

ФИО: Дядев Владислав Александрович

Ссылка на вк: <https://vk.com/alvas07>

Прототип номера из ЕГЭ: №17 (обработка числовых последовательностей)

Задание сложного уровня

В файле 17-hard.txt дана последовательность целочисленных координат на оси Ox . Также дана функция: $f(x) = \frac{3x^3 - 4x^2 + 29x + 14}{(x^2 - 6x + 9)(x^2 - 5x + 4)}$. Найдите все пары координат такие, что для каждой из них функция $f(x)$ определена, а также модуль разности значений функции в этих точках имеет чётное число единиц в двоичной записи. У значений функции $f(x)$ берется только целая часть. В ответе запишите через пробел сначала количество найденных пар, затем наибольшее из расстояний между точками $(x_1, f(x_1))$ и $(x_2, f(x_2))$ таких пар (x_1, x_2) – первый и второй элементы в паре соответственно). От расстояния взять только целую часть. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Решение задания сложного уровня

Решение с пошаговыми комментариями представлено на скриншоте. Помимо этого, файлы с условием, пошаговым решением и входными данными представлены в репозитории на GitHub:

<https://github.com/Alvas07/Test/tree/main/Hard%20Task>

```
# Шаг 1: задаем функцию f(x), ставим целочисленное деление по условию
def f(x):
    return (3 * x ** 3 - 4 * x ** 2 + 29 * x + 14) // (
        (x ** 2 - 6 * x + 9) * (x ** 2 - 5 * x + 4)
    )

# Шаг 2: считываем из файла массив координат через генератор
coords = [int(x) for x in open("17-hard.txt")]

# Шаг 3: инициализируем массив для ответа
ans = []

# Шаг 4: запускаем цикл, который будет перебирать подряд идущие пары координат
for i in range(len(coords) - 1):
    # Шаг 5: переобозначаем для удобства координаты как x1 и x2
    x1, x2 = coords[i], coords[i + 1]
    # Шаг 6: проверяем, что функция f(x) определена при x1 и x2 (т. е. знаменатель ненулевой)
    if all(((x ** 2 - 6 * x + 9) * (x ** 2 - 5 * x + 4)) != 0 for x in (x1, x2)):
        # Шаг 7: считаем значения функций f(x1) и f(x2)
        y1, y2 = f(x1), f(x2)
        # Шаг 8: считаем модуль разности значений
        d = abs(y1 - y2)
        # Шаг 9: проверяем четность числа единиц в двоичной записи
        if bin(d)[2:].count("1") % 2 == 0:
            # Шаг 10: считаем расстояние между точками (x1, y1) и (x2, y2) по формуле
            r = int(((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5)
            # Шаг 11: добавляем в массив ans это расстояние
            ans.append(r)

# Шаг 12: выводим в ответ кол-во пар (длина массива) и максимальное расстояние (максимум массива)
print(len(ans), max(ans))
```