

# Лабораторная 7. Вариант 2.

## Задача. Палиндромные потомки

### Описание

В данной лабораторной работе вам необходимо реализовать две функции. Одна обычным способом, другая рекурсивным.

Также нужно создать декоратор, который будет отслеживать время работы каждой функции и эмпирически показать, какой из способов является оптимальным.

В отдельном файле test.txt прописать минимум 10 всевозможных случаев(рассмотреть также частные случаи), включающих проверку как для больших, так и маленьких по длине или значению входных данных.

Формат записи, следующий:

```
Случай 1
#Обычная функция
Название функции:
Аргументы:
Время выполнения: __ сек.
Результат:
```

```
#Рекурсивная функция
Название функции:
Аргументы:
Время выполнения: __ сек.
Результат
```

```
...
Случай N
...
```

### Формулировка задачи

Может быть так, что число не является палиндромом, а вот его потомок — является.

Прямой потомок получается из цифр, равных сумме сложения всех пар соседних цифр предка.

Например, 123312 — это не палиндром, а вот его потомок 363 — является. Цифры числа 363 — суммы соседних цифр числа 123312:  $3 = 1 + 2$ ;  $6 = 3 + 3$ ;  $3 = 1 + 2$ .

Напишите рекурсивную функцию, которая возвращает True, если число само по себе или любой из его потомков до двух цифр включительно является палиндромом.

Рассмотреть случаи, если число содержит одну цифру или количество цифр в числе на входе нечетное выводить False

### Входные данные

Число

### Выходные данные

True или False

## Пример 1

### Входные данные

```
palindrome_descendant(13001120)  
# 13001120 → 4022 → 44
```

### Выходные данные

```
True
```

## Пример 2

### Входные данные

```
palindrome_descendant(11211230)  
# 11211230 → 2333 → 56 → 11
```

### Выходные данные

```
True
```

## Пример 3

### Входные данные

```
palindrome_descendant(23336014)  
# 23336014 → 5665
```

### Выходные данные

```
True
```

## Пример 4

### Входные данные

```
palindrome_descendant(11)
```

### Выходные данные

```
True
```

## Дополнительные тесты

Файл `main.py` проверяется с помощью линтера [super\\_linter](#). При проверке игнорируются ошибки D, S, I.