

Лабораторная 2. Вариант 1.

Задача 1. Корень зла

Python изначально разрабатывался только для решения математических задач, поэтому давайте используем его, чтобы найти корни уравнения.

Вводится 3 вещественных числа a , b , c — коэффициенты уравнения вида:

$$ax^2 + bx + c = 0$$

Напишите функцию `find_roots(float, float, float)`, которая находит корни квадратного уравнения

Входные данные

Три вещественных числа (a , b , c)

Выходные данные

- Если у уравнения нет решений — следует вывести «No solution»
- Если один корень, то вывести его значение с точностью до сотых
- Если корня два, то вывести их сумму с точностью до сотых

Пример 1

Входные данные

```
find_roots(2, 5, 3)
```

Выходные данные

```
-2.5
```

Пример 2

Входные данные

```
find_roots(1, 2, 2)
```

Выходные данные

```
No solution
```

Пример 3

Входные данные

```
find_roots(1, 2, 1)
```

Выходные данные

-1.0

Задача 2. Фибоначчи

Числа Фибоначчи находят применение во многих областях от математики до биологии. В информатике Фибоначчи используются для [алгоритмов сортировки](#), [генерации случайных чисел](#) и кодирования [структур данных](#). Также задачу реализовать функцию [Фибоначчи](#) часто задают на собеседованиях.

Реализуйте функцию `fib(int)`, находящую положительные Числа Фибоначчи. Аргументом функции является порядковый номер числа.

- $f(0) = 0$
- $f(1) = 1$
- $f(n) = f(n-1) + f(n-2)$

Входные данные

Одно натуральное число — порядковый номер

Выходные данные

Одно натуральное число — элемент числовой последовательности.

Пример 1

Входные данные

```
fib(3)
```

Выходные данные

2

Пример 2

Входные данные

```
fib(5)
```

Выходные данные

5

Дополнительные тесты

Файл `main.py` проверяется с помощью линтера [super_linter](#). При проверке игнорируются ошибки D, S, I.