

## #95: КАПИТАЛИЗАЦИЯ

Сложность: 1 из 10

Капитализация — это начисление «процентов на процент», когда доход, начисленный за один расчётный период, прибавляется к общей сумме вложенных денег, и в следующий расчётный период процент начисляется уже на большую сумму.

Напишите функцию **capitalization**, которая принимает три аргумента: вложенную сумму, годовой процент по вкладу и срок вклада в месяцах. Все три аргумента — целые числа. Функция должна возвращать сумму, которую получит пользователь по окончании вклада. Возвращаемая сумма должна быть целым числом.

### Пример использования функции

```
money = capitalization(10000, 10, 2)

print(money)

10167
```

## #82: СЧАСТЛИВЫЙ ИЛИ ВСТРЕЧНЫЙ

Сложность: 1 из 10

Напишите функцию **ticket\_type**, которая получает номер билета в виде строки из шести цифр и возвращает его тип: счастливый, встречный, пьяный или обычный.

**Счастливым** называют такой билет, что сумма первых трех цифр его номера равна сумме последних трех цифр.

**Встречным** называют такой билет, что сумма первых трех цифр его номера отличается на единицу от суммы последних трех цифр.

**Пьяным** называют такой билет, что сумма первых трех цифр его номера отличается на двойку от суммы последних трех цифр.

**Обычными** называют все остальные билеты.

### Пример использования функции:

```
result = ticket_type("123321")

print(result)

счастливый
```

## #87: ВЧЕРА И ЗАВТРА

Сложность: 2 из 10

Написать функцию **dates**, которая принимает аргумент-строку **date** (текущий день) и возвращает кортеж из двух элементов: предыдущий день и следующий день, относительно переданного.

Аргумент **date** передается в виде строки формата ДД.ММ.ГГГГ. Возвращаемый кортеж также содержит даты в виде строк в формате ДД.ММ.ГГГГ

Считаем, что год не високосный.

**Пример использования функции:**

```
result = dates('12.03.2016')
print(result)
('11.03.2016', '13.03.2016')
```

## #89: ЭТАЖ

Сложность: 2 из 10

Написать функцию **floor**, которая возвращает этаж, на котором живет пользователь.

Функция принимает два позиционных аргумента:

1. Номер квартиры.
2. Количество квартир на этаже.

Программа должна называться **clients.py** и содержать только функцию **floor**.

**Пример вызова функции**

```
result = floor(8, 3)
```

\* В переменную **result** будет возвращено значение 3.

## #72: ВИСОКОСНЫЙ ГОД

Сложность: 2 из 10

Написать функцию **is\_leap\_year**, которая принимает год и возвращает **True** если год високосный и **False** в противном случае.

Год является високосным, если его номер кратен 4. Но из кратных 100 високосными являются лишь кратные 400. То есть 2000 - високосный, а 1800 - нет.

## #85: МНОЖЕСТВЕННОЕ ЧИСЛО

Сложность: 2 из 10

В русском языке, если нужно рассказать о нескольких предметах, люди используют множественное число.

В зависимости от количества предметов, форма существительного после числа может меняться.

Например, один ананас, два ананаса, пять ананасов. Как видите, количество предметов меняется (один, два, пять) и у слова **ананас** изменяется окончание (ананас, ананаса, ананасов).

Напишите функцию **choose\_plural**, которая принимает два аргумента: целое число (количество предметов) и строку содержащую три слова разделенных запятыми, которые представляют собой формы множественного числа.

Функция должна возвращать правильную форму множественного числа в зависимости от переданного первым аргументом значения.

**Пример использования функции:**

```
# 7 анансов. Для числа 7 правильная форма будет "ананасов".
print(choose_plural(7, "ананас,ананаса,ананасов"))
ананасов

# 21 ананас. Для числа 21 правильная форма будет "ананас".
print(choose_plural(21, "ананас,ананаса,ананасов"))
ананас
```

## #49: ШИФР ЦЕЗАРЯ

Сложность: 3 из 10

Напишите функцию **code**, которая кодирует переданную в неё строку с помощью [шифра Цезаря](#).

Функция принимает два аргумента:

1. текст для кодирования;
2. необязательный аргумент **key** (ключ сдвига) со значением по умолчанию 1.

```
text_007 = code("Агент 007, срочно выйдите на связь")
print(text_007)
Бджоу!118-!тспшоп!гькейуж!об!тгèиэ
```

## #67: РАССТОЯНИЕ МЕЖДУ ТОЧКАМИ

Сложность: 3 из 10

Написать функцию **distance**, которая вычисляет расстояние между двумя точками на плоскости или в пространстве.

Функция принимает два аргумента. Каждый аргумент - это кортеж из двух или трех элементов: X и Y для точки на плоскости и X, Y, Z - для точки в пространстве.

### Пример использования

```
print(distance((-1, 3), (6, 2)))
7.0710678118654755
print(distance((-1, 3, 3), (6, 2, -2)))
8.660254037844387
```

## #79: МАГАЗИНЫ ПОБЛИЗОСТИ

Сложность: 4 из 10

Написать функцию **near\_shops**, которая находит все магазины в радиусе R от заданной точки.

Функция принимает три аргумента:

1. Координаты точки, вокруг которой будем искать. В виде кортежа из двух элементов (X, Y).
2. Радиус поиска - R.
3. Список магазинов из кортежей вида ("название магазина", (Координата\_X, Координата\_Y))

Функция должна возвращать список из кортежей вида ("название магазина", Расстояние\_до\_магазина). Кортежи в списке должны быть отсортированы по расстоянию до магазина (от близких к дальним). В список должны попасть только магазины, которые расположены в пределах R.

### Пример использования

```
shops = [
    ("Магазин 1", (1, 1)),
    ("Магазин 2", (-1, 0)),
    ("Магазин 3", (2, -1)),
    ("Магазин 4", (2, 4)),
    ("Магазин 5", (2, 0)),
]

my_shops = near_shops((2, 3), 3, shops)

[('Магазин 4', 1.0), ('Магазин 1', 2.23606797749979), ('Магазин 5', 3.0)]
```

Для сортировки конечного списка вы можете воспользоваться функцией [sorted](#).