

# Rust Test Assignment

## Часть 1

Реализуйте функцию для разделения некоторой вычислительной работы между несколькими потоками.

Разделение должно происходить только на некотором пороге - если длина ввода меньше этого порога, обработка должна происходить в одном потоке.

Входные данные:

```
Vec<T>  
f(t: T) -> R
```

Порог перехода для многопоточной обработки может быть константой. На выходе функция должна возвращать `Vec<R>`, в котором порядок элементов соответствует входному списку

```
out[0] = f(in[0])
```

## Часть 2

**Для выполнения второй части необходимо использовать функцию из первой части.**

Есть список из  $n$  натуральных чисел ( $n > 0$ , значения в списке от 1 до `u64::MAX`). Так же имеется константа максимального количества итераций ( $k$ ).

Имеется преобразование: если число четное, то разделить на 2, если число нечетное, то умножить на 3 и прибавить 1.

Для каждого числа из списка необходимо получить количество итераций выполнения преобразования, которое приведёт к значению 1, или результат последнего преобразования, если количество итераций равно или превышает  $k$ .

Результат ожидается в виде списка, в котором индекс результата соответствует индексу в списке чисел.

Пример:

```
k = 8
```

Входные данные: [1, 2, 3, 100]

Выходные данные: [0, 1, 7, 88]

```
// 1: ответ = 0 итераций
```

```
// 2 -> 1: ответ = 1 итерация
```

```
// 3 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1: ответ = 7 итераций
```

```
// 100 -> 50 -> 25 -> 76 -> 38 -> 19 -> 58 -> 29 -> 88: ответ = 88
```

**Обе части должны иметь тесты.**