

# Лабораторная работа №3

## Математические вычисления

- 5. Дана переменная `val1 = 254`. Осуществите целочисленное деление на 4 (без остатка) и выведите полученный результат в терминал.
- 6. Дана переменная `val1 = 157`. Осуществите ее деление по модулю на 3 и выведите полученный результат в терминал.
- 8. На вход подается список `my_list`, минимум из 6 элементов. Найдите сумму его первого и последнего элемента и выведите полученный результат в терминал.
- 12.  $\frac{n-20}{\sqrt{n^3}}$
- 19.  $\frac{2n^2-4n+10}{2n}$

```
from typing import List, Union

def task_5(val: int = 254):
    print('task-5')
    print(val)
    print('val // 4 = ', val // 4)

def task_6(val: int = 157):
    print('\ntask-6')
    print(val)
    print('val % 3 = ', val % 3)

def task_8(ms: List[Union[int, float]] = 157):
    print('\ntask-8')
    print(ms)
    print("ms[0] + ms[-1] =", ms[0] + ms[-1])

def task_12(n: int = 157):
    print('\ntask-12')
    print(n)
    print("value = ", (n-20)/(n**(3/2)))

def task_19(n: int = 157):
    print('\ntask-19')
    print(n)
    print("value = ", (2*n**2-4*n+10)/(2*n))

def main():
    task_5()
    task_6()
```

```
task_8(ms=[1, 2, 3])
task_12()
task_19()

if __name__ == "__main__":
    main()
```

## Вывод

```
task-5
254
val // 4 = 63

task-6
157
val % 3 = 1

task-8
[1, 2, 3]
ms[0] + ms[-1] = 4

task-12
157
value = 0.06964197654271607

task-19
157
value = 155.03184713375796
```

## Управляемые конструкции:

- 4. На вход программы подается 2 значения: val1, val2. Если val1 делится на val2 без остатка, то выведите в терминал «Bingo!» (без кавычек), иначе выведите получившийся остаток от деления.
- 8. Пользователь вводит с клавиатуры целочисленное значение. Проверьте лежит ли оно в диапазоне [-15, 10] и выведите в терминал «YES» или «NO» (без кавычек) в зависимости от результата проверки
- 10. На вход подается 3 значения (a, b, c). Решите следующее квадратное уравнение  $ax^2 - bx + 2c = 0$  и выведите в терминал полученные корни. Например, x=5 (если корень один), x1=2 x2=6 (если имеется 2 корня), либо фразу «Корней нет!» (без кавычек).
- 15. Пользователь вводит с клавиатуры свой текущий возраст. Определите имеет ли он право участвовать в голосовании и выведите в терминал

«YES» или «NO» (без кавычек) в зависимости от результата проверки.

- 16. Пользователь вводит с клавиатуры номер месяца, а программа должна вывести в терминал количество дней в нем. Если заданное значение меньше 1, либо больше 12 выведите в терминал «Ошибка ввода!» (без кавычек).

```
def task_4(a: int, b: int):
    print("\ntask_4")
    print("a, b =", a, b)
    if a % b == 0:
        print("Bingo!")
    else:
        print("a % b =", a % b)

def task_8():
    print("\ntask_8")
    n = int(input("Input n: "))
    if -15 <= n <= 10:
        print(f"yes, {n} in [-15, 10]")
    else:
        print(f"no, {n} not in [-15, 10]")

def task_10(a: int, b: int, c: int):
    b = -b
    c = 2 * c
    print('\ntask_10')
    print(f'{a}x^2 + {b}x + c = 0')
    if (a == 0) and (b == 0) and (c == 0):
        print('no answer')
        return None
    elif (a == 0):
        if (b == 0):
            print('no answer')
        else:
            print((-c) / b)
        return None
    d = (b ** 2) - (4 * a * c)
    if d < 0:
        print('no answer')
    d = d ** 0.5
    x1 = (-b - d) / (2 * a)
    x2 = (-b + d) / (2 * a)
    if d == 0:
        print(x1)
    else:
        print(x1, x2)
```

```

def task_15(age: int):
    print('\ntask_15')
    print(age)
    if age >= 18:
        print('yes, age >= 18')
    else:
        print('no, age < 18')

def task_16(month: int):
    print('\ntask_16')
    print('month', month)
    days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    if 1 <= month <= 12:
        print(days[month])
    else:
        print('error in data')

def main():
    task_4(4, 2)
    task_4(3, 2)
    task_8()
    task_10(1, -4, 1.5) # -1 -3
    task_10(1, 0, 0) # 0
    task_10(0, 0, 1) # no answer

    task_15(17)
    task_15(19)

    task_16(0) # error
    task_16(13) # error
    task_16(1) # 31
    task_16(2) # 28

if __name__ == "__main__":
    main()

```

```

task_4
a, b = 4 2
Bingo!

task_4
a, b = 3 2
a % b = 1

task_8
Input n: 10

```

```
yes, 10 in [-15, 10]
```

```
task_10
```

```
1x^2 + 4x + c = 0
```

```
-3.0 -1.0
```

```
task_10
```

```
1x^2 + 0x + c = 0
```

```
0.0
```

```
task_10
```

```
0x^2 + 0x + c = 0
```

```
no answer
```

```
task_15
```

```
17
```

```
no, age < 18
```

```
task_15
```

```
19
```

```
yes, age >= 18
```

```
task_16
```

```
month 0
```

```
error in data
```

```
task_16
```

```
month 13
```

```
error in data
```

```
task_16
```

```
month 1
```

```
28
```

```
task_16
```

```
month 2
```

```
31
```

## Задачи на циклы:

- 3. Дан список `my_list1 = [1, 2, 3, 4, 9, 7, 4, 5.3, 9.7, 3]`. Используя цикл `while` найдите сумму его элементов и выведите полученный результат в терминал.

- 6. Дан список `my_list1 = [1, 2, 3, 4, 9, 7, 4, 5.3, 9.7, 3]`. Используя цикл `while` найдите сумму элементов с нечетным индексом и выведите полученный результат в терминал.
- 7. Дан список `my_list = [1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2]`. Используя цикл `while` посчитайте количество вхождения в него элементов со значением 2 и выведите в терминал полученный результат.
- 11. Используя цикл `while` посчитайте сумму значений от -34 до 15, которые нацело делятся на 4 и выведите в терминал полученный результат
- 14.  $\sum_{n=1}^z \frac{n-20}{\sqrt{(n^3)}}$

```
from typing import List, Union

def task_3(ms: List[Union[int, float]]) -> Union[int, float]:
    print('\ntask-3')
    print(ms)
    index = 0
    sm = 0
    while index < len(ms):
        sm += ms[index]
        index += 1
    print('sum', sm)

def task_6(ms: List[Union[int, float]]):
    print('\ntask-6')
    print(ms)
    index = 0
    sm = 0
    while index < len(ms):
        if index % 2 == 1:
            sm += ms[index]
        index += 1
    print('sum', sm)

def task_7(ms: List[Union[int, float]]):
    print('\ntask-7')
    print(ms)
    index = 0
    count = 0
    while index < len(ms):
        if ms[index] == 2:
            count += 1
        index += 1
    print('count', count)

def task_17(start, end):
    print('\ntask-17')
```

```

    index = start
    sm = 0
    while index % 4 != 0:
        index += 1
    while index <= end:
        index += 4
        sm += index
    print('sum', sm)
    return index

def task_14(z: int):
    print('\ntask-14')
    sm = 0
    for n in range(1, z):
        sm += (n - 20) / (n ** 1.5)
    print("sum", sm)
    return sm

def main():
    task_3([1, 2, 3, 4, 9, 7, 4, 5.3, 9.7, 3])

    task_6([1, 2, 3, 4, 9, 7, 4, 5.3, 9.7, 3])

    task_7([1, 2, 3, 4, 5, 6, 2, 3, 2, 1, 2])

    task_17(start=-34, end=15)

    task_14(z=100)

if __name__ == "__main__":
    main()

```

Output