

ГУАП

КАФЕДРА № 41

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

Е.К. Григорьев

инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5

Рекурсия и замыкания

по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4116

подпись, дата

Четвергов В.Ю.

инициалы, фамилия

Санкт-Петербург 2023

Цель работы: познакомиться с основными способами объявления и использования рекурсивных функций, а также механизма замыканий в Python. Вариант 7

### Часть 1. Задания на рекурсию:

Напишите рекурсивную функцию вычисления числа Фибоначчи `fibonacci`, на вход которой подается целочисленное значение `n`

```
def fib(n):  
    if n in (1, 2):  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)  
  
n = int(input("n="))  
print("Фибоначчи для n =", n, "равно", fib(n))
```

Напишите рекурсивную функцию `pow_n`, на вход которой подается два значения `n` и `k` (по умолчанию равно 8). Функция должна возвращать значение следующего вида:  $n^k$

```
def pow_n(n, k=8):  
    if k == 0:  
        return 1  
    elif k == 1:  
        return n  
    else:  
        return n * pow_n(n, k - 1)  
  
n = int(input("Введите число n: "))  
k = int(input("Введите число k (по умолчанию 8): ") or "8")  
result = pow_n(n, k)  
print(f"{n} в степени {k} равно {result}")
```

Напишите рекурсивную функцию `remove_brackets`, которой на вход подается строка, содержащая одну пару из открывающейся и закрывающейся скобки. Функция должна вернуть строку, состоящую из символов, находящихся в скобках исходной строки

```
def remove_brackets(s):  
    if s[0] == '(' and s[-1] == ')':
```

```

        return s[1:-1]

    else:

        return "Неверный формат строки"

input_str = input("Введите строку с одной парой скобок: ")
result = remove_brackets(input_str)
print(f"Содержимое скобок: {result}")

```

Напишите рекурсивную функцию `num_reverse`, на вход которой подается целочисленное значение. Функция должна возвращать число, где цифры расположены в обратном порядке. Например, 1456 -> 6541

```

def num_reverse(n):

    if n < 10:

        return n

    else:

        return (n % 10) * (10 ** (len(str(n)) - 1)) + num_reverse(n // 10)

num = int(input("Введите целое число: "))
result = num_reverse(num)
print(f"Результат: {result}")

```

## **Часть 2. Задания на замыкания:**

Напишите функцию `closure_count_str`, на вход которой подается строка. Функция должна возвращать другую функцию, принимающую символ и возвращающую количество его повторений

```

def closure_count_str(s):

    def count_char(c):

        return s.count(c)

    return count_char

user_str = input("Введите строку: ")
count_func = closure_count_str(user_str)

```

```

user_char = input("Введите символ для подсчета: ")

count = count_func(user_char)

print(f"Символ {user_char} встречается {count} раз(а) в строке
\"{user_str}\"")

```

Напишите функцию `closure_list_pow`, на вход которой подается список целочисленных или вещественных значений. Используя механизм замыкания верните функцию, принимающую значение степени, в которую необходимо возвести каждый элемент списка и возвращающую полученный результат

```

def closure_list_pow(lst):
    def list_pow(power):
        return [i ** power for i in lst]
    return list_pow

# Запрашиваем у пользователя список чисел
numbers = input("Введите список чисел через запятую: ")
numbers_list = [float(num.strip()) for num in numbers.split(",")]

# Создаем функцию, которая будет возводить числа в степень
pow_func = closure_list_pow(numbers_list)

# Запрашиваем у пользователя значение степени
power = float(input("Введите значение степени: "))

# Возводим числа в заданную степень
result = pow_func(power)

# Выводим результат
print(result)

```

Напишите функцию `closure_str`, на вход которой подается строка. Функция должна возвращать другую функцию, принимающую номер индекса и возвращающую символ, располагаемый в строке по этому индексу. Если задаваемый индекс выходит за пределы строки, то верните пустой символ.

```

def closure_str(s):
    def get_char_at_idx(idx):
        if idx < len(s):
            return s[idx]
        else:

```

```
        return ""

    return get_char_at_idx

my_str = input("Введите строку: ")

get_char = closure_str(my_str)

idx = int(input("Введите индекс: "))

print(f"Символ под индексом {idx}: {get_char(idx)}")
```

Вывод: я ознакомлен с основными способами работы с рекурсией и способом замыкания Python