МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №4

Дисциплина: «Языки программирования»

Выполнил: студент 2 курса
группы ИТС-б-о-20-1
Попов Данила Владимирович
Проверил:
к.ф.-м.н., доцент
кафедры инфокоммуникаций
Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2021

Цель работы: приобретение навыков по работе с рекурсивными функциями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

Ссылка на репозиторий: https://github.com/12W300/Four.git

Решено задание: Самостоятельно изучите работу со стандартным пакетом Python timeit. Оцените с помощью этого модуля скорость работы итеративной и рекурсивной версий функций factorial и fib . Во сколько раз измениться скорость работы рекурсивных версий функций factorial и fib при использовании декоратора lru_cache.

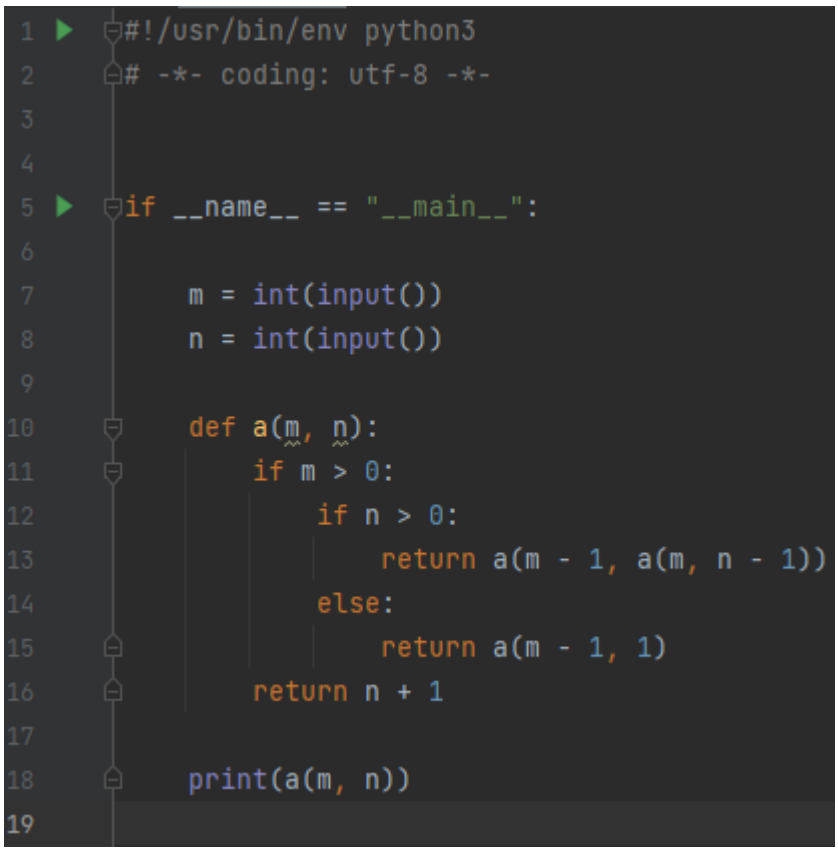
```
Факториал без оптимизации выполняется за 0.0008917000000000022 секунд
Факториал с оптимизацией выполняется за 0.0002882999999999983 секунд
75025
Фиббоначи без оптимизации выполняется за 0.024702799999999997 секунд
75025
Фиббоначи с оптимизацией выполняется за 0.024492499999999993 секунд

Process finished with exit code 0
```

Рисунок 1 – пример

Решено индивидуальное задание для варианта 13:

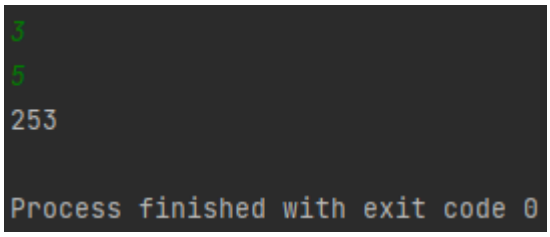
13. Напишите программу вычисления функции Аккермана для всех неотрицательных целых аргументов m и n:

$$A(m,n) = \begin{cases} A(0,n) = n+1 \\ A(m,0) = A(m-1,1), & m \\ A(m,n) = A(m-1,A(m,n-1)), & m,n>0. \end{cases}$$



```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-


if __name__ == "__main__":

    m = int(input())
    n = int(input())

    def a(m, n):
        if m > 0:
            if n > 0:
                return a(m - 1, a(m, n - 1))
            else:
                return a(m - 1, 1)
        return n + 1

    print(a(m, n))

```

Рисунок 2 – код индивидуального задания



```
3
5
253

Process finished with exit code 0
```

Рисунок 3 – работа кода

Вопросы для защиты работы:

1. Для чего нужна рекурсия?

Рекурсия занимает меньше времени и может работать в обратную сторону.

2. Что называется базой рекурсией?

Выражение, которое её останавливает.

3. Как получить текущее значение максимальной глубины рекурсии в языке Python?

Командой «sys.getrecursionlimit()».

4. Что произойдет если число рекурсивных вызовов превысит максимальную глубину рекурсии в языке Python?

Будет ошибка «RecursionError».

5. Как изменить максимальную глубину рекурсии в языке Python?

Командой «sys.setrecursionlimit(limit)».

6. Каково назначение декоратора lru_cache?

Его используют для уменьшения количества лишних вычислений.

7. Что такое хвостовая рекурсия?

Хвостовая рекурсия — частный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции.

Вывод: в ходе лабораторной работы были приобретены навыки по работе с множествами при написании программ с помощью языка программирования Python версии 3.x.