Практические задания по программированию на языке **Python**

**Уровень сложности «Лёгкий»**

1. Преобразуйте число введённое пользователем из десятичной системы в двоичную, восьмеричную и шестнадцатеричную и выведите на экран.
2. Дана строка '123456789'. Используя операции индексирования и среза выведите на экран третий и пятый символы, а также подстроку '567'. Реализуйте вывод двумя способами: используя только положительные индексы и только отрицательные.
3. Измените строку 'кот', записав ее символы в обратном порядке. Выведите результат на экран.
4. Выведите первый и последний элемент списка.
5. Напишите функцию sum\_range(start, end), которая суммирует все целые числа от значения start до величины end включительно. Если пользователь задаст первое число большее чем второе, просто поменяйте их местами.

**Уровень сложности «Средний»**

1. **Найти площадь и периметр прямоугольного треугольника по двум заданным катетам(**Площадь прямоугольного треугольника равна половине площади прямоугольника, стороны которого равны длинам катетов.**)**
2. **Перевернуть число.** Вводится целое число. Вывести число, обратное введенному по порядку составляющих его цифр. Например, введено 3425, надо вывести 5243.

# Проверить уникальность элементов списка. В списке чисел проверить, все ли элементы являются уникальными, то есть встречается ли каждое число только один раз.

1. Чтобы проверить понимание параметров и область их видимости, Николай создал 3 функции (представлены ниже).

Попытайтесь предугадать, как поведет себя каждая из них при запуске (возникнут ли ошибки, что возвратится).

**def** **func1**():

param = 4

**def** **inner**():

param += 1

**return** param

**def** **func2**():

param = 4

**def** **inner**(var):

var += 1

inner(param)

**return** param

**def** **func3**():

param = 4

**def** **inner**(var):

var += 1

**return** var

param = inner(param)

**return** param

**5.** Создайте функцию three\_args(), которая принимает 1, 2 или 3 строго ключевых параметра.

В результате ее работы на печать в консоль выводятся значения переданных переменных, но только если они не равны **None**.

Получим, например, следующее сообщение: «Переданы аргументы: var1 = 2, var3 = 10».

**Уровень сложности «Сложный»**

**1.** Антон попал в коллизию: его функция time\_now() работает очень странно. Казалось бы, задача простая: показать текущее время с сообщением. Тем не менее, время не меняется. Код предоставлен ниже с примерами. Постарайтесь решить проблему незадачливого программиста.

from datetime import datetime

from time import sleep

deftime\_now(msg, \*, dt=datetime.now()): print(msg, dt)

# Тесты

time\_now('Сейчас такое время:') sleep(1)

time\_now('Прошла секунда: ')

sleep(1)

time\_now('Ничего не понимаю…')

**Результат выполнения**

Сейчас такое время: 2021-03-14 15:48:55.117455

Прошла секунда: 2021-03-14 15:48:55.117455

Ничего не понимаю… 2021-03-14 15:48:55.117455

**2.** Чтобы лучше разобраться в типах параметров функций Инна создала inspect\_function(), которая в качестве аргумента принимает другую функцию (главное, не встроенную, built-in).

В результате работы она выводит следующие данные: название анализируемой функции, наименование всех принимаемых ею параметров и их типы (позиционные, ключевые и т.п.).

Попробуйте повторить результат девушки.

**3.** На входе имеем список строк разной длины.

Необходимо написать функцию all\_eq(lst), которая вернет новый список из строк одинаковой длины.

Длину итоговой строки определяем исходя из самой большой из них.

Если конкретная строка короче самой длинной, дополнить ее нижними подчеркиваниями с правого края до требуемого количества символов.

Расположение элементов начального списка не менять.

**4.** Дмитрий считает, что когда текст пишут в скобках (как вот тут, например), его читать не нужно.

Вот и надумал он существенно укоротить время чтения, написав функцию, которая будет удалять все, что расположено внутри скобок.

Помогите ленивому Диме разработать функцию shortener(st), которая будет удалять все, что внутри скобок и сами эти скобки, возвращая очищенный текст (скобки могут быть вложенными).

1. Валентина прогуляла лекцию по математике.

Преподаватель решил подшутить над нерадивой студенткой и

попросил ее на практическом занятии перечислить все положительные делители некоторых целых чисел.

Для несложных примеров студентка быстро нашла решения (для числа 6 это: 1, 2, 3, 6; а для числа 16 это: 1, 2, 4, 8, 16), но этим все не закончилось.

На домашнее задание ей дали варианты посложнее: 23436, 190187200, 380457890232.

Решить такое вручную, как вы понимаете, практически нереально.

Вот Валентина и обратилась к вам за помощью.

Помогите ей (при помощи функции all\_divisors(number)).

Постарайтесь найти самое оптимальное решение.

Результат представьте в виде списка (не забудьте отсортировать по возрастанию).