**Цели практики:**

* Освоить на практике разные типы данных
* Попрактиковаться с нюансами работы этих типов данных (в особенности слайсов и maps)

**Типы данных**

* **Ловушки:**
  + Использование неправильного типа данных для переменной может привести к неожиданным результатам, таким как переполнение типа, потеря точности данных.
* **Задачи:**
  + Написать функцию, которая принимает на вход переменную типа int32 и добавляет к ней константу. Добиться переполнения int32 в результате работы этой функции.
  + Написать функцию, работающую с float32. Функция будет вычислять расстояние между двумя точками на поверхности Земли (координаты latitude/longitude). Формулу вычисления найдите самостоятельно и после реализации сравните полученные значения с тем же самым расстоянием, полученным по картам Яндекс/Google.
  + Написать функцию, работающую с float32. На входе две координаты (latitude/longitude). Определите, что они указывают на одну и ту же локацию. Подсказка: сравнение вещественных числе на строгое равенство - некорректный подход.
  + Напишите 2 варианта решения нахождения факториала: рекурсивный и итеративный. Попробуйте найти лимит рекурсии в golang. Для нахождения лимита рекурсии необязательно использовать вычисление факториала.

**Slices**

* **Ловушки:**
  + Слайсы изменяемы, поэтому будьте осторожны при передаче их функциям в качестве аргументов.
  + Слайсы могут увеличиваться и уменьшаться по мере необходимости, но если не соблюдать осторожность, это может привести к неожиданным результатам.
  + Слайсы могут быть nil, поэтому обязательно проверьте это перед их использованием.
* **Задачи:**
  + Напишите функцию для реверса слайса (вернуть слайс в обратном порядке)
  + Напишите функцию для удаления дублей из слайса.
  + Напишите функцию для сортировки слайса любым способом. Поэксперементируйте с разными вариантами реализации: изменение in place без возврата значения, возврат из функции нового значения, сравните время выполнения для очень больших размеров слайсов.
  + Напишите функцию, которая добавляет значения в переданный слайс и возвращает измененный слайс. Проверьте разные значения len/capacity при создании слайса через make. Попробуйте поменять по индексу значения в изначальном и новом слайсах. Определите, когда добавление значений в слайс создает новый слайс, а когда возвращает тот же самый слайс (тот же массив под капотом) с новым элементом.

**Maps**

* **Ловушки:**
  + Maps изменяемы, поэтому будьте осторожны при передаче их функциям в качестве аргументов.
  + Maps могут быть nil, поэтому обязательно проверьте это перед их использованием.
  + Отличие nil map от nil slice в том, что в nil slice можно добавлять значения.
  + Ключи в map должны быть уникальными, иначе значение последнего ключа будет единственным доступным значением (перезапишет ранее добавленное в map значение)
* **Задачи:**
  + Напишите функцию для поиска наиболее часто встречающегося слова в строке. Программа должна использовать map для хранения количества слов. Напечатайте слова в порядке частоты их встречи.
  + Напишите простую функцию итерации по ключам map. Запустите несколько раз и посмотрите на выдаваемый порядок ключей. Объяснение результатов можно найти в [документации](https://go.dev/ref/spec#For_range). Также напечатайте в консоль map целиком (просто передав в fmt.Println) и запустите несколько раз. Обратите внимание на порядок ключей. [Объяснение](https://tip.golang.org/doc/go1.12" \l "fmt) ([код](https://cs.opensource.google/go/go/+/refs/tags/go1.18:src/fmt/print.go;l=769))
  + Напишите функцию для преобразования слайса строк в map. Ключами в map должны быть строки в слайсе, а значениями в map должно быть количество раз, которое каждая строка появляется в слайсе (задача похожа на предыдущую про частоту встречи слов в строке, попробуйте реализовать ее таким образом, чтобы можно было переиспользовать логику для пред. задачи).
  + Напишите функцию для поиска пересечения двух map. Функция должна вернуть новую map, содержащую ключи, присутствующие в обеих исходных map.

**Runes**

* **Ловушки:**
  + Руны — это символы в кодировке UTF-8, поэтому будьте осторожны при сравнении их с другими символами.
  + Руны можно использовать для обозначения любого символа, включая смайлики и другие специальные символы.
* **Задачи:**
  + Напишите функцию, переворачивающую строку рун.
  + Напишите функцию для удаления всех пробелов из строки рун.
  + Напишите функцию для поиска наиболее часто встречающегося символа в строке рун.
  + Напишите функцию для поиска длины строки без учета пробелов. Учтите, что пробелом могут быть так же символы табуляции.

**Pointers**

* **Ловушки:**
  + Указатели могут быть опасны при неправильном использовании. Например, разыменование нулевого указателя приведет к панике.
  + Указатели также могут привести к утечкам памяти, если с ними не обращаться осторожно.
* **Задачи:**
  + Напишите функцию, которая меняет местами две переменных через указатель.
  + Напишите реализацию linked list на указателях. Напишите функцию, которая создает реверсивный linked list, не меняя исходный.
  + Напишите функцию, которая создает копию слайса структур. Важно: в структуре обязательно должно быть поле типа указатель. Проверьте, что после создания копии изменения в исходном слайсе не влияют на изменения в скопированном слайсе.

**Бонусные задачи:**

* Напишите программу, реализующую простую хэш таблицу, не используя map. Подсказка: использовать слайс предопределенного размера. Считать хэш значения, на основе которого понимать в какой элемент слайса будет попадать значение. Реализацию разрешения коллизий можно сделать на основе ранее реализованного linked list.