Домашнее задание №2

## Общие замечания.

* При решении задач нужно использовать только те возможности языка Go, которые были рассмотрены на тренинге на момент выдачи домашнего задания (некоторые исключения могут оговариваться особо в условии конкретных задач).
* Решение каждой задачи помещайте в отдельный каталог вида Task\_NN, например – Task\_01 (см. примеры кода к лекциям).
* Все решения запакуйте в один zip-архив и отправьте по почте тренеру.

## Задача 1.

Создайте функцию, которая работает следующим образом. Функция получает на вход целое число (тип int) и возвращает пару значений. Первое значение в паре – строка **"even"** или **"odd"**, в зависимости от количества простых делителей входного целого числа. То есть 15 = 3 \* 5 -> even, 30 = 2 \* 3 \* 5 -> odd, 9 = 3 \* 3 -> even, 7 = 7 -> odd. Второе значение в паре – булевский флаг. Он равен true, если входное целое число больше или равно 2, и false, если входное число меньше или равно единице (в этом случае первое значение в паре – пустая строка). В основной функции main протестировать созданную функцию на нескольких входных данных.

## Задача 2.

Создать функцию where, которая получает на вход срез целых чисел и функцию-предикат. Функция-предикат получает на вход целое число и возвращает true/false. Функция where возвращает срез, состоящий из чисел среза-аргумента, удовлетворяющих функции-предикату.

Создать функцию foreach для среза целых чисел и функции-действия. Функция-действие выполняет некий код для своего целого аргумента. Функция foreach запускает функцию-действие для каждого числа из своего среза-аргумента.

Протестировать работу функций where и foreach, используя в качестве предиката и действия анонимные функции.

## Задача 3.

Создать функцию sequence, получающую на вход произвольное количество целых чисел. Логика работы функции зависит от количества её аргументов:

* одно число a: вернуть срез с целыми числами от 0 до a включительно (или от a до 0, если a – отрицательное);
* два числа a и b: вернуть срез с целыми числами от a до b (или от b до a, если b меньше a);
* чисел больше двух – вернуть срез с этими числами;
* нет чисел – вернуть пустой срез.

## Задача 4.

Функция group получает в качестве аргумента карту, в которой ключ имеет тип byte, а значение для ключа – тип string. Результат работы функции – карта с байтовыми ключами из диапазона 0...9 и значениями для ключа в виде срезов из строк. Функция group выполняет группировку значений из исходной карты по признаку **последней цифры числа-ключа**. Например, если в исходной карте есть пары 11:"red" и 51:"green", то они должны быть сгруппированы в пару 1:["red", "green"]. Пар с пустыми срезами в карте-результате быть не должно.