрыша.

Пример:

Сколько ставим? 1234

Какой коэффициент? 1.716

Потенциальный выигрыш: 2117.544

Усложнение задачи: сделайте так, чтобы после точки выводилось не больше двух цифр.

**Задача 2. День рождения**

В честь дня рождения сына отец не придумал ничего лучше, кроме как подарить денег. Зато он придумал хоть и странную, но интересную формулу, по которой высчитывается сколько он подарит: возраст сына умножается на 1.5 и на его температуру тела. Остаётся только позавидовать такой фантазии.

Напишите программу, которая запрашивает у пользователя возраст (целое число) и температуру тела (вещественное число) и в результате выводит сколько отец подарит сыну денег на день рождения.

**Задача 3. Индекс массы тела**

Алексей работает диетологом в частной клинике, каждый день он принимает пациентов разных возрастов и с разными показателями роста (в метрах) и веса (в кг). Для каждого человека ему нужно считать индекс массы тела - это вес поделить на рост в квадрате. По государственным стандартам индекс округляется до двух знаков после точки. Затем по этому индексу определяется, всё ли в порядке у человека с массой тела: если до 18.5, то недобор; до 25 - всё нормально, до 30 уже идёт избыток, а после 30 - ожирение. Напишите такую программу для Алексея.

## 11.3 Приведение типов между int и float

**Задача 1. Космические рейнджеры**

В далеком (а может и не очень) будущем на некоторой планете можно купить космический корабль за пол-кредита (CR). Один CR это 2200 чатлов. Причем чатлы неделимы и являются всегда целым числом. Напишите простую программу-конвертор валют. В программу вводится количество чатлов, а она сообщает сколько это CR и сколько кораблей можно купить на эту сумму.

Пример 1:

Сколько чатлов? 3150

Это 1.4318181818181819 CR

Можно купить кораблей: 2

Пример 2:

Сколько чатлов? 4400

Это 2.0 CR

Можно купить кораблей: 4

**Задача 2. Компьютерное зрение**

Вы участвуете в разработке робота, который играет в шахматы на реальной, физической шахматной доске размером 0.8 х 0.8 метра. Робот смотрит камерой на доску и видит расположение фигур в вещественных числах с очень высокой точностью.

Напишите программу, в которую вводятся вещественные координаты шахматной фигуры, а программа определяет в какой клетке доски находится эта фигура. Каждая клетка доски имеет размер 10х10 см и целочисленные координаты, состоящие из двух чисел. Например левая верхняя клетка имеет целые координаты (0, 0), справа от неё клетка (1, 0) а снизу (0, 1). Также обеспечьте контроль ввода: нельзя выходить за границы доски.

Пример:

Введите местоположение фигуры

По горизонтали: 0.731

По вертикали: 0.149

Фигура находится в клетке (7, 1)

Введите местоположение фигуры

По горизонтали: 0.833

По вертикали: -0.132

Клетки с такой координатой не существует

**Задача 3. Точность и аккуратность**

Робот из задачи “Компьютерное зрение” правильно определяет на какой клетке стоят фигуры. Но вот беда, часто по вине соперника-человека фигуры стоят на доске не ровно по центру клетки, а со смещением. Если во время игры такое смещение станет слишком велико, то станет непонятно в какой клетке стояла фигура. Чтобы избежать этой ситуации нужно чтобы робот поправлял фигуры на доске, выставляя их по центру клетки. Модифицируйте программу “Компьютерное зрение” так, чтобы она выдавала не только номера клетки, в которой находится фигура но и две вещественных поправки: на сколько нужно подвинуть фигуру по горизонтали и вертикали для того чтобы она оказалась по центру своей клетки.

Пример:

Введите местоположение фигуры

По горизонтали: 0.731

По вертикали: 0.167

Фигура находится в клетке (7, 1)

Поправьте положение фигуры на (0.019, -0.017)

## 11.4 Математические функции. Работа с модулем math

**Задача 1. Герон**

Существует, так называемая, формула Герона, позволяющая вычислить площадь треугольника, зная длины его сторон.

S= √ (p \* (p - a)\*(p - b)\*(p - c)) ,где

S - площадь;

p - полупериметр треугольника (a+b+c)/2;

a,b,c - длины сторон треугольника.

Напишите программу, которая принимает на вход длины сторон треугольника и выводит его площадь

**Задача 2. Игра**

Вам предстоит создать 2D-игру (вид сверху, игрок двигается в двух плоскостях).

Начнём с управления персонажем: получаем от игрока пройденное расстояние и угол, по которому двигался персонаж. Зная эту информацию, нужно изменить текущие координаты (0, 0) на новые (х, у). Чтобы это сделать, требуется узнать, какое расстояние персонаж преодолеет по горизонтали (по оси Х, x = расстояние × косинус угла) и по вертикали (по оси У, y = расстояние × синус угла).

Напишите программу, которая получает на вход расстояние и угол поворота. Выведите координаты нового местоположения персонажа.

**Задача 3. Мега-калькулятор**

Кеша учится в третьем классе, и уже умеет программировать на питоне. Как и многие его одноклассники, он очень любит сразу применять все полученные знания на практике. Вчера Кеша узнал про модуль math и его основные возможности, поэтому решил написать мега-калькулятор, который бы применял сразу все функции к введенному пользователем числу. Чтобы ничего не забыть, он пользуется шпаргалкой, которую прикрепили к уроку

Напишите программу, которая получает от пользователя вещественное число, по очереди применяет к нему функции модуля Math и выводит результат:

округляет вниз

округляет вверх

берет модуль числа

извлекает квадратный корень

возводит экспоненту в степень, равную числу

считает натуральный логарифм числа

считает логарифм числа по основанию 2

считает логарифм числа по основанию 10

считает синус и косинус числа

И так как Кеша самый умный в классе, он решил попробовать посчитать факториал числа. Для этого ему пришлось придумать и реализовать контроль ввода: факториал вычисляется, только если введенное число было натуральным.

## 11.6 Практическая работа

**Задача 1. Конвертация**

При оплате покупок картой за рубежом банки делают конвертацию через промежуточную валюту. Например, если оплачивать отечественной картой товар в евро, то сначала эта сумма конвертируется в доллары, а потом — в рубли.

Напишите программу, которая получает на вход стоимость покупки в евро, а затем выводит ответ в рублях. Представим, что мы живём в альтернативной реальности, где 1 евро = 1.25 доллара, а 1 доллар = 60.87 руб.

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 2. Грубая математика**

В одном центре математического анализа работал практикант, который писал программы для расчёта функций. Однажды он очень устал и неправильно понял техническое задание, поэтому функции стали грубо рассчитываться.

Его программа работает следующим образом: вводится последовательность из N вещественных чисел, при этом положительные числа округляются вверх, а отрицательные — вниз.

Напишите программу, которая выводит натуральный логарифм от числа, если оно положительное, и экспоненту в степени числа, если оно отрицательное.

Пример:

Введите кол-во чисел: 3

Введите число: 1.3

x = 2 log(x) = 0.6931471805599453

Введите число: -2.1

x = -3 exp(x) = 0.049787068367863944

Введите число: -5.9

x = -6 exp(x) = 0.0024787521766663585

**Что оценивается**

Результат вывода соответствует условию, применяются корректные функции, округление осуществляется в нужную сторону.

Формат вывода соответствует примеру.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 3. Аналог Steam**

Вы пишете программу-инсталлятор для компьютерной игры. Пока инсталлятор скачивает обновление, для пользователя необходимо отображать количество скачанных процентов, чтобы он понимал, успеет ли заварить чай, прежде чем завершится процесс. Каждое обновление игры требует разного количества мегабайт, при этом у разных игроков разная скорость интернет-соединения.

Напишите программу, принимающую на вход размер файла обновления в мегабайтах и скорость интернет-соединения в мегабайтах в секунду. Для каждой секунды программа должна рассчитывать и выводить на экран процент скачанного объёма до тех пор, пока скачивание не завершится. В конце программа должна показать, сколько секунд заняло скачивание обновления. Обеспечьте контроль ввода.

Пример:

Укажите размер файла для скачивания: 123

Какова скорость вашего соединения: 27

Прошло 1 сек. Скачано 27 из 123 Мб (22%)

Прошло 2 сек. Скачано 54 из 123 Мб (44%)

Прошло 3 сек. Скачано 81 из 123 Мб (66%)

Прошло 4 сек. Скачано 108 из 123 Мб (88%)

Прошло 5 сек. Скачано 123 из 123 Мб (100%)

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 4. Первая цифра**

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

**Что оценивается**

Результат вывода соответствует условию.

В вычислениях не используется for, if и работа со строками.

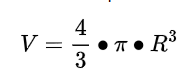
Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 5. Вот это объемы!**

Для курсовой работы по физике Андрею нужно сравнить объёмы двух планет: Земли и теоретически возможной для нашей вселенной планеты. Андрей хорошо разбирается в формулах, но плохо считает. Объём Земли ему известен — это 1.08321 \* 10 \*\* 12 км3.

Объём теоретически возможной планеты ему нужно посчитать. У него есть формула:



В ней V — это объём, π — число пи, а R — радиус планеты.

Напишите программу, которая получает на вход радиус случайной планеты и выводит на экран, во сколько раз планета Земля меньше или больше теоретически возможной планеты по объёму. Ответ округлите до трёх знаков после запятой.

Пример 1:

Введите радиус случайной планеты: 3389.5

Объём планеты Земля больше в 6.641 раз

Пример 2:

Введите радиус теоретически возможной планеты: 7000

Объём планеты Земля меньше в (1/0.754) = 1.326 раз

**Что оценивается**

Результат вывода соответствует условию.

Формат вывода соответствует примеру.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 6. Ход конём**

В рамках разработки шахматного ИИ стоит новая задача: по заданным вещественным координатам коня и точки программа должна определить, может ли конь ходить в эту точку. Используйте как можно меньше конструкций if и логических операторов. Обеспечьте контроль ввода.

Пример:

Введите местоположение коня:

0.071

0.118

Введите местоположение точки на доске:

0.213

0.068

Конь в клетке (0, 1). Точка в клетке (2, 0).

Да, конь может ходить в эту точку.

**Что оценивается**

Результат вывода соответствует условию.

Формат вывода соответствует примеру.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 7. За что?**

Вы встретились со старым другом, который тоже изучает программирование, но в другом учебном заведении. За чашкой кофе он пожаловался, что их сумасбродный препод дал задание написать программу, которая из двух введённых чисел определяет наибольшее, не используя при этом условные операторы, циклы и встроенные функции вроде max/min/sorted. Радуясь, что на вашем курсе такого не требуют, вы всё-таки решаете помочь другу. Напишите для него программу.

Пример:

Введите первое число: 10

Введите второе число: 5

Наибольшее число: 10

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Советы и рекомендации**

Рассмотрите разность суммы и разности чисел, сумму разности и суммы чисел.

При необходимости можете использовать функцию abs(), позволяющую взять модуль числа.

**Что оценивается в работе**

Использование именованных индексов, не просто i (видео 7.2).

Введены правильные числа: без дополнительных действий пользователя, без использования +1 (видео 7.4).

Работа сдана через Fork или корректно оформленный repl.it.

Правильно оформлен input — без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками: “print ()”,“input ()” — неверно, “print()” — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные имеют корректные названия, в качестве имён не используются имена встроенных функций (официальная документация со списком встроенных функций).

**Советы и рекомендации**

Выводите значение счётчика при отладке программы (видео 8.4).

Будьте аккуратны при изменении чисел внутри цикла, которые его задают (видео 8.5).

Помните о приоритете арифметических операций PEP 8 (мы вводим and, or).

Прочитайте руководство по написанию кода на Python PEP 8

Обратите внимание на список встроенных функций (официальная документация).

Прочитайте описание конструкции for-else.

Прочитайте описание enumerate для исключения конструкций вида for i in range(len(array)).

# Модуль 12

## 12.2 Функции и их вызов

**Задача 1. Робот**

В одном отеле для эксперимента на вход хотят поставить робота, который будет спрашивать у прохожих, не желают ли они зайти. При ответе «Да» робот должен приветствовать человека и пустить в отель. Для робота написали вот такую программу с использованием функции:

Def greeting()

print('Привет!')

print('Добро пожаловать!')

while True:

a = input('Зайдёте? Да/Нет: ')

if a = 'Да':

greeting

print('Следующий.\n')

Однако программист очень торопился и допустил несколько ошибок.

Скопируйте программу в реплит, найдите и исправьте ошибки. Убедитесь, что программа работает корректно.

Пример результата:

Зайдёте? Да/Нет: Да

Привет!

Добро пожаловать!

Следующий.

Зайдёте? Да/Нет: Да

Привет!

Добро пожаловать!

Следующий.

Зайдёте? Да/Нет: Нет

Следующий.

Зайдёте? Да/Нет: ...

**Задача 2. Провизия**

Одна государственная компания поставляет еду на разные труднодоступные базы (полярные, горные и так далее) в разных уголках страны. В компании для удобства расчёта количества еды была реализована такая программа:

print("Сколько мешков рыбы и мяса?")

a = int(input())

b = int(input())

print("Всего", a+b, "шт.")

print("Сколько буханок белого и чёрного хлеба?")

a = int(input())

b = int(input())

print("Всего", a+b, "шт.")

print("Сколько вёдер воды и молока?")

a = int(input())

b = int(input())

print("Всего", a+b, "шт.")

Скопируйте программу в реплит и модифицируйте её: напишите функцию, которая заменит повторяющийся код внутри основной программы.

**Задача 3. Почта**

Василий пришёл получать посылку на почту. Разумеется, в почтовом извещении ему нужно было написать свои фамилию, имя и адрес проживания, чтобы кто-нибудь не получил посылку за него, например члены его семьи (а им бы хотелось!).

Напишите функцию для вывода фамилии, имени и адреса для конкретного члена семьи. Выведите информацию о нём три раза (без цикла).

Пример результата:

Фамилия: Иванов

Имя: Василий

Улица: Пушкина

Дом: 32

Фамилия: Иванов

Имя: Василий

Улица: Пушкина

Дом: 32

Фамилия: Иванов

Имя: Василий

Улица: Пушкина

Дом: 32

## 12.3 Функции с одним параметром

**Задача 1. Вода**

Одна бутылка воды «КлирВотер» от производителя «ВодЗавод» в разных магазинах стоит по-разному.

Напишите программу, которая три раза вызывает функцию aboutWater, передаёт в неё один аргумент — цену на воду и выводит на экран название воды, производителя и цену.

Пример:

Название: КлирВотер

Производитель: ВодЗавод

Цена: 25

Название: КлирВотер

Производитель: ВодЗавод

Цена: 30

Название: КлирВотер

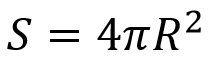
Производитель: ВодЗавод

Цена: 40

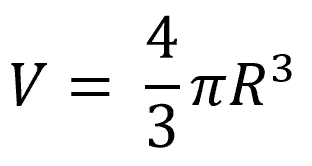
**Задача 2. Вот это объёмы 2**

Андрей продолжает писать курсовую работу по физике, и теперь ему нужно находить не только объём планеты, но и её площадь. Для этого он использует две такие формулы:

Формула для площади сферы:



Формула для объёма шара:



Так как в самом курсовом проекте эти формулы пригодятся ещё не раз, Андрей решил поступить рационально и просто написать функцию для каждой формулы.

Напишите программу, которая на вход получает от пользователя радиус планеты (вещественное число) и вызывает функции sphereArea и sphereVolume. Реализуйте эти функции: первая считает и выводит на экран площадь сферы, вторая — объём шара.

**Задача 3. Простые числа**

Пользователь вводит число N — количество чисел в последовательности. Напишите функцию isPrime, которая проверяет, является ли число простым и выводит ответ в консоль.

## 12.4 Функции с несколькими параметрами

**Задача 1. Среднее арифметическое**

Программа получает от пользователя два числа — a и b. Реализуйте функцию, которая принимает на вход числа a и b, считает и выводит в консоль среднее арифметическое всех чисел из отрезка [a; b]. Обеспечьте контроль ввода: не забывайте, что а всегда должно быть меньше, чем b.

Пример:

Введите левую границу: 3

Введите правую границу: 8

Среднее: 5.5

Усложнение: сделайте это без использования циклов.

**Задача 2. Почта 2**

На почте немного поменялись правила: теперь в почтовом извещении нужно указывать фамилию, имя, страну проживания, город, улицу, номер дома и номер квартиры.

Реализуйте функцию, которая получает все эти данные и выводит на экран. В программе вызовите функцию три раза с разными значениями аргументов.

Подсказка: семь аргументов.

**Задача 3. GPS-навигатор 2.0**

Нам поручили усовершенствовать GPS-навигатор, добавив в него новую фишку. Теперь пользователь может не только смотреть расстояние от себя до объекта, но и задавать в навигаторе две произвольные точки, после чего на экран ему выводится расстояние между ними. Для этого пользователь вводит четыре действительных числа x1, y1, x2, y2 — это как раз координаты этих двух точек.

Напишите программу, где у пользователя спрашивается, чего он хочет — найти расстояние от себя до точки или найти расстояние между двумя произвольными точками, после чего запрашиваются необходимые координаты точек и выводится ответ на экран.

## 12.6 Практическая работа

**Задача 1. Сумма чисел**

Напишите функцию summa\_n, которая принимает одно целое положительное число N и выводит сумму всех чисел от 1 до N включительно.

Пример работы программы:

Введите число: 5

Я знаю, что сумма чисел от 1 до 5 равна 15

Что оценивается

Результат вывода соответствует условию.

В input содержится корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 2. Функция в функции**

Евгений проходит специальный тест по программированию. Всё шло хорошо, пока герой не наткнулся на тему «Функции». Задание звучит так:

Основная ветка программы, не считая заголовков функций, состоит из одной строки кода. Это вызов функции test(). В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция positive(), тело которой содержит команду вывода на экран слова «Положительное». Если число отрицательное, то вызывается функция negative(), её тело содержит выражение вывода на экран слова «Отрицательное».

Помогите Евгению и реализуйте такую программу.

Что оценивается

Результат вывода соответствует условию.

В input содержится корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 3. Апгрейд калькулятора**

Степан использует калькулятор для расчёта суммы и разности чисел, но на работе ему требуются не только обычные арифметические действия. Он ничего не хочет делать вручную, поэтому решил немного расширить функционал калькулятора.

Напишите программу, запрашивающую у пользователя число и действие, которое нужно сделать с числом: вывести сумму его цифр, максимальную или минимальную цифру. Каждое действие оформите в виде отдельной функции, а основную программу зациклите.

Запрошенные числа должны передаваться в функции суммы, максимума и минимума при помощи аргументов.

Что оценивается

Результат вывода соответствует условию.

В input содержится корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

Нет обращения к глобальным переменным снаружи функции.

Числа, с которыми должны работать функции, передаются через аргументы.

Советы и рекомендации

Не забывайте, что параметры функции и переменные снаружи могут иметь разные названия.

При необходимости пересмотрите видео по работе с аргументами либо ознакомьтесь с дополнительными источниками информации по этой теме, например с этой статьёй:

Функции и их аргументы в Python 3

**Задача 4. Число наоборот**

Вводится последовательность чисел, оканчивающаяся нулём. Реализуйте функцию, которая принимает в качестве аргумента каждое число, переворачивает его и выводит на экран.

Пример:

Введите число: 1234

Число наоборот: 4321

Введите число: 1000

Число наоборот: 0001

Введите число: 0

Программа завершена!

Дополнительно: добейтесь такого вывода чисел, в начале которых идут нули.

Пример:

Введите число: 1230

Число наоборот: 321

Ноль, который мы убрали, называется ведущим.

Что оценивается

Результат вывода соответствует условию.

Формат вывода соответствует примеру.

В input содержится корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 5. Текстовый редактор**

Продолжаем разрабатывать новый текстовый редактор. В этот раз нам поручили написать для него код, который считает, сколько раз в тексте встречается любая выбранная буква или цифра (а не только буквы Ы, как раньше).

Напишите функцию count\_letters(), которая принимает на вход текст и подсчитывает, какое в нём количество цифр K и букв N. Функция должна вывести на экран информацию о найденных буквах и цифрах в определённом формате.

Пример:

Введите текст: 100 лет в обед

Какую цифру ищем? 0

Какую букву ищем? Л

Количество цифр 0: 2

Количество букв Л: 1

**Что оценивается**

Результат вывода соответствует условию.

В input содержится корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 6. НОД**

Напишите функцию, вычисляющую наибольший общий делитель двух чисел.

**Что оценивается**

Результат вывода соответствует условию.

В input содержится корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 7. Недоделка**

Вы пришли на работу в компанию по разработке игр, целевая аудитория — дети и их родители. У предыдущего программиста было задание сделать две игры в одном приложении, чтобы пользователь мог выбирать одну из них. Однако программист, на место которого вы пришли, перед увольнением не успел выполнить эту задачу и оставил только небольшой шаблон проекта. Используя этот шаблон, реализуйте игры «Камень, ножницы, бумага» и «Угадай число».

Правила игры «Камень, ножницы, бумага»: программа запрашивает у пользователя строку и выводит, победил он или проиграл. Камень бьёт ножницы, ножницы режут бумагу, бумага кроет камень.

Правила игры «Угадай число»: программа запрашивает у пользователя число до тех пор, пока он не отгадает загаданное.

def rock\_paper\_scissors():

# Здесь будет игра «Камень, ножницы, бумага»

def guess\_the\_number():

# Здесь будет игра «Угадай число»

def mainMenu():

# Здесь главное меню игры

mainMenu():

pass

**Что оценивается**

Игры работают корректно.

В input содержится корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Что оценивается (общее)**

Использование аргументов функции для передачи данных (при помощи input данные вводятся СНАРУЖИ функции и при помощи аргументов передаются в функцию).

Использование именованных индексов, не просто i (видео 7.2).

Ввод правильных чисел, без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).

Отправка работы через Fork или корректно оформленный repl.it.

Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).

Переменные со значащими именами, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками: print (),input () — неверно, print() — верно.

Правильное оформление блоков if-elif-else, одинаковые отступы во всех блоках одного уровня.

Корректные названия переменных; в качестве имён не используются имена встроенных функций (список встроенных функций из официальной документации).

**Советы и рекомендации**

Выводите значение счётчика при отладке программы (видео 8.4).

Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).

Помните о приоритете арифметических операций PEP 8 (мы вводим and, or).

Почитайте руководство по написанию кода на Python:

PEP 8 (на английском языке),

PEP 8 (на русском языке).

Обратите внимание на список встроенных функций из официальной документации.

Прочитайте описание конструкции for-else.

Прочитайте описание enumerate для исключения конструкций вида for i in range(len(array)).

# Модуль 13

## 13.2 Возврат значений из функций. Оператор return

**Задача 1. Сумма чисел 2**

Пользователь вводит число N. Напишите функцию summa\_n, которая принимает одно целое положительное число N и находит сумму всех чисел от 1 до N включительно. Функция вызывается два раза: сначала от числа N, а затем от полученной суммы.

Пример работы программы:

Введите число: 5

Сумма от 1 до 5 = 15

Сумма от 1 до 15 = 120

**Задача 2. «Назад в будущее»**

Вы — один из разработчиков языка программирования Python, и вы пишете специальный математический модуль, который можно было бы просто подключить внутри программы и облегчить жизнь всем программистам.

Реализуйте функцию gcd, которая получает два параметра — два числа — и возвращает наибольший общий делитель этих двух чисел.

Пример работы программы:

Введите первое число: 6

Введите второе число: 10

НОД = 2

**Задача 3. Приоритет задач**

В одном дата-центре ресурсы распределены так, что сначала обрабатываются крупные задачи, а затем уже идут небольшие. Каждая из этих задач, по сути, просто огромный поток цифр. Ваша задача, как программиста этого центра, написать программу, которая поможет определять, какую из задач нужно решать в первую очередь.

Вводится последовательность из N чисел. Нужно определить номер числа, у которого больше всего цифр, и вывести на экран соответствующее сообщение. Если число отрицательное, то считать его за 0. Для подсчёта количества цифр реализуйте функцию numeral\_count.

Пример работы программы:

Введите кол-во задач: 4

Введите число: 6

Введите число: 14

Введите число: 1

Введите число: 13434

Первая задача на обработку: 13434

## 13.3 Представление вещественных чисел в программе

**Задача 1. Возможности компьютера**

В одной IT-компании тестируют возможности различных языков программирования, компиляторов и, конечно же, компьютеров. Компания дала вам задачу понять, какое самое маленькое возможное число можно получить путём постоянного деления числа на 2. Изначально число равно единице. Также, помимо самого числа, компания просит вывести количество делений. Реализуйте такую программу.

**Задача 2. Тестирование**

Команде программистов отдали на тестирование новую модель суперкомпьютера. Для начала программисты решили проверить, как у компьютера обстоят дела с вычислениями вещественных чисел. Разработчики компьютера предупредили, что на входе он работает только с экспоненциальной формой числа.

Пользователь вводит число N в экспоненциальной форме, где мантисса всегда равна числу от 1 до 9, а порядок меньше нуля. Также есть переменная Х, которая изначально равна единице. Посчитайте, сколько раз нужно прибавить N к Х, чтобы оно перевалило за двойку.

Дополнительно: обеспечьте контроль ввода.

Пример 1:

Введите число в эксп. форме: 1e-3

Кол-во прибавлений: 1001

Пример 2:

Введите число в эксп. форме: 5.02e-1

Кол-во прибавлений: 2

**Задача 3. Урок информатики**

На одном из уроков информатики учитель объяснял тему «Числа с плавающей точкой», но несколько учеников никак не могли понять, почему эта точка «плавает» и как вообще выглядят числа в такой форме. Для наглядности учитель написал программу, которая берёт число больше десяти и выводит его в формате плавающей точки.

Пользователь вводит положительное число x (x > 10). Напишите функцию, которая выводит его в формате плавающей точки, то есть x = a \*10 \*\* b, где 1 ≤ a < 10.

Пример 1:

Введите число: 16

Формат плавающей точки: x = 1.6 \* 10 \*\* 1

Пример 2:

Введите число: 92345

Формат плавающей точки: x = 9.2345 \* 10 \*\* 4

## 13.4 Особенности работы с вещественными числами

**Задача 1. Опять налоги**

У правительства одной из стран есть бухгалтерская программа, которая суммирует налоги её граждан, компаний плюс НДС с товаров. Страна развивалась, суммарные налоги увеличивались, и бухгалтеры заметили, что после добавления к общей сумме налогов некого НДС от какого-то продукта общая сумма перестала меняться…

Нужно помочь бухгалтерам: напишите функцию, на вход которой подаются два числа — общая сумма налога tax и новый налог new\_tax, который нужно добавить к общей сумме. Функция должна проверять, возможно ли сложить эти два числа или нет, и выводить соответствующее сообщение о том, увеличится ли бюджет или нет.

Пример 1:

Введите бюджет страны: 1.23e2

Новые поступления (налог): 1.2e1

Результат: Бюджет увеличится

Пример 2:

Введите бюджет страны: 1.231221200034e12

Новые поступления (налог): 1.2e-4

Результат: Бюджет не изменится

**Задача 2. Сравнение**

Так как в Python операции с вещественными числами могут давать неожиданные результаты (в частности, 0.1 + 0.2 не будет в точности равняться 0.3), стоит задача с этим как-то справляться.

Напишите функцию eqv, которая принимает три числа и затем сравнивает сумму первых двух чисел с третьим с определённой степенью точности: до 15-го знака после точки. Если равенство выполняется, то функция возвращает True, иначе возвращает False.

Пример 1:

Введите первое число: 1.1

Введите второе число: 2.2

Введите третье число: 3.3

True

Пример 2:

Введите первое число: 1e-14

Введите второе число: 1e-14

Введите третье число: 3e-14

False

## 13.6 Практическая работа

**Задача 1. Урок информатики 2**

В прошлый раз учитель написал программу, которая выводит числа в формате плавающей точки, однако он вспомнил, что не учёл одну важную вещь: числа-то могут идти от нуля.

Задано положительное число x (x > 0). Ваша задача — преобразовать его в формат плавающей точки, то есть x = a \* 10^b, где 1 ≤ a < 10. Обратите внимание, что x теперь больше нуля, а не больше единицы. Обеспечьте контроль ввода.

Пример 1:

Введите число: 92345

Формат плавающей точки: x = 9.2345 \* 10 \*\* 4

Пример 2:

Введите число: 0.0012

Формат плавающей точки: x = 1.2 \* 10 \*\* -3

**Что оценивается**

Результат вывода соответствует условию.

Обеспечен контроль ввода и обрабатываемые числа > 0.

Input содержит корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 2. Функция максимума**

Юра пишет различные полезные функции для Python, чтобы остальным программистам стало проще работать. Он захотел написать функцию, которая будет находить максимум из перечисленных чисел. Функция для нахождения максимума из двух чисел у него уже есть. Юра задумался: может быть, её можно как-то использовать для нахождения максимума уже от трёх чисел?

Помогите Юре написать программу, которая находит максимум из трёх чисел. Для этого используйте только функцию нахождения максимума из двух чисел.

По итогу в программе должны быть реализованы две функции:

maximum\_of\_two — функция принимает два числа и возвращает одно (наибольшее из двух);

maximum\_of\_three — функция принимает три числа и возвращает одно (наибольшее из трёх); при этом она должна использовать для сравнений первую функцию maximum\_of\_two.

Что оценивается

Результат вывода соответствует условию.

Найден корректный максимум трёх чисел.

Input содержит корректное приглашение для ввода.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 3. Число наоборот 2**

Пользователь вводит два числа: N и K. Напишите программу, которая заменяет каждое число на число, которое получается из исходного записью его цифр в обратном порядке, затем складывает их, снова переворачивает и выводит ответ на экран.

Пример:

Введите первое число: 102

Введите второе число: 123

Первое число наоборот: 201

Второе число наоборот: 321

Сумма: 522

Сумма наоборот: 225

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 4. Недоделка 2**

Вы всё так же работаете в конторе по разработке игр и смотрите различные программы прошлого горе-программиста. В одной из игр для детей, связанной с мультяшной работой с числами, вам нужно было написать код согласно следующим условиям: программа получает на вход два числа; в первом числе должно быть не менее трёх цифр, во втором — не менее четырёх, иначе программа выдаёт ошибку. Если всё нормально, то в каждом числе первая и последняя цифры меняются местами, а затем выводится их сумма.

И тут вы натыкаетесь на программу, которая была написана предыдущим программистом и которая как раз решает такую задачу. Однако старший программист попросил вас немного переписать этот код, чтобы он не выглядел так ужасно. Да и вам самим становится, мягко говоря, не по себе от него.

Постарайтесь разделить логику кода на три отдельные логические части (функции):

count\_numbers — получает число и возвращает количество цифр в числе;

change\_number — получает число, меняет в нём местами первую и последнюю цифры и возвращает изменённое число;

main — функция ничего не получает на вход, внутри она запрашивает нужные данные от пользователя, выполняет дополнительные проверки и вызывает функции 1 и 2 для выполнения задачи (проверки и изменения двух чисел).

Разбейте приведённую ниже программу на функции. Повторений кода должно быть как можно меньше. Также сделайте, чтобы в основной части программы был только ввод чисел, затем изменённые числа и вывод их суммы.

first\_n = int(input("Введите первое число: "))

first\_num\_count = 0

temp = first\_n

while temp > 0:

first\_num\_count += 1

temp = temp // 10

if first\_num\_count < 3:

print("В первом числе меньше трёх цифр.")

else:

last\_digit = first\_n % 10

first\_digit = first\_n // 10 \*\* (first\_num\_count - 1)

between\_digits = first\_n % 10 \*\* (first\_num\_count - 1) // 10

first\_n = last\_digit \* 10 \*\* (first\_num\_count - 1) + between\_digits \* 10 + first\_digit

print('Изменённое первое число:', first\_n)

second\_n = int(input("\nВведите второе число: "))

second\_num\_count = 0

temp = second\_n

while temp > 0:

second\_num\_count += 1

temp = temp // 10

if second\_num\_count < 4:

print("Во втором числе меньше четырёх цифр.")

else:

last\_digit = second\_n % 10

first\_digit = second\_n // 10 \*\* (second\_num\_count - 1)

between\_digits = second\_n % 10 \*\* (second\_num\_count - 1) // 10

second\_n = last\_digit \* 10 \*\* (second\_num\_count - 1) + between\_digits \* 10 + first\_digit

print('Изменённое второе число:', second\_n)

print('\nСумма чисел:', first\_n + second\_n)

**Что оценивается**

Программа разбита на несколько функций.

Выполнены условия по организации основного тела программы.

**Задача 5. Маятник**

Известно, что амплитуда качающегося маятника с каждым разом затухает на 8,4% от амплитуды предыдущего колебания. Если качнуть маятник, то, строго говоря, он не остановится никогда, просто амплитуда будет постоянно уменьшаться до тех пор, пока мы не сочтём такой маятник остановившимся. Напишите программу, определяющую, сколько раз качнётся маятник, прежде чем он, по нашему мнению, остановится.

Программа получает на вход начальную амплитуду колебания в сантиметрах и конечную амплитуду колебаний, которая считается остановкой маятника. Обеспечьте контроль ввода.

Пример:

Введите начальную амплитуду: 1

Введите амплитуду остановки: 0.1

Маятник считается остановившимся через 27 колебаний

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Задача 6. Яйца**

В рамках программы колонизации Марса компания «Спейс Инжиниринг» вывела особую породу черепах, которые, по задумке, должны размножаться, откладывая яйца в марсианском грунте. Откладывать яйца слишком близко к поверхности опасно из-за радиации, а слишком глубоко — из-за давления грунта и недостатка кислорода. Вообще, факторов очень много, но специалисты проделали большую работу и предположили, что уровень опасности для черепашьих яиц рассчитывается по формуле: D = x^3 − 3x^2 − 12x + 10, где x — глубина кладки в метрах, а D — уровень опасности в условных единицах. Для тестирования гипотезы нужно взять пробу грунта на безопасной, согласно формуле, глубине.

Напишите программу, находящую такое значение глубины х, при котором уровень опасности как можно более близок к нулю. На вход программе подаётся максимально допустимое отклонение уровня опасности от нуля, а программа должна рассчитать приблизительное значение х, удовлетворяющее этому отклонению. Известно, что глубина точно больше нуля и меньше четырёх метров. Обеспечьте контроль ввода.

Пример:

Введите максимально допустимый уровень опасности: 0.01

Приблизительная глубина безопасной кладки: 0.732421875 м

**Что оценивается**

Результат вывода соответствует условию.

Input содержит корректное приглашение для ввода.

Формат вывода соответствует примеру.

Вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием).

**Что оценивается (общее)**

Использование именованных индексов, не просто i (видео 7.2).

Использование правильных чисел, без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).

Практическая работа сдана через Fork или корректно оформленный repl.it.

Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).

Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).

Пробелы после запятых, пробелы при бинарных операциях.

Отсутствие пробелов после имён функций и перед скобками: print (), input () — неверно, print() — верно.

Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

Переменные имеют корректные названия, в качестве имён не используются имена встроенных функций (список встроенных функций — официальная документация).

**Советы и рекомендации**

Выводите значение счётчика при отладке программы (видео 8.4).

Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).

Помните о приоритете арифметических операций PEP 8 (мы вводим and, or).

Почитайте руководство по написанию кода на Python:

PEP 8 (на английском языке),

PEP 8 (на русском языке).

Обратите внимание на список встроенных функций (официальная документация).

Прочитайте описание конструкции for-else.

Прочитайте описание enumerate для исключения конструкций вида for i in range(len(array)).