Федеральное государственное автономное образовательное учреждение высшего образования

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Прикладная математика и фундаментальная информатика»

Практическое занятие №7

по дисциплине «Практикум по программированию»

на тему: «Функции в Python»

Вариант №14

Выполнил

Студент гр. **ФИТ-212** **Курпенов К.И.**

группа Фамилия И,О. студента подпись

Принял:

Преподаватель **Моисеева Н.А.**

Фамилия И.О, преподавателя дата, подпись

Омск 2022

**ЗАДАНИЕ 1.**

**Задача 1.1.**

Условие:

Напишите функцию Python, чтобы найти максимальное из трёх чисел.

Решение:

```python
def get_max(a: int, b: int, c: int) -> int:
    if a < b:
        if b < c:
            return c
        else:
            return b
    elif b < c:
        if c < a:
            return a
        else:
            return c
    elif a < c:
        if c < b:
            return b
        else:
            return c
    else:
        return -1
```

**Задача 1.2.**

Условие:

Напишите функцию для суммирования всех чисел в списке.

Решение:

```python
def get_sum(array: list) -> int:
    s = 0
    for i in array:
        s += i
    return s
```

**Задача 1.3.**

Условие:

Напишите функцию Python, которая принимает список и возвращает новый список с уникальными элементами первого списка.

Решение:

```python
def get_unique(array: list) -> list:
    unique = []
    for i in array:
        if i in unique:
            unique.append(i)
    return unique
```

**Задача 1.4.**

Условие:

Напишите функцию, которая выводит первые n строк треугольника Паскаля.

Решение:

```python
def get_triangle(rows: int) -> list:
    row = [1]
    output = []
    for _ in range(rows):
        output.append(row)
        row = [sum(x) for x in zip([0] + row, row + [0])]
    return output
```

**Задача 1.5.**

Условие:

Напишите функцию для создания и печати списка, в котором значения являются квадратами чисел от 1 до 30 включительно.

Решение:

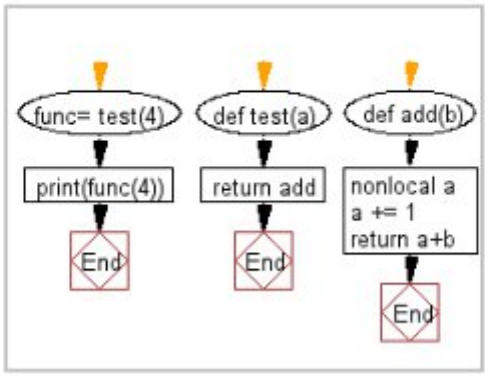
```python
def get_squares(count: int) -> list:
    return [i**2 for i in range(1, count + 1)]
```

**Задача 1.6.**

Условие:

Напишите программу для доступа к функции внутри функции.

**Блок-схема:**




Решение:

```
1 def test(a):
2     def add(b):
3         nonlocal a
4         a += 1
5         return a + b
6     return add
7
8
9 func = test(4)
10 print(func(4))
11
```

**Задача 1.7.**

Условие:

Напишите функцию для определения количества локальных переменных, объявленных в функции.

Решение:

```
1 def test() -> tuple:
2     a = 1
3     s = "a"
4     return a, s
5
6
7 print(test.__code__.co_nlocals)
8
```

**Задача 1.8.**

Условие:

Напишите программу, которая вызывает заданную функцию через определённые миллисекунды.

Решение:

```python
from time import sleep
from math import sqrt

def get_sqrt(number: int, pause: int) -> float:
    sleep(pause / 1000)
    return sqrt(number)
```

**Задача 1.9.**

Условие:

Напишите программу, которая принимает арифметическое выражение в качестве аргумента и выводит результат этого выражения.

Решение:

```python
def get_result(a: float, b: float, operator: str) -> float:
    return eval(f"{a} {operator} {b}")
```

**ЗАДАНИЕ 2.**

**Задача 1.**

Условие:

Составить программу для нахождения чисел из интервала `[M, N]`, имеющих наибольшее количество делителей.

Решение:

```python
def get_divisors_count(number: int):
    c = 2
    for i in range(2, c):
        if not (number % i):
            c += 1
    return c


def get_max_divisions_number(n: int, m: int) -> int:
    maximum = n
    for i in range(n, m + 1):
        temp = get_divisors_count(i)
        if maximum < temp:
            temp = maximum
    return maximum
```

**Задача 2.**

Условие:

Четыре точки заданы своими координатами `X(x1, x2), Y(y1, y2), Z(z1, z2), P(p1, p2)`. Выяснить, какие из них находятся на максимальном расстоянии друг от друга и вывести на экран значение этого расстояния. Вычисление расстояния между двумя точками оформить в виде процедуры.

Решение:

```python
from math import sqrt


def get_distance(x1: int, y1: int, x2: int, y2: int) -> float:
    return sqrt((x2 - x1)**2 + (y2 - y1)**2)


x1, x2 = map(int, input().split())
y1, y2 = map(int, input().split())
z1, z2 = map(int, input().split())
p1, p2 = map(int, input().split())

d1 = get_distance(x1, x2, y1, y2)
d2 = get_distance(x1, x2, z1, z2)
d3 = get_distance(x1, x2, p1, p2)
d4 = get_distance(y1, y2, z1, z2)
d5 = get_distance(y1, y2, p1, p2)
d6 = get_distance(z1, z2, p1, p2)

print(max([d1, d2, d3, d4, d5, d6]))
```