

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.9

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-20-1

Примаков В. Д. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

ВЫПОЛНЕНИЕ

Пример

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

def recursion(n):
    if n == 1:
        return 1

    return n + recursion(n - 1)

if __name__ == '__main__':
    print(f"Recursion: {recursion(5)}")

    n = 0
    k = 5
    for i in range(1, k + 1):
        n += i

    print(f"'for': {n}")
```

```
Recursion: 15
'for': 15

Process finished with exit code 0
```

Индивидуальное задание

4. Создайте рекурсивную функцию, печатающую все возможные перестановки для целых чисел от 1 до N .

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

def all_perms(arr):
    if len(arr) == 1:
        return [arr]
    else:
        a = arr[0]
        p = all_perms(arr[1:])
        r = []
        for pp in p:
            for i in range(len(pp)):
                tmp = pp[0:i] + [a] + pp[i:]
                r.append(tmp)
            r.append(pp + [a])
        return r

if __name__ == '__main__':
    print(all_perms([1, 2]))
```

```
[[1, 2], [2, 1]]
```

```
Process finished with exit code 0
```

```
[[1, 2, 3], [2, 1, 3], [2, 3, 1], [1, 3, 2], [3, 1, 2], [3, 2, 1]]
```

```
Process finished with exit code 0
```

```
[[1, 2, 3, 4], [2, 1, 3, 4], [2, 3, 1, 4], [2, 3, 4, 1], [1, 3, 2, 4], [3, 1, 2, 4], [3, 2, 1, 4], [3, 2, 4, 1], [1, 3, 4, 2], [3, 1, 4, 2], [3, 4, 1, 2], [3, 4, 2, 1], [2, 1, 4, 3], [2, 4, 1, 3], [2, 4, 3, 1], [1, 4, 2, 3], [4, 1, 2, 3], [4, 1, 3, 2], [4, 2, 1, 3], [4, 2, 3, 1], [4, 3, 1, 2], [4, 3, 2, 1]]
Process finished with exit code 0
```

Ссылки на репозитории

GitHub - https://github.com/surai5a/laba_2_9

Ответы на контрольные вопросы

1. Рекурсия существенно сокращает объем кода и входит во многие встроенные функции языков.
2. База рекурсии – это тривиальный случай, при котором решение задачи очевидно, то есть не требуется обращение функции к себе.
3. Компьютер использует стек вызовов — специальную область памяти, где хранит данные о точках перехода между фрагментами кода. последовательность шагов, выполняемых при вызове функции:
 - a. Программа сталкивается с вызовом функции.
 - b. Создается фрейм стека, который помещается в стек.
 - c. Процессор переходит к точке начала выполнения функции.
 - d. Инструкции внутри функции начинают выполняться.

После завершения функции, выполняются следующие шаги:

- e. Регистры восстанавливаются из стека вызовов.
 - f. Фрейм стека вытягивается из стека. Освобождается память, которая была выделена для всех локальных переменных и аргументов.
 - g. Обработывается возвращаемое значение.
 - h. ЦП возобновляет выполнение кода (исходя из обратного адреса).
4. Чтобы получить текущее значение максимальной глубины рекурсии следует вызвать функцию `sys.getrecursionlimit()`
 5. Когда предел достигнут, возникает исключение: `RuntimeError: Maximum Recursion Depth Exceeded`
 6. Можно изменить предел глубины рекурсии с помощью вызова: `sys.setrecursionlimit(limit)`
 7. `lru_cache` можно использовать для уменьшения количества лишних вычислений.
 8. Хвостовая рекурсия — частный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции. Оптимизация происходит, вызывая исключение, если оно является его прародителем, и перехватывает исключения, чтобы подделать оптимизацию хвоста.