#### Практика

 Придумайте имена для переменных, объявите и инициализируйте их подходящим значением.

* Сайт по продаже недвижимости. Переменная содержит стоимость квартиры.
* Программа для учета товара в обувном магазине. Переменная содержит размер пары обуви.
* Информационная система для посетителей зоопарка. Переменная содержит количество животных, которые находятся в зоопарке в данный момент.
* Интернет магазин электроники. Переменная для цены нового товара.

Это задание для самостоятельной работы, его не нужно сдавать на проверку.

#### Практика

Дан текст с информацией о персонаже игры. Напишите программу, которая выводит этот текст. Вынесите из текста все числа в переменные, придумайте для них хорошие имена и выведите на экран.

===================

Супер гонки. Круг 4

===================

Шумахер (358)

===================

Водитель: Шумахер

Скорость: 358

-------------------

Оснащение

Двигатель: +254

Колеса: +93

Руль: +49

-------------------

Действия плохой погоды

Ветер: -21

Дождь: -17

### Задание 1: Улучшаем гонщика

Исправьте код программы так, чтобы значение скорости рассчитывалось автоматически. Скорость — это сумма показателей «Двигатель», «Колёса», «Руль» минус показатели «Дождь» и «Ветер».

package main  
  
  
import (  
    "fmt"  
)  
  
  
func main() {  
    var lap int = 4  
    var engine int = 254  
    var wheels int = 93  
    var steeringWheel int = 49  
    var wind int = 21  
    var rain int = 17  
  
  
    */\* расчет скорости \*/*  
    var speed int = 358  
  
  
    fmt.Println("===================")  
    fmt.Print("Супергонки. Круг ", lap, "\n")  
    fmt.Println("===================")  
    fmt.Print("Шумахер (", speed, ")\n")  
    fmt.Println("===================")  
    fmt.Println("Водитель: Шумахер")  
    fmt.Print("Скорость: ", speed, "\n")  
    fmt.Println("-------------------")  
    fmt.Println("Оснащение")  
    fmt.Print("Двигатель: +", engine, "\n")  
    fmt.Print("Колёса: +", wheels, "\n")  
    fmt.Print("Руль: +", steeringWheel, "\n")  
    fmt.Println("-------------------")  
    fmt.Println("Действия плохой погоды")  
    fmt.Print("Ветер: −", wind, "\n")  
    fmt.Print("Дождь: −", rain, "\n")  
}

### Задание 2: Калькулятор стоимости товара

Напишите программу для расчёта полной стоимости товара. Даны следующие значения: стоимость товара — 6400, стоимость доставки — 350, размер скидки — 700. Полная стоимость рассчитывается следующим образом:

стоимость товара + стоимость доставки − скидка

**Пример работы программы**

Калькулятор стоимости товара со скидкой.  
Стоимость товара: 6400  
Стоимость доставки: 350  
Размер скидки: 700  
---------  
Итого: 6050

### Задание 3: Время, время!

Напишите программу для ресторана быстрого питания «Крылышки и ножки». Определите, какое количество клиентов успеет обслужить один кассир за рабочую смену. Время на обслуживание складывается из времени, за которое клиент делает заказ, и времени, за которое кассир собирает заказ и выдаёт его клиенту.

Исходные данные: длительность смены — 400 минут; время, за которое клиент делает заказ, — 2 минуты; время, за которое кассир собирает заказ, — 4 минуты.

**Пример работы программы**

Расчёт количества клиентов, обслуживаемых за смену.  
Введите длительность смены в минутах: 480  
Сколько минут клиент делает заказ: 2  
Сколько минут кассир собирает заказ: 4  
-----Считаем-----  
За смену длиной 480 минут кассир успеет обслужить 80 клиентов.

### Задание 4: Непонятная квитанция

Управляющая компания рассылает странные квитанции: в них указана только полная стоимость ремонта многоквартирного дома. А сколько должны платить жильцы конкретной квартиры — непонятно.

Напишите программу, с помощью которой жильцы квартир могут посчитать, какую сумму им надо заплатить. Сумма делится делится на все квартиры вне зависимости от площади, количества жильцов и комнат.

Исходные данные: в доме 10 подъездов, в каждом подъезде 40 квартир, сумма, указанная в квитанции, — 4 000 000 рублей.

**Пример работы программы**

Расчёт стоимости ремонта для каждой квартиры.  
Сумма, указанная в квитанции: 4000000 рублей.  
Подъездов в доме: 10  
Квартир в каждом подъезде: 40  
----Результат-----  
Каждая квартира должна платить по 10000 рублей.

Практика

1. Оказывается, не все люди умеют вычислять квадрат чисел. А ведь это просто необходимо! Буквально каждый день, а то и не один раз, нам нужно вычислить квадрат какого-нибудь числа.  
   Напишите программу, которая вычисляет квадрат числа. Пользователь должен ввести число (a), а программа выводит квадрат числа (a \* a).
2. Теперь вернемся к программе с расчетом стоимости товара. Измените ее так, чтобы пользователь мог вводить свои данные.  
   Переменные для хранения цены, стоимости доставки и скидки у нас уже есть, вам нужно дописать код, который будет записывать в них введенные пользователем числа. Не забудьте добавить подсказки, чтобы было понятно, что нужно вводить.

#### Практика

**1.** Мы же готовимся к тому, чтобы стать настоящими веб-разработчиками, а что должно быть на каждом сайте? Конечно — регистрация! Самое время вам сделать первый шаг в этом направлении!

Напишите программу, которая принимает на вход имя пользователя и пароль как строки, а также его возраст целым числом, а потом выведите эти данные. Не забывайте делать подсказки пользователю, что ему нужно ввести.

Например, если пользователь введет имя username, пароль password, а возраст 32, то программа должна вывести такой текст:

Поздравляю, username, теперь вы зарегистрированы!  
Ваш пароль: password  
Ваш возраст: 32

Конечно, выводить пароль небезопасно, вдруг его кто-нибудь подсмотрит. Но мы же с вами только начинаем, и можем позволить себе такие допущения.

**2.** Регистрация у нас уже есть. Теперь нужно предоставить пользователям вход.

Для этого вы должны написать программу, которая будет принимать на вход логин пользователя и пароль, после чего выводить сообщение о том, что пользователь успешно вошел.

Пример работы программы:

Введите логин:  [введенный логин]

Введите пароль: [введенный пароль]  
-----  
[введенный логин], вы успешно зашли!

### Задание 1: Конструктор квестов

Напишите конструктор текстовых квестов. Пользователь вводит названия планеты, звёздной системы, имя рейнджера и сумму вознаграждения. Программа генерирует текст квеста на основе шаблона:

Как вам, <Ranger>, известно, мы - раса мирная, поэтому на наших военных кораблях используются наемники с других планет. Система набора отработана давным-давно. Обычно мы приглашаем на наши корабли людей с планеты <Planet> системы <Star>.  
  
Но случилась неприятность - в связи с большими потерями, в последнее время, престиж профессии сильно упал, и теперь не так-то просто завербовать призывников. Главный комиссар планеты <Planet>, впрочем, отлично справлялся, но недавно его наградили орденом Сутулого с закруткой на спине, и его на радостях парализовало! Призыв под угрозой срыва!  
  
<Ranger>, вы должны прибыть на планету <Planet> системы <Star> и помочь выполнить план призыва. Мы готовы выплатить вам премию в <Money> кредитов за эту маленькую услугу.

<Planet>, <Star>, <Ranger> и <Money> должны быть заменены на введённые пользователем значения.

#### Пример работы программы

Как тебя зовут?  
Денис  
Название планеты?  
Амапал  
Звездная система?  
Кубанама  
Сумма вознаграждения?  
10000  
  
  
Как вам, Денис, известно, мы - раса мирная, поэтому на наших военных кораблях используются наемники с других планет.  
Система набора отработана давным-давно. Обычно мы приглашаем на наши корабли людей с планеты Амапал системы Кубанама.  
Но случилась неприятность - в связи с большими потерями, в последнее время, престиж профессии сильно упал, и теперь не так-то просто завербовать призывников.  
Главный комиссар планеты Амапал, впрочем, отлично справлялся, но недавно его наградили орденом Сутулого с закруткой на спине, и его на радостях парализовало!  
Призыв под угрозой срыва!  
Денис, вы должны прибыть на планету Амапал системы Кубанама и помочь выполнить план призыва. Мы готовы выплатить вам премию в 10000 кредитов за эту маленькую услугу.

### Задание 2: Симулятор маршрутки

Напишите программу, которая симулирует работу маршрутки. Программа умеет объявлять остановки и узнавать у пользователя, сколько человек вышло на этой остановке и сколько зашло. Реализуйте маршрут из четырёх остановок, а в конце рассчитайте, сколько денег заработано, при условии, что билет стоит 20 рублей. Расходы следующие:

* четверть — на зарплату водителя;
* пятая часть — на топливо;
* пятая часть — на налоги;
* пятая часть — на ремонт машины.

#### Пример работы программы

Прибываем на остановку ул. Будапештская  
В салоне пассажиров: 0  
Сколько пассажиров зашло на остановке: 7  
Отправляемся с остановки ул. Будапештская  
  
Прибываем на остановку ул. Славы 4  
В салоне пассажиров: 7  
Сколько пассажиров вышло на остановке: 5  
Сколько пассажиров зашло на остановке: 4  
Отправляемся с остановки ул. Славы 4  
  
Прибываем на остановку Вечный огонь  
В салоне пассажиров: 6  
Сколько пассажиров вышло на остановке: 4  
Сколько пассажиров зашло на остановке: 5  
Отправляемся с остановки Вечный огонь  
  
Прибываем на остановку Аэропорт  
В салоне пассажиров: 7  
Конечная, все пассажиры вышли  
  
Всего заработано 320 руб.  
Зарплата водителя: 80 руб.  
Расходы на топливо: 64 руб.  
Налоги 64 руб.  
Расходы на ремонт машины: 64 руб.  
Итого доход: 48 руб.

### Задание 3: Обмен местами

Допишите программу, которая меняет местами значения переменных. Добейтесь того, чтобы в переменной a лежало значение из b, в b лежало значение из a.

package main  
  
import (  
    "fmt"  
)  
   
func main() {  
    a := 42  
    b := 153  
    fmt.Println("До обмена значений местами")  
    fmt.Println("a:", a)  
    fmt.Println("b:", b)  
  
    // обмен местами значений переменных  
      
    fmt.Println("После обмена значений местами")  
    fmt.Println("a:", a)  
    fmt.Println("b:", b)  
}

Пример работы программы

До обмена значений местами  
a: 42  
b: 153  
После обмена значений местами  
a: 153  
b: 42

### Задание 4: Злостные вредители

Доработаем программу, разобранную в практической работе темы 3.4 «Составные операторы». Теперь пользователь может ввести любую скорость роста бамбука, скорость его поедания гусеницами, высоту саженца и количество дней.

#### Пример работы программы

Высота саженца?  
100  
Скорость роста?  
50  
Скорость поедания?  
20  
Количество дней?  
3  
К середине 3 дня бамбук вырастет до 185 сантиметров

#### Практика

##### Задача 1. Космонавты

Для пилотируемой миссии на Марс отобрали троих космонавтов. Беда в том, что они оказались настолько похожи, что выбрать командира экипажа оказалось затруднительно. Тогда решили проверить их IQ. Напишите программу, которая запрашивает IQ трёх космонавтов и назначает командира по наивысшему IQ.

##### Задача 2. Скидочный калькулятор

Напишите программу — скидочный калькулятор для ресторана. Скидка действует только в будний день. Пользователь вводит номер дня недели (от 1 до 7) и сумму чека, программа рассчитывает сумму скидки как 10% от суммы чека, начиная с 10 000 руб, если этот день не будний.

##### Задача 3. Високосный год

Напишите программу, которая определяет количество дней в году. Алгоритм такой: год должен делиться на 4, однако если он делится на 100, но не делится на 400, то год не високосный.

##### Задача 4. Угадай число\* (сложная задача)

Напишите программу, которая угадывает число, задуманное пользователем от 1 до 5. Программа должна спрашивать, больше или меньше задуманное число (от какого-то другого числа) и на основе ответов пользователя определить то, что он загадал.

#### Практика

##### Задача 1. Минимум

Напишите программу, которая ищет минимум из двух чисел.

Подсказка: учтите, что числа могут быть равны!

##### Задача 2. Складываем в уме

Напишите программу, которая проверяет то, как вы умеете складывать два числа в уме. Программа должна выводить два разных сообщения на верный и неверный ответ пользователя. В последнем случае, надо показывать правильный результат. Пользователь должен ввести с клавиатуры два целых числа, а потом результат их сложения.

##### Задача 3. Проверка на чётное число

Напишите программу, которая проверяет, четное ли число ввел пользователь.

Подсказка: для проверки чётности используйте оператор %. Он вычисляет остаток от деления двух чисел. Пример:

remainder := x % 2 // вычисляет остаток от деления `x` на 2

##### Задача 4. Калькулятор опыта

Напишите программу, которая определяет уровень персонажа в компьютерной игре. Пользователь вводит число "очков опыта", а программа вычисляет уровень. Новый уровень дается при достижении 1000, 2500 и 5000 "очков опыта". Начальный уровень равен 1, а максимальный 4 (при достижении 5000 очков и больше).

##### Задача 5. Кратность числа

Напишите программу, которая проверяет, делится ли одно число на другое без остатка.

##### Задача 6. Улучшим барберов

Кстати, именно оператор % мог помочь нам с барберами. Перепишите программу про барбершопы с использованием % , для проверки на проблему с целочисленным делением.

##### Задача 7. Меню ресторана

Напишите программу, которая выводит меню бизнес-ланча ресторана, в зависимости от дня недели.

В меню есть общая часть, а есть уникальная часть, которая зависит от дня недели. Пользователь должен ввести номер дня недели, от 1 (понедельник) до 7 (воскресенье), а программа должна вывести на экран день недели и меню этого дня.

##### Задача 8. Зарплата\* (сложная задача)

В отделе маркетинга работают 3 сотрудника. У каждого — разные зарплаты. Напишите программу, которая вычисляет разницу между самой высокой и низкой зарплатой сотрудника, а также среднюю зарплату отдела.

Практика

Задача 1. Калькулятор скидки

Вы покупаете 3 товара в магазине. Если сумма вашего чека превышает 10 000 руб., вам будет сделана скидка 10%.

Напишите программу, которая запрашивает 3 стоимости товара и вычисляет сумму чека.

Задача 2. Автомобиль

Автомобиль выехал из Москвы в Рязань. Напишите программу, которая вычисляет, сумеет ли добраться автомобиль до Рязани за два часа, с заданной пользователем средней скоростью движения. Программа должна выводить вычисленное расстояние, а если автомобиль сумеет добраться до Рязани, — сообщение "Вы приехали".

Расстояние между городами — 200 км.

Задача 3. Модуль числа

Напишите программу, которая вычисляет модуль числа.

Подсказка: чтобы обратить знак числа в переменной ‘x’ надо писать вот так:

x = -x

Задача 4. Барберы\* (сложная задача)

В небольшом городке проживает 18 000 взрослых человек: 9 000 мужчин и 9 000 женщин. Согласно традиции этого городка, все мужчины обязаны носить бороду. Один предприимчивый житель открыл сеть барбершопов, чтобы помочь мужчинам ухаживать за их бородами. Однако, он никак не может посчитать, сколько всего специалистов-барберов для этого нужно.

Напишите программу, которая будет рассчитывать: сколько мужчин можно постричь во всех барбершопах, исходя из следующих данных:

* Каждый из 9000 мужчин посещает барбершоп раз в месяц (30 дней)
* Один барбер способен обслужить одного клиента за 1 час
* Смена барбера — 8 часов

Программа должна спросить, сколько мужчин проживает в городе и сколько всего барберов уже работает во всех барбершопах, и посчитать сколько барберов нужно, и если их недостаточно — выдать сообщение об этом. Если барберов достаточно — сказать и об этом.

Подсказка: в конструкции условия можно сравнивать не только переменную с числом, но и две переменные! А еще — целые математические выражения. Например:

if a \* b > c {

    // ...

}

**1.Условный оператор if**

Задача

Цель практической работы

Закрепить на практике полученные знания об операторе if, вложенных конструкциях с if и области видимости переменных.

Что входит в работу

1. Баллы ЕГЭ.
2. Три числа.
3. Банкомат.
4. Три числа: ещё попытка.
5. Ресторан.
6. По желанию. Студенты.

Задания по желанию — необязательные, повышенной сложности, требуют самостоятельного изучения дополнительного материала.

Задание 1. Баллы ЕГЭ

Что нужно сделать

Напишите программу, которая запрашивает у пользователя результаты ЕГЭ по трём экзаменам и проверяет, поступил ли он в институт или нет. Сумма проходных баллов в институт — 275.

Рекомендация

Пример работы программы:

Баллы ЕГЭ.  
Введите результат первого экзамена:  
91  
Введите результат второго экзамена:  
36  
Введите результат третьего экзамена:  
56  
Сумма проходных баллов: 275  
Количество набранных баллов: 183  
Вы не поступили.

Задание 2. Три числа

Что нужно сделать

Напишите программу, которая запрашивает у пользователя три числа и сообщает, есть ли среди них число больше пяти.

Рекомендация

Пример работы программы:

Три числа.  
Введите первое число:  
3  
Введите второе число:  
6  
Введите третье число:  
2  
Среди введённых чисел есть число больше 5.

Задание 3. Банкомат

Что нужно сделать

Банкомат выдаёт только купюры номиналом 100 рублей, а максимальная сумма снятия — 100 000 рублей. Напишите программу, которая проверяет допустимость введённой пользователем суммы средств. Если сумма для снятия доступна, сообщите «Операция успешно выполнена», в ином случае укажите причину, по которой невозможно выполнить операцию.

Рекомендация

Пример работы программы:

Банкомат.  
Введите сумму снятия со счёта:  
500  
Операция успешно выполнена.  
Вы сняли 500 рублей.

Задание 4. Три числа: ещё попытка

Что нужно сделать

Напишите программу, которая запрашивает у пользователя три числа и выводит количество чисел, которые больше или равны пяти.

Рекомендация

Пример работы программы:

Три числа.  
Введите первое число:  
3  
Введите второе число:  
5  
Введите третье число:  
7  
Среди введённых чисел 2 больше или равны 5.

Задание 5. Ресторан

Что нужно сделать

Напишите программу, которая запрашивает день недели, число гостей и сумму чека и рассчитывает сумму к оплате. В ресторане действуют следующие правила:

* по понедельникам должна применяться скидка 10% на всё меню, потому что понедельник — день тяжёлый;
* по пятницам, если сумма чека превышает 10 000 рублей, включается дополнительная скидка в размере 5%;
* если число гостей в одной компании превышает пять человек, автоматически включается надбавка на обслуживание 10%.

Рекомендация

Пример работы программы:

Введите день недели:  
5  
Введите число гостей:  
7  
Введите сумму чека:  
12000  
Скидка по пятницам: 600  
Надбавка на обслуживание: 1200  
Сумма к оплате: 12600

Задание 6 (по желанию). Студенты

Что нужно сделать

Перед старостой группы стоит задача разделить весь курс, состоящий из N студентов, на K групп. Напишите программу, которая поможет старосте сделать это: он вводит N, K и порядковый номер студента, а программа определяет, в какую группу он попадёт.

Рекомендация

В одну группу могут попадать студенты из разных частей списка.

**2.Логические операторы и составные условия**

Задание 1. Определение координатной плоскости точки

Что нужно сделать

Напишите программу, которая поможет пользователю определить, к какой координатной четверти принадлежит точка. Пользователь вводит числа X и Y, а программе необходимо вывести, какой координатной четверти принадлежит точка.

Рекомендация

Используйте логическое умножение.

Задание 2. Проверить, что одно из чисел — положительное

Что нужно сделать

Проверка пользовательского ввода на различные ограничения является частой задачей. Попросите пользователя ввести три числа и проверьте, что хотя бы одно является положительным. Результат проверки необходимо сообщить пользователю.

Рекомендация

Используйте логическое сложение.

Задание 3. Проверить, что есть совпадающие числа

Что нужно сделать

Пользователь при вводе может ошибиться, и многие программы пытаются привлечь внимание к таким ошибкам. Например, текстовые редакторы подсвечивают ошибки в словах, электронные таблицы — в формулах. Реализуйте программу, которая запрашивает у пользователя три числа и выводит подсказку, если два числа или более совпадают.

Рекомендация

Используйте логическое сложение.

Задание 4. Сумма без сдачи

Что нужно сделать

Программное обеспечение банкоматов постоянно решает задачу, как имеющимися купюрами сформировать сумму, введённую пользователем. Попробуйте решить похожую задачу и определить, сможет ли пользователь заплатить за товар без сдачи или нет. Для этого он будет вводить стоимость товара и номиналы трёх монет.

Рекомендация

Проверьте все сценарии, когда сумма может быть сформирована одной монетой, двумя или всеми тремя.

Задание 5. Решение квадратного уравнения

Что нужно сделать

Что компьютеры делают быстрее людей? Выполняют вычисления!

Напишите программу, решающую квадратные уравнения. Пользователь вводит коэффициенты a, b и c квадратного уравнения. Программа должна вывести корни уравнения. Для решения уравнения необходимо:

Вычислить дискриминант по формуле: D = (b × b − 4 × a × c)

* Если D > 0, будет два различных корня, которые находятся по формуле.
* Если D = 0, будет один корень, который находится по формуле.
* Если D < 0, то вывести, что корней нет.

Советы и рекомендации

Используйте переменные типа float64.

Для возведения b в квадрат воспользуйтесь [функцией math.Pow](https://golang.org/pkg/math/#Pow).

Для вычисления квадратного корня воспользуйтесь [функцией math.Sqrt](https://golang.org/pkg/math/#Sqrt).

Задание 6. Счастливый билет

Что нужно сделать

Напишите программу, в которую пользователь будет вводить четырёхзначный номер билета, а программа будет выводить, является ли он зеркальным, счастливым или обычным билетом.

Советы и рекомендации

При решении задачи необходимо применить целочисленное деление и остаток от деления. Примеры вывода программы:

1234 -> обычный билет

3425 -> счастливый билет

1221 -> зеркальный билет

Задание 7 (по желанию). Игра «Угадай число»

Что нужно сделать

Ну и какой же компьютер без игр? Давайте научим его играть в «Угадай число». Пользователь загадывает число от 1 до 10 (включительно). Программа пытается это число угадать, для этого она выводит число, а пользователь должен ответить: угадала программа, больше загаданное число или меньше.

Рекомендация

Программа не должна делать больше четырёх попыток в процессе угадывания.

**3.Введение в циклы: For как единая точка входа**

#### Задание 1. Написание простого цикла

##### Что нужно сделать

Напишите программу, которая будет принимать от пользователя произвольное число и в цикле выводить на экран все значения от нуля до указанного числа.

#### Задание 2. Сумма двух чисел по единице

##### Что нужно сделать

Напишите программу, которая запрашивает у пользователя два числа и складывает их в цикле следующим образом: берёт первое число и прибавляет к нему по единице, пока не достигнет суммы двух чисел.

#### Задание 3. Расчёт суммы скидки

##### Что нужно сделать

Напишите программу, которая принимает на вход цену товара и скидку. Посчитайте и верните на экран сумму скидки. Скидка должна быть не больше 30% от цены товара и не больше 2000 рублей.

#### Задание 4. Предыдущее занятие на if

##### Что нужно сделать

Есть три переменные со значениями 0. Первую нужно наполнить до 10, вторую — до 100, третью — до 1000. Напишите цикл, в котором эти переменные будут увеличиваться на один.

##### Рекомендация

Используйте условия для пропуска переменных, которые уже достигли своих лимитов.

#### Задание 5 (по желанию). Задача на постепенное наполнение корзин разной ёмкости (if и continue и break)

##### Что нужно сделать

Представьте, что у вас есть три корзины разной ёмкости. Пользователю предлагается ввести, какое количество яблок помещается в каждую корзину. После этого программа должна заполнить все корзины по очереди, учитывая, какие корзины уже заполнены, строго соблюдая очерёдность заполнения и добавляя по одному яблоку в каждой итерации.

##### Советы и рекомендации

Используйте if и continue и break.

Пример: пользователь решил, что у корзин будет ёмкость на шесть, четыре и девять яблок. Программа должна заполнить корзину 1 и в следующей итерации перейти к корзине 2, далее в следующей итерации перейти к корзине 3. Если очередная корзина уже заполнена, программа должна переходить к следующей по очереди, и так по кругу, пока не заполнит все.

#### Задание 6 (по желанию). Движение лифта

##### Что нужно сделать

В доме — 24 этажа. Лифт должен ходить вверх-вниз, пока не доставит всех пассажиров на первый этаж. Три пассажира ждут на четвёртом, седьмом и десятом этажах. При движении вверх лифт не должен останавливаться, при движении вниз — должен собирать всех, но не более двух человек в лифте. При этом лифт каждый раз доезжает до самого верхнего этажа и только после этого начинает движение вниз. Напишите программу, которая доставит всех пассажиров на первый этаж.

**4.Вложенные циклы**

Задача

Цель практической работы

Закрепление на практике полученных знаний о работе с вложенными циклами.

Что входит в работу

1. Зеркальные билеты.
2. Шахматная доска.
3. Вывод ёлочки.
4. По желанию. Счастливые билеты.

Задание по желанию — необязательное, повышенной сложности, требует самостоятельного изучения дополнительного материала.

Задание 1. Зеркальные билеты

Что нужно сделать

Выведите, сколько зеркальных билетов находится среди всех шестизначных номеров от 100000 до 999999.

Советы и рекомендации

Для зеркального отображения числа используйте цикл.

Пример работы программы:

Количество зеркальных билетов среди всех шестизначных номеров   
от 100000 до 999999:  
900

Задание 2. Шахматная доска

Что нужно сделать

Одним из видов компьютерного искусства является псевдографика, когда из символов создаются картины. Попробуйте вывести два изображения. Запросите у пользователя размер шахматной доски в клетках и выведите шахматную доску на экран. Белые поля выведите звёздочкой, а чёрные — пробелом.

Советы и рекомендации

Можно использовать fmt.Print и fmt.Println.

Пример работы программы:

Шахматная доска.  
Введите ширину:  
5  
Введите высоту:  
4  
 \* \*  
\* \* \*  
 \* \*  
\* \* \*

Задание 3. Вывод ёлочки

Что нужно сделать

Усложним задачу рисования: попробуйте вывести ёлочку. В первой строке выведите одну звёздочку, во второй — на две больше, в третьей — ещё на две больше, и так до количества строк, указанных пользователем.

Правила вывода ёлочки: она симметрична, количество строк соответствует введённому пользователем.

Советы и рекомендации

Необходимо сначала выводить пробелы, а затем — звёздочки. Посмотрите, как количество пробелов и звёздочек в каждой строке связано с введённым количеством строк и номером текущей строки. Внутри цикла по строкам, скорее всего, понадобятся два цикла: один — для вывода пробелов, второй — для вывода звёздочек.

Пример работы программы:

Вывод ёлочки.  
Введите высоту ёлочки:  
5  
    \*  
   \*\*\*  
  \*\*\*\*\*  
 \*\*\*\*\*\*\*  
\*\*\*\*\*\*\*\*\*

Задание 4 (по желанию). Счастливые билеты

Что нужно сделать

Помогите человеку, желающему ездить только со счастливым билетом (сумма первых трёх цифр должна быть равна сумме последних трёх цифр). Для этого он готов покупать любое количество билетов, но, с одной стороны, он не хочет переплачивать, а с другой — он хочет быть твёрдо уверен, что среди купленных билетов будет счастливый. Необходимо написать программу, которая для билетов в диапазоне от 100000 до 999999 выведет, сколько минимум нужно купить билетов, чтобы среди них оказался счастливый, если учесть, что номер текущего билета мы не знаем. Иными словами, нам надо найти максимальное расстояние между счастливыми билетами. Так, после 100001 до следующего счастливого числа — 100010 — придётся купить 9 билетов, а между 400220 и следующим счастливым билетом — 400301 — расстояние будет уже в 81 билет.

Советы и рекомендации

Необходимо запоминать предыдущий счастливый билет и максимальное расстояние, найденное ранее. При нахождении очередного счастливого билета необходимо находить расстояние до предыдущего и сравнивать его с максимальным. Если новое больше, то запоминать его. В любом случае текущее счастливое число нужно запоминать, как и предыдущее, и проверять числа дальше, до следующего счастливого.

Пример работы программы:

Минимальное количество билетов, которое нужно купить,   
чтобы среди них оказался счастливый: 1001

**5.Оператор switch**

Задача

#### Цель практической работы

Закрепить на практике полученные знания об операторах switch и fallthrough.

#### Что входит в работу

1. Времена года.
2. Дни недели.
3. По желанию. Расчёт сдачи.

Задание по желанию — необязательная, повышенной сложности, требует самостоятельного изучения дополнительного материала.

#### Задание 1. Времена года

##### Что нужно сделать

Пользователь вводит месяц, программа должна вывести, на какое время года (зиму, весну, лето, осень) этот месяц выпадает.

Как группировать:

* декабрь, январь, февраль — зима;
* март, апрель, май — весна;
* июнь, июль, август — лето;
* сентябрь, октябрь, ноябрь — осень.

##### Рекомендация

Пример работы программы:

Времена года.  
Введите месяц:  
январь  
Время года — зима.

### **Задание 2. Дни недели**

##### Что нужно сделать

Пользователь вводит будний день недели в сокращённой форме (пн, вт, ср, чт, пт) и получает развёрнутый список всех последующих рабочих дней, включая пятницу.

##### Рекомендация

Пример работы программы:

Дни недели.  
Введите будний день недели: пн, вт, ср, чт, пт:  
вт  
вторник  
среда  
четверг  
пятница

### **Задание 3 (по желанию). Расчёт сдачи**

##### Что нужно сделать

Напишите функцию, которая посчитает, сможет ли продавец в киоске обслужить всех покупателей. В киоске каждый лимонад стоит пять долларов. Клиенты стоят в очереди, чтобы купить у вас, и заказывают по одному лимонаду. Каждый покупатель может купить только один лимонад и заплатить купюрами номиналом 5, 10 или 20 долларов. Вы должны дать каждому покупателю сдачу с его купюры.

Обратите внимание, что сначала у вас нет сдачи.

##### Советы и рекомендации

Сигнатура функции lemonadeChange(bills []int) bool,

где bills — это купюры, которые мы получаем от покупателей, по одной купюре от каждого.

Верните true, если вы можете предоставить каждому покупателю правильную сдачу.

Пример 1

Расчёт сдачи.  
Ввод: [5,5,5,10,20]  
Вывод: true

Пояснение:

От первых трёх клиентов мы собираем три купюры по 5 долларов по порядку.

От четвёртого покупателя мы получаем купюру номиналом 10 долларов и возвращаем 5 долларов сдачи.

От пятого клиента мы получаем 20 долларов и возвращаем купюры номиналом 10 и 5 долларов.

Поскольку все клиенты получили правильную сдачу, мы выводим true.

Пример 2

Расчёт сдачи.  
Ввод: [10,10]  
Вывод: false

Пояснение:

Поскольку купюр в кассе нет, мы не можем дать сдачу первому покупателю.

Пример 3

Расчёт сдачи.  
Ввод: [5,5,10,10,20]  
Вывод: false

Пояснение:

От первых двух клиентов по порядку мы получаем две купюры по 5 долларов.

Для следующих двух клиентов мы собираем купюры по 10 долларов и возвращаем 5 долларов третьему и четвёртому клиентам.

Последнему покупателю мы не можем вернуть сдачу в размере 15 долларов, потому что у нас есть только две купюры по 10 долларов.

Поскольку не все покупатели получили правильную сдачу, ответ — false.

**6.Диапазоны значений**

Задача

Цель практической работы

Закрепить на практике полученные знания о диапазонах, переполнении и конвертации целочисленных типов.

Что входит в работу

1. Переполнение.
2. Минимальный тип данных.

Задание 1. Переполнение

Что нужно сделать

В данном модуле мы рассмотрели примеры по целочисленным типам, их размерам в памяти и то, что происходит при её переполнении. Напишите программу, которая в цикле с использованием встроенных констант (на предельные значения целых чисел, в пакете math) будет подсчитывать, сколько приходится переполнений чисел типа uint8, uint16 в диапазоне от 0 до uint32.

Советы и рекомендации

Для нахождения количества переполнений используйте цикл. Воспользуйтесь константами максимальных значений из пакета math.

Задание 2. Минимальный тип данных

Что нужно сделать

Достаточно часто при передаче по Сети или сохранении больших объёмов данных приходится выбирать тип с минимальным размером памяти, чтобы экономить трафик или место на диске. Напишите программу, в которую пользователь вводит два числа (int16), а программа выводит, в какой минимальный тип данных можно сохранить результат умножения этих чисел.

Советы и рекомендации

Обратите внимание, что положительный результат можно сохранить в меньшем типе за счёт использования uint8, uint16. Чтобы не возникло проблем с переполнением в процессе умножения, числа считывайте в int16, а перед умножением переведите их в int32.

Проверить на примерах:

1 1 результат uint8

1 −1 результат int8

640 100 результат uint16

−640 100 результат int32

3000 3000 результат uint32

−3000 3000 результат int32

**7.Числа с плавающей запятой. Математические вычисления**

Задача

Цель практической работы

Закрепить на практике полученные знания о числах с плавающей точкой, конверсии и преобразовании типов.

Что входит в работу

1. Разложение ex в ряд Тейлора.
2. Проблемы округления процентов.

Задание 1. Разложение ex в ряд Тейлора

Что нужно сделать

Напишите программу, вычисляющую ex посредством разложения в ряд Тейлора с заданной пользователем точностью. Пользователь вводит значение x и с точностью до какого знака после запятой необходимо вычислить значение функции.

Советы и рекомендации

Получить значение точности (эпсилон) можно, разделив 1 на 10, возведённую в степень, равную количеству знаков после запятой.

Для x = 0 и n = 1 вывод должен быть 1

Для x = 1 и n = 3 вывод должен быть 2,7182539682539684

Для x = 1 и n = 5 вывод должен быть 2,7182815255731922

Задание 2. Проблемы округления процентов

Что нужно сделать

В связи с особенностями хранения дробных чисел они не очень подходят для хранения денежных значений (может потеряться значимая часть при переполнении, да и дробная часть больше двух знаков не нужна). Но мы попробуем решить задачу начисления процентов, используя именно их.

* Пусть пользователь вводит сумму, которую он кладёт в банк, ежемесячный процент капитализации и количество лет, в течение которых будет открыт вклад.
* Необходимо ежемесячно пересчитывать сумму, округляя её до целого количества копеек в меньшую сторону. Например, если после начисления процентов остаток включает 35,78 копейки, то оставляем только 35 копеек, а дробную часть отбрасываем.
* По окончании работы программы необходимо вывести, какую сумму получит вкладчик на руки и какая сумма будет зачислена в пользу банка за счёт округления копеек.

Советы и рекомендации

Для округления копеек можно умножить получившееся после капитализации процентов число на 100, округлить в меньшую сторону, затем поделить опять на 100. Отбрасываемую часть можно получить вычитанием текущего значения остатка на счёте и того, который был до округления.

Для 1000 рублей, 1% и одного года программа должна вывести 1126,78 и 0,04350000000022192 (возможно меньше знаков).

Для 1000 рублей, 1% и десяти лет программа должна вывести 3299,41 и 0,5216000000013992 (возможно меньше знаков).

**8.Комплексные типы данных. Строки**

Задача

Цель практической работы

Закрепить на практике полученные знания работы со строками и приведением типа.

Что входит в работу

1. Определение количества слов, начинающихся с большой буквы.
2. Вывод чисел, содержащихся в строке.

Задание 1. Определение количества слов, начинающихся с большой буквы

Что нужно сделать

Напишите программу, которая выведет количество слов, начинающихся с большой буквы в строке: Go is an Open source programming Language that makes it Easy to build simple, reliable, and efficient Software.

Рекомендация

Пример работы программы:

Определение количества слов, начинающихся с большой буквы в строке:  
Go is an Open source programming Language that makes it Easy   
to build simple, reliable, and efficient Software  
Строка содержит 5 слов с большой буквы.

Задание 2. Вывод чисел, содержащихся в строке

Что нужно сделать

Напишите программу, которая выведет все части строки

a10 10 20b 20 30c30 30 dd,

которые можно привести к числу в десятичном формате.

Рекомендация

Пример работы программы:

Исходная строка:  
a10 10 20b 20 30c30 30 dd  
В строке содержатся числа в десятичном формате:  
10 20 30

**9.Работа с файлами**

Задача

#### Цель практической работы

Закрепить на практике полученные знания работы с файловой системой, пакетами os, io, ioutil, интерфейсом io.Reader и уровнями доступа.

#### Что входит в работу

1. Работа с файлами.
2. Интерфейс io.Reader.
3. Уровни доступа.
4. Пакет ioutil.
5. Комбинации круглых скобок.\*

Задания со “\*” выполняется по желанию — необязательное, повышенной сложности, требует самостоятельного изучения дополнительного материала.

#### Задание 1. Работа с файлами

##### Что нужно сделать

Напишите программу, которая на вход получала бы строку, введённую пользователем, а в файл писала № строки, дату и сообщение в формате:

2020-02-10 15:00:00 продам гараж.

При вводе слова exit программа завершает работу.

#### Задание 2. Интерфейс io.Reader

##### Что нужно сделать

Напишите программу, которая читает и выводит в консоль строки из файла, созданного в предыдущей практике, без использования ioutil. Если файл отсутствует или пуст, выведите в консоль соответствующее сообщение.

##### Рекомендация

Для получения размера файла воспользуйтесь методом Stat(), который возвращает информацию о файле и ошибку.

#### Задание 3. Уровни доступа

##### Что нужно сделать

Напишите программу, создающую текстовый файл только для чтения, и проверьте, что в него нельзя записать данные.

##### Рекомендация

Для проверки создайте файл, установите режим только для чтения, закройте его, а затем, открыв, попытайтесь прочесть из него данные.

### **Задание 4. Пакет ioutil**

##### Что нужно сделать

Перепишите задачи 1 и 2, используя пакет ioutil.

### **Задание 5 (по желанию). Комбинации круглых скобок**

##### Что нужно сделать

Напишите программу, которая на вход принимала бы интовое число и для него генерировала бы все возможные комбинации круглых скобок.

##### Рекомендация

Первый пример вывода программы:

Введите количество пар скобок:  
3  
["((()))","(()())","(())()","()(())","()()()"]

Второй пример вывода программы:

Введите количество пар скобок:  
1  
["()"]

**10.Функции в программировании**

Задача

Цель практической работы

Закрепить полученные знания о функциях и способах передачи аргументов.

Что входит в работу

1. Функция, принимающая аргументы.
2. Функция, принимающая значение по ссылке.

Задание 1. Функция, принимающая аргументы

Что нужно сделать

Напишите функцию, которая принимает в качестве аргументов два числа типа int, вычисляет сумму чётных чисел заданного диапазона и выводит результат в консоль.

Рекомендация

Если первое число, переданное в качестве аргумента, будет больше второго, просто поменяйте их местами.

Задание 2. Функция, принимающая значение по ссылке

Что нужно сделать

Напишите функцию, которая принимает в качестве аргументов указатели на два типа int и меняет их значения местами.

Рекомендация

В методе main создайте и присвойте значения двум переменным типа int, выведите значения этих переменных в консоль до вызова функции и после.

**11.Функции, возвращающие результат**

Задача

Цель практической работы

Закрепить полученные знания о функциях, возвращающих результат.

Что входит в работу

1. Функция, возвращающая результат.
2. Функция, возвращающая несколько значений.
3. Именованные возвращаемые значения.
4. Область видимости переменных.

Задание 1. Функция, возвращающая результат

Что нужно сделать

Напишите функцию, которая на вход получает число и возвращает true, если число четное, и false, если нечётное.

Рекомендация

Программа запрашивает у пользователя или генерирует случайное число, передает в функцию в качестве аргумента и выводит в консоль результат её работы.

Задание 2. Функция, возвращающая несколько значений

Что нужно сделать

Напишите программу, которая с помощью функции генерирует три случайные точки в двумерном пространстве (две координаты), а затем с помощью другой функции преобразует эти координаты по формулам: x = 2 × x + 10, y = −3 × y − 5.

Задание 3. Именованные возвращаемые значения

Что нужно сделать

Напишите программу, которая на вход получает число, затем с помощью двух функций преобразует его. Первая умножает, а вторая прибавляет число, используя именованные возвращаемые значения.

Задание 4. Область видимости переменных

Что нужно сделать

Напишите программу, в которой будет три функции, попарно использующие разные глобальные переменные. Функции должны прибавлять к поданному на вход числу глобальную переменную и возвращать результат. Затем вызовите по очереди три функции, передавая результат из одной в другую.

**12.Композитные типы данных. Одномерные массивы. Базовые алгоритмы на массивах**

Задача

Цель домашнего задания

Закрепить навыки работы с массивами.

Что входит в домашнее задание

1. Подсчитать чётные и нечётные числа в массиве.
2. Написать функцию, реверсирующую массив.

Задание 1. Подсчёт чётных и нечётных чисел в массиве

Что нужно сделать

Разработайте программу, позволяющую ввести 10 целых чисел, а затем вывести из них количество чётных и нечётных чисел. Для ввода и подсчёта используйте разные циклы.

Что оценивается

Для введённых чисел 1, 1, 1, 2, 2, 2, 3, 3, 3, 4 программа должна вывести: чётных — 4, нечётных — 6.

Задание 2. Функция, реверсирующая массив

Что нужно сделать

* Напишите функцию, принимающую на вход массив и возвращающую массив, в котором элементы идут в обратном порядке по сравнению с исходным.
* Напишите программу, демонстрирующую работу этого метода.

Что оценивается

При вводе 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 программа должна выводить при помощи дополнительной функции, реверсировав массив: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

**13.Простые алгоритмы на массивах**

Задача

Цель домашнего задания

Научиться работать с одномерными массивами.

Что входит в домашнее задание

1. Написать функцию, которая производит слияние отсортированных массивов.
2. Отсортировать массив пузырьком.

Задание 1. Слияние отсортированных массивов

Что нужно сделать

Напишите функцию, которая производит слияние двух отсортированных массивов длиной четыре и пять в один массив длиной девять.

Советы и рекомендации

1. Обратите внимание на размеры массивов.
2. В качестве среды разработки может помочь GoLand или VS Code.

Что оценивается

* Правильность размеров.
* Правильный порядок элементов в конечном массиве.

Как отправить задание на проверку

Выполните задание в файле Go. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на файл через форму для сдачи домашнего задания.

Или отправьте файл через онлайн-редактор REPL, или архивом.

Задание 2. Сортировка пузырьком

Что нужно сделать

Отсортируйте массив длиной шесть пузырьком.

Советы и рекомендации

* Принцип сортировки пузырьком можно посмотреть [на «Википедии»](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BF%D1%83%D0%B7%D1%8B%D1%80%D1%8C%D0%BA%D0%BE%D0%BC), там есть наглядная демонстрация, или на YouTube.
* В качестве среды разработки может помочь GoLand или VS Code.

**14.Двумерные массивы**

Задача

Цель домашнего задания

Научиться работать с двумерными массивами.

Что входит в задание

1. Написать функцию, вычисляющую определитель.
2. Написать функцию, умножающую две матрицы.

Задание 1. Подсчёт определителя

Что нужно сделать

Напишите функцию, вычисляющую определитель матрицы размером 3 × 3.

Советы и рекомендации

* Алгоритм вычисления представлен в [«Википедии»](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C).
* В качестве среды разработки может помочь Goland или VScode.

Что оценивается

* Правильность ответа.
* Проект собирается без ошибок.

Как отправить задание на проверку

Выполните задание в файле GO. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на документ через форму для сдачи домашнего задания.

Задание 2. Умножение матриц

Что нужно сделать

Напишите функцию, умножающую две матрицы размерами 3 × 5 и 5 × 4.

Советы и рекомендации

* Алгоритм умножения матриц проиллюстрирован в [«Википедии»](https://ru.wikipedia.org/wiki/%D0%A3%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86).
* В качестве среды разработки может помочь Goland или VScode.

Что оценивается

* Правильность размеров исходных и конечной матрицы.
* Арифметическая правильность ответа.

**15.Анонимные функции**

Задача

Цель домашнего задания

Научиться работать с преобразованием типов и анонимными функциями.

Что входит в задание

1. Написать функцию для расчётов по формуле в задании.
2. Написать функцию, которая на вход принимает анонимную функцию, а внутри оборачивает её в defer.

Задание 1. Расчёт по формуле

Что нужно сделать

Напишите функцию, производящую следующие вычисления.

S = 2 × x + y ^ 2 − 3/z, где x — int16, y — uint8, a z — float32.

Тип S должен быть во float32.

Советы и рекомендации

* Обратите внимание, к какому типу надо привести конечный результат.
* В качестве среды разработки может помочь Goland или VScode.

Что оценивается

* Правильность ответа.
* Проект собирается без ошибок.

Как отправить задание на проверку

Выполните задание в файле GO. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на документ через форму для сдачи домашнего задания.

Задание 2. Анонимные функции

Что нужно сделать

Напишите функцию, которая на вход принимает функцию вида A func (int, int) int, а внутри оборачивает и вызывает её при выходе (через defer).

Вызовите эту функцию с тремя разными анонимными функциями A. Тела функций могут быть любыми, но главное, чтобы все три выполняли разное действие.

Рекомендация

В качестве среды разработки может помочь Goland или VScode.

Что оценивается

* Правильность размеров исходных и конечной матриц.
* Арифметическая правильность ответа.

**16.Продвинутые алгоритмы на массивах. Поиск**

Задача

Цель домашнего задания

Закрепить навыки работы с массивами.

Что входит в задание

1. Посчитать числа в неупорядоченном массиве после заданного числа.
2. Найти первое вхождение числа в упорядоченном массиве (числа могут повторяться).

Задание 1. Подсчёт чётных и нечётных чисел в массиве

Что нужно сделать

Заполните массив неупорядоченными числами на основе генератора случайных чисел. Введите число. Программа должна найти это число в массиве и вывести, сколько чисел находится в массиве после введённого. При отсутствии введённого числа в массиве — вывести 0. Для удобства проверки реализуйте вывод массива на экран.

Что оценивается

Программа должна корректно считать числа в массиве после заданного.

Как отправить задание на проверку

Выполните задание в файле Go. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на файл через форму для сдачи домашнего задания.

Либо отправьте файл через онлайн-редактор REPL или архивом.

Задание 2. Нахождение первого вхождения числа в упорядоченном массиве (числа могут повторяться)

Что нужно сделать

Заполните упорядоченный массив из 12 элементов и введите число. Необходимо реализовать поиск первого вхождения заданного числа в массив. Сложность алгоритма должна быть минимальная.

Что оценивается

Верность индекса.

При вводе массива 1 2 2 2 3 4 5 6 7 8 9 10 и вводе числа 2 программа должна вывести индекс 1.

**17.Сортировка с пониженной алгоритмической сложностью**

Задача

Цель домашнего задания

Освоить использование динамических массивов.

Что входит в задание

1. Написать функцию для разделения чётных и нечётных чисел в массиве.
2. Написать функцию, которая ищет в нескольких предложениях символ в последнем слове.

Задание 1. Чётные и нечётные

Что нужно сделать

Напишите функцию, которая принимает массив чисел, а возвращает два массива: один из чётных чисел, второй из нечётных.

Рекомендация

В качестве среды разработки используйте GoLand или VS Code.

Что оценивается

* Правильность ответа.
* Отсутствие ошибок при сборке проекта.

Как отправить задание на проверку

Выполните задание в файле Go. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на файл через форму для сдачи домашнего задания.

Или отправьте файл через онлайн-редактор REPL, или архивом.

Задание 2. Поиск символов в нескольких строках

Что нужно сделать

Напишите функцию, которая на вход принимает массив предложений (длинных строк) и массив символов типа rune, а возвращает 2D-массив, где на позиции [i][j] стоит индекс вхождения символа j из chars в последнее слово в предложении i (строку надо разбить на слова и взять последнее). То есть сигнатура следующая:

func parseTest(sentences []string, chars []rune)

Советы и рекомендации

* В качестве среды разработки используйте Goland или VScode.
* Не забудьте проверить, что вы получили больше чем 0 аргументов.
* Подход не важен: можно переписать сортировку пузырьком или отсортировать, а потом перевернуть.
* Пример входных данных

sentences := [4]string{"Hello world", "Hello Skillbox", "Привет Мир", "Привет Skillbox"}

chars := [5]rune{'H','E','L','П','М'}

Пример вывода результата в первом элементе массива

'H' position 0

'E' position 1

'L' position 9

**18.Композитные типы данных. Срезы. Динамическое выделение памяти**

Задача

Цель домашнего задания

Научиться алгоритму сортировки вставками и неограниченному числу аргументов.

Что входит в домашнее задание

1. Написать функцию для сортировки вставками массива.
2. Написать функцию, принимающую на вход неограниченное число переменных типа int и сортирующую этот массив пузырьков в обратном порядке.

Задание 1. Сортировка вставками

Что нужно сделать

Напишите функцию, сортирующую массив длины 10 вставками.

Советы и рекомендации

* Алгоритм сортировки доступен на [«Википедии»](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0_%D0%B2%D1%81%D1%82%D0%B0%D0%B2%D0%BA%D0%B0%D0%BC%D0%B8).
* В качестве среды разработки используйте GoLand или VSCode.

Что оценивается

* Правильность ответа.
* Проект собирается без ошибок.

Как отправить задание на проверку

Выполните задание в файле GO. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на документ через форму для сдачи домашнего задания.

Задание 2. Анонимные функции

Что нужно сделать

Напишите анонимную функцию, которая на вход получает массив типа integer, сортирует его пузырьком и переворачивает (либо сразу сортирует в обратном порядке, как посчитаете нужным).

**19.Подмножества и работа со срезами**

Задача

Цель задания

Написать программу для нахождения подстроки в кириллической подстроке. Программа должна запускаться с помощью команды:

 go run main.go --str "строка для поиска" --substr "поиска"

Для реализации такой работы с флагами воспользуйтесь пакетом flags, а для поиска подстроки в строке вам понадобятся руны.

Что нужно сделать

1. Спроектировать алгоритм поиска подстроки.
2. Определить строку и подстроку, используя флаги.
3. Написать алгоритм реализацию для работы со строками UTF-8 (для этого необходимо воспользоваться рунами).

Что оценивается

* Алгоритм может работать с различными символами (кириллица, китайские иероглифы).
* Использованы руны.

Как отправить задание на проверку

Выполните задание в файле Go. Загрузите файл на Google Диск, откройте доступ для всех по ссылке. Отправьте ссылку на файл через форму для сдачи домашнего задания.

Или отправьте файл через онлайн-редактор REPL, или архивом.

|  |  |
| --- | --- |
| Input | привет мир!, вет |
| Output | true |

|  |  |
| --- | --- |
| Input | Программирование - это просто, вание |
| Output | true |

|  |  |
| --- | --- |
| Input | Программирование - это просто, корабль |
| Output | false |

**20.Строки и текстовые данные. Базовые алгоритмы со строками**

Задача

Цель задания

Написать программу аналог cat.

Что нужно сделать

Программа должна получать на вход имена двух файлов, необходимо  конкатенировать их содержимое, используя strings.Join.

Что оценивается

* При получении одного файла на входе программа должна печатать его содержимое на экран.
* При получении двух файлов на входе программа соединяет их и печатает содержимое обоих файлов на экран.
* Если программа запущена командой **go run firstFile.txt secondFile.txt resultFile.txt**, то она должна написать два соединённых файла в результирующий.

Общие условия

Разработка выполняется в среде golang или vs code.

|  |  |
| --- | --- |
| first.txt | контент первого файла |
| second.txt | контент второго файла |
| result .txt | контент первого файла  контент второго файла |
| Input | go run first.txt second.txt result.txt |

|  |  |
| --- | --- |
| first.txt | контент первого файла |
| second.txt | контент второго файла |
| Input | go run first.txt second.txt |
| Output | контент первого файла  контент второго файла |

**21.Композитные типы данных. Карты и структуры**

Задача

Цель задания

Научиться работать с композитными типами данных: структурами и картами

Что нужно сделать

Напишите программу, которая считывает ввод с stdin, создаёт структуру student и записывает указатель на структуру в хранилище map[studentName] \*Student.

type Student struct {  
  
name string  
  
age int  
  
grade int   
  
}

Программа должна получать строки в бесконечном цикле, создать структуру Student через функцию newStudent, далее сохранить указатель на эту структуру в map, а после получения EOF (ctrl + d) вывести на экран имена всех студентов из хранилища. Также необходимо реализовать методы put, get.

Общие условия

Разработка выполняется в среде golang или vs code.

|  |  |
| --- | --- |
| Input | go run main.go |
| Строки | Вася 24 1  Семен 32 2 |
| EOF |  |
| Output | Студенты из хранилища:  Вася 24 1  Семен 32 2 |

Критерии оценки

Зачёт:

* при получении одной строки (например, «имяСтудента 24 1») программа создаёт студента и сохраняет его, далее ожидает следующую строку или сигнал EOF (Сtrl + Z);
* при получении сигнала EOF программа должна вывести имена всех студентов из map.

**22.Переиспользование кода, модульность**

Задача

Цель задания

Научиться декомпозировать и рефакторить код на примере программы, написанной в прошлом уроке.

Что нужно сделать

В прошлом домашнем задании вы написали программу для работы со студентами. Мы указываем в стандартный ввод данные о студенте, а после получения сигнала об окончании работы программы она выводит имена всех студентов на экран.

Вам нужно отрефакторить код прошлого домашнего задания. Декомпозируйте его так, чтобы логике одной сущности соответствовал один пакет.

Для того, чтобы вы могли использовать методы и переменные, которые объявлены в других пакетах, сделайте их экспортируемыми.

Структура программы после рефакторинга может выглядеть следующим образом:



Критерии оценки

Зачёт:

* программа работает в прежнем режиме, но декомпозирована на пакеты;
* при получении одной строки (например, «имяСтудента 24 1») программа создаёт данные о студенте и сохраняет их, далее ожидает следующую строку или сигнал EOF (Сtrl + d);
* при получении сигнала EOF программа должна вывести имена всех студентов из map.

**23.Многопоточность, работа с каналами**

Задача

Задание 1. Конвейер

Цели задания

* Научиться работать с каналами и горутинами.
* Понять, как должно происходить общение между потоками.

Что нужно сделать

Реализуйте паттерн-конвейер:

1. Программа принимает числа из стандартного ввода в бесконечном цикле и передаёт число в горутину.
2. Квадрат: горутина высчитывает квадрат этого числа и передаёт в следующую горутину.
3. Произведение: следующая горутина умножает квадрат числа на 2.
4. При вводе «стоп» выполнение программы останавливается.

Советы и рекомендации

Воспользуйтесь небуферизированными каналами и waitgroup.

Что оценивается

Ввод : 3

Квадрат : 9

Произведение : 18

Как отправить задание на проверку

Выполните задание в файле вашей среды разработки и пришлите ссылку на архив с вашим проектом через форму ниже.

Задание 2. Graceful shutdown

Цель задания

Научиться правильно останавливать приложения.

Что нужно сделать

В работе часто возникает потребность правильно останавливать приложения. Например, когда наш сервер обслуживает соединения, а нам хочется, чтобы все текущие соединения были обработаны и лишь потом произошло выключение сервиса. Для этого существует паттерн graceful shutdown.

Напишите приложение, которое выводит квадраты натуральных чисел на экран, а после получения сигнала ^С обрабатывает этот сигнал, пишет «выхожу из программы» и выходит.

Советы и рекомендации

Для реализации данного паттерна воспользуйтесь каналами и оператором select с default-кейсом.

Что оценивается

Код выводит квадраты натуральных чисел на экран, после получения ^С происходит обработка сигнала и выход из программы.

**24.Обмен данными по сети**

Задача

Цель практической работы

Научиться:

* работать с запросами POST, GET, PUT, DELETE;
* применять принципы написания обработчиков HTTP-запросов.

Что нужно сделать

Напишите HTTP-сервис, который принимает входящие соединения с JSON-данными и обрабатывает их следующим образом:

    1. Сделайте обработчик создания пользователя. У пользователя должны быть следующие поля: имя, возраст и массив друзей. Пользователя необходимо сохранять в мапу. Пример запроса:

POST /create HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Host: localhost:8080  
{"name":"some name","age":"24","friends":[]}

Данный запрос должен возвращать ID пользователя и статус 201.

    2. Сделайте обработчик, который делает друзей из двух пользователей. Например, если мы создали двух пользователей и нам вернулись их ID, то в запросе мы можем указать ID пользователя, который инициировал запрос на дружбу, и ID пользователя, который примет инициатора в друзья. Пример запроса:

    POST /make\_friends HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Host: localhost:8080  
{"source\_id":"1","target\_id":"2"}

Данный запрос должен возвращать статус 200 и сообщение «username\_1 и username\_2 теперь друзья».

    3. Сделайте обработчик, который удаляет пользователя. Данный обработчик принимает ID пользователя и удаляет его из хранилища, а также стирает его из массива friends у всех его друзей. Пример запроса:

    DELETE /user HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Host: localhost:8080  
{"target\_id":"1"}

Данный запрос должен возвращать 200 и имя удалённого пользователя.

    4. Сделайте обработчик, который возвращает всех друзей пользователя. Пример запроса:

    GET /friends/user\_id HTTP/1.1  
Host: localhost:8080  
Connection: close

После /friends/ указывается id пользователя, друзей которого мы хотим увидеть.

    5. Сделайте обработчик, который обновляет возраст пользователя. Пример запроса:

PUT /user\_id HTTP/1.1  
Content-Type: application/json; charset=utf-8  
Host: localhost:8080  
{"new age":"28"}

Запрос должен возвращать 200 и сообщение «возраст пользователя успешно обновлён».

**25.Микросервисная архитектура**

Задача

Цель практической работы

Научиться:

* писать микросервис и proxy,
* тестировать написанное приложение.

Что нужно сделать

В прошлом домашнем задании вы писали приложение, которое принимает HTTP-запросы, создаёт пользователей, добавляет друзей и так далее. Давайте теперь приблизим наше приложение к реальному продукту.

* Отрефакторьте приложение так, чтобы вы могли поднять две реплики данного приложения.
* Используйте любую базу данных, чтобы сохранять информацию о пользователях, **или** можете сохранять информацию в файл, предварительно сереализуя в JSON.
* Напишите proxy или используйте, например, nginx.
* Протестируйте приложение.