

GoLang Assignment

By Suteerth Subramaniam

- [Main Repo](#)
- [Golang Exercism Learning](#)

Hello World

The screenshot shows the Exercism.io interface for the 'Hello World' exercise. At the top, a green banner says 'You've completed Hello World.' with a checkmark icon. To the right is a 'Discover more exercises' button. Below the banner, the exercise title 'Hello World' is displayed with a globe icon and a checkmark. There are two buttons: 'Completed' (with a checkmark) and 'Tutorial Exercise'. A navigation bar includes 'Overview' (selected), 'Your iterations 1', and 'Open in editor'. The main area shows 'Iteration 1' (Latest) submitted via Editor 7 hours ago. The code file 'hello_world.go' contains the following Go code:

```
1 package greeting
2
3 // HelloWorld greets the world.
4 func HelloWorld() string {
5     return "Hello, World!"
6 }
```

To the right of the code, there's an 'Analysis' section with a green dot and 'Passed', and a 'Tests' section. Below these is a cartoon illustration of two people working on a laptop with a lightbulb icon. A callout text says 'No auto suggestions? Try human mentoring.'

Gropher's Lasagna

 **Gopher's Gorgeous Lasagna** ✓

Published Learning Exercise

Overview Your iterations 1 Community Solutions Code Review Open in editor

Iteration 1 Latest Published
Submitted via Editor, 16 minutes ago

Passed

Analysis **Tests**



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Gopher's Gorgeous Lasagna exercise and start writing better Go.

Get mentoring

```
lasagna.go
1 package lasagna
2
3 // TODO: define the 'OvenTime' constant
4 const OvenTime = 40
5 // RemainingOvenTime returns the remaining minutes based on the 'actual' minutes already in the
6 func RemainingOvenTime(actualMinutesInOven int) int {
7     return OvenTime - actualMinutesInOven;
8 }
9
10 // PreparationTime calculates the time needed to prepare the lasagna based on the amount of layers
11 func PreparationTime(numberOfLayers int) int {
12     return 2*numberOfLayers;
13 }
14
15 // ElapsedTime calculates the time elapsed cooking the lasagna. This time includes the preparation
16 func ElapsedTime(numberOfLayers, actualMinutesInOven int) int {
17     return PreparationTime(numberOfLayers) + actualMinutesInOven;
18 }
19
```

Weather Forecast

 **Weather Forecast** ✓

Completed Learning Exercise

Overview Your iterations 1 Community Solutions Code Review Open in editor

Iteration 1 Latest
Submitted via Editor, a few seconds ago

Passed

Analysis **Tests**



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Weather Forecast exercise and start writing better Go.

Get mentoring

```
weather_forecast.go
1 // Package weather provides tools to perform weather forecasts.
2 package weather
3
4 // CurrentCondition represents the weather at the time.
5 var CurrentCondition string
6 // CurrentLocation represents the location in question.
7 var CurrentLocation string
8
9 // Forecast returns a string describing the weather forecast of a particular location taking in
10 func Forecast(city, condition string) string {
11     CurrentLocation, CurrentCondition = city, condition
12     return CurrentLocation + " - current weather condition: " + CurrentCondition
13 }
14
```

Cars Assemble

Iteration 1 Latest
Submitted via Editor, a few seconds ago

cars_assemble.go

```
1 package cars
2 // CalculateWorkingCarsPerHour calculates how many working cars are
3 // produced by the assembly line every hour.
4 func CalculateWorkingCarsPerHour(productionRate int, successRate float64) float64 {
5     return float64(productionRate) * successRate * 0.01;
6 }
7
8 // CalculateWorkingCarsPerMinute calculates how many working cars are
9 // produced by the assembly line every minute.
10 func CalculateWorkingCarsPerMinute(productionRate int, successRate float64) int {
11     var productionRatePerMinute float64 = float64(productionRate) / 60.0;
12     return int(productionRatePerMinute * successRate * 0.01);
13 }
14
15 // CalculateCost works out the cost of producing the given number of cars.
16 func CalculateCost(carsCount int) uint {
17     var groupsOfTen int = carsCount / 10;
18     var individualCars int = carsCount % 10;
19     return uint((95000 * groupsOfTen) + (10000 * individualCars));
20 }
21
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Cars Assemble exercise and start writing better Go.

Get mentoring

Annalyn's Infiltration

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

annalyns_infiltration.go

```
1 package annalyn
2
3 // CanFastAttack can be executed only when the knight is sleeping.
4 func CanFastAttack(knightIsAwake bool) bool {
5     return !knightIsAwake;
6 }
7
8 // CanSpy can be executed if at least one of the characters is awake.
9 func CanSpy(knightIsAwake, archerIsAwake, prisonerIsAwake bool) bool {
10    return knightIsAwake || archerIsAwake || prisonerIsAwake;
11 }
12
13 // CanSignalPrisoner can be executed if the prisoner is awake and the archer is sleeping.
14 func CanSignalPrisoner(archerIsAwake, prisonerIsAwake bool) bool {
15    return prisonerIsAwake && !archerIsAwake;
16 }
17
18 // CanFreePrisoner can be executed if the prisoner is awake and the other 2 characters are asleep
19 // or if Annalyn's pet dog is with her and the archer is sleeping.
20 func CanFreePrisoner(knightIsAwake, archerIsAwake, prisonerIsAwake, petDogIsPresent bool) bool {
21     if petDogIsPresent {
22         return !archerIsAwake;
23     } else {
24         return prisonerIsAwake && !archerIsAwake && !knightIsAwake;
25     }
26 }
27
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Annalyn's Infiltration exercise and start writing better Go.

Get mentoring

Welcome To Tech Palace!

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

• Passed

welcome_to_tech_palace.go

```
1 package techpalace
2 import "strings"
3 // WelcomeMessage returns a welcome message for the customer.
4 func WelcomeMessage(customer string) string {
5     // Using the function from strings package to convert the string to upper case before appen
6     return "Welcome to the Tech Palace, " + strings.ToUpper(customer);
7 }
8
9 // AddBorder adds a border to a welcome message.
10 func AddBorder(welcomeMsg string, numStarsPerLine int) string {
11     var res string = "";
12     // A hard and fast solution to get the stars in the message
13     for i := 0 ; i < numStarsPerLine ; i++ {
14         res += "*";
15     }
16     return res + "\n" + welcomeMsg + "\n" + res;
17 }
18
19 // CleanupMessage cleans up an old marketing message.
20 func CleanupMessage(oldMsg string) string {
21     // The oldMsg is made up of 3 sections, the middle one is what we want to extract from. Thi
22     var chunks []string = strings.Split(oldMsg, "\n");
23     // The format of this is a * in the beginning, so first we remove these from the beginning
24     var withoutStars string = strings.Trim(chunks[1], "*");
25     // Lastly, we remove the whitespaces.
26     var cleanedMessage string = strings.TrimSpace(withoutStars, " ");
27     return cleanedMessage;
28 }
29
```

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Welcome To Tech Palace! exercise and start writing better Go.

Get mentoring

Party Robot

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

• Passed

party_robot.go

```
1 package partyrobot
2 import "fmt"
3 // Welcome greets a person by name.
4 func Welcome(name string) string {
5     return fmt.Sprintf("Welcome to my party, %s!", name);
6 }
7
8 // HappyBirthday wishes happy birthday to the birthday person and exclaims their age.
9 func HappyBirthday(name string, age int) string {
10    return fmt.Sprintf("Happy birthday %s! You are now %d years old!", name, age);
11 }
12
13 // AssignTable assigns a table to each guest.
14 func AssignTable(name string, table int, neighbor, direction string, distance float64) string {
15     var paddedTable string = "";
16     if table < 10 {
17         paddedTable = fmt.Sprintf("%0*d", table);
18     } else if table >= 10 && table < 100 {
19         paddedTable = fmt.Sprintf("%0*d", table);
20     } else {
21         paddedTable = fmt.Sprintf("%d", table);
22     }
23     return fmt.Sprintf("%s\nYou have been assigned to table %s. Your table is %s, exactly %.1f
24 }
25
```

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Party Robot exercise and start writing better Go.

Get mentoring

Vehicle Purchase

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

vehicle_purchase.go

```
1 package purchase
2
3 // NeedsLicense determines whether a license is needed to drive a type of vehicle. Only "car" and "truck" require a license.
4 func NeedsLicense(kind string) bool {
5     if kind == "car" || kind == "truck" {
6         return true;
7     } else {
8         return false;
9     }
10}
11
12 // ChooseVehicle recommends a vehicle for selection. It always recommends the vehicle that comes first lexicographically.
13 func ChooseVehicle(option1, option2 string) string {
14     var lexographicallyFirst string;
15     if option1 < option2 {
16         lexographicallyFirst = option1;
17     } else {
18         lexographicallyFirst = option2;
19     }
20     return lexographicallyFirst + " is clearly the better choice.";
21}
22
23 // CalculateResellPrice calculates how much a vehicle can resell for at a certain age.
24 func CalculateResellPrice(originalPrice, age float64) float64 {
25     if age < 3 {
26         return originalPrice * 0.8;
27     } else if age >= 3 && age < 10 {
28         return originalPrice * 0.7;
29     } else {
30         return originalPrice * 0.5;
31     }
32}
33
```

Passed

Analysis Tests

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Vehicle Purchase exercise and start writing better Go.

Get mentoring



Card Tricks

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

Passed

card_tricks.go

```
1 package cards
2 // FavoriteCards returns a slice with the cards 2, 6 and 9 in that order.
3 func FavoriteCards() []int {
4     var deck []int = []int{2,6,9};
5     return deck;
6 }
7 // GetItem retrieves an item from a slice at given position.
8 // If the index is out of range, we want it to return -1.
9 func GetItem(slice []int, index int) int {
10    if index >= len(slice) || index < 0 {
11        return -1;
12    } else {
13        return slice[index];
14    }
15 }
16 // SetItem writes an item to a slice at given position overwriting an existing value.
17 // If the index is out of range the value needs to be appended.
18 func SetItem(slice []int, index, value int) []int {
19    if index < 0 || index >= len(slice) {
20        return append(slice, value)
21    }
22    slice[index] = value;
23    return slice;
24 }
25 // PrependItems adds an arbitrary number of values at the front of a slice.
26 func PrependItems(slice []int, values ...int) []int {
27    return append(values, slice...);
28 }
29 // RemoveItem removes an item from a slice by modifying the existing slice.
30 func RemoveItem(slice []int, index int) []int {
31    if index >= len(slice) || index < 0 {
32        return slice;
33    }
```

Analysis Tests

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Card Tricks exercise and start writing better Go.

Get mentoring

Blackjack

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
23     case "nine":  
24         return 9;  
25     case "ten","jack", "queen", "king":  
26         return 10;  
27     default:  
28         return 0;  
29     }  
30 }  
31 // FirstTurn returns the decision for the first turn, given two cards of the  
32 // player and one card of the dealer.  
33 func FirstTurn(card1, card2, dealerCard string) string {  
34     playerTotal := ParseCard(card1) + ParseCard(card2);  
35     dealerHand := ParseCard(dealerCard);  
36     switch {  
37         case playerTotal == 22:  
38             return "P";  
39         case playerTotal == 21:  
40             if dealerHand >= 10 {  
41                 return "S";  
42             } else {  
43                 return "W";  
44             }  
45         case playerTotal >= 17 && playerTotal <= 20:  
46             return "S";  
47         case playerTotal >= 12 && playerTotal <= 16:  
48             if dealerHand < 7 {  
49                 return "S";  
50             } else {  
51                 return "H"  
52             }  
53         default:  
54             return "H";  
55     }  
56 }  
57 }
```

Passed |

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Blackjack exercise and start writing better Go.

[Get mentoring](#)

Need For Speed

Iteration 1 Latest Published Submitted via Editor, a few seconds ago Passed

```
14     speed: speed,
15     batteryDrain: batteryDrain,
16     battery: 100,
17     distance: 0,
18   }
19   return newCar
20 }
21
22 // TODO: define the 'Track' type struct
23 type Track struct {
24   distance int
25 }
26
27 // NewTrack creates a new track
28 func NewTrack(distance int) Track {
29   newTrack := Track {
30     distance: distance,
31   }
32   return newTrack
33 }
34
35 // Drive drives the car one time. If there is not enough battery to drive one more time,
36 // the car will not move.
37 func Drive(car Car) Car {
38   if car.battery >= car.batteryDrain {
39     car.battery -= car.batteryDrain
40     car.distance += car.speed
41   }
42   return car
43 }
44
45 // CanFinish checks if a car is able to finish a certain track.
46 func CanFinish(car Car, track Track) bool {
47   if car.distance + (car.speed * int(float64(car.battery) / float64(car.batteryDrain))) >= track.distance {
48     return true
49   }
50 }
```

Analysis Tests More



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Need For Speed exercise and start writing better Go.

Get mentoring

Animal Magic

Animal Magic Published Learning Exercise

Overview Your iterations 1 Community Solutions Code Review Open in editor

Iteration 1 Latest Published Submitted via Editor, a few seconds ago Passed

```
animal_magic.go
1 package chance
2 import "math/rand"
3
4 // RollADie returns a random int d with 1 <= d <= 20.
5 func RollADie() int {
6   return (rand.Intn(20) + 1)
7 }
8
9 // GenerateWandEnergy returns a random float64 f with 0.0 <= f < 12.0.
10 func GenerateWandEnergy() float64 {
11   return rand.Float64()*12.0
12 }
13
14 // ShuffleAnimals returns a slice with all eight animal strings in random order.
15 func ShuffleAnimals() []string {
16   var animals []string = []string{"ant", "beaver", "cat", "dog", "elephant", "fox", "giraffe", "pig"}
17   rand.Shuffle(len(animals), func(i, j int){
18     animals[i], animals[j] = animals[j], animals[i]
19   })
20   return animals
21 }
```

Analysis Tests More



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Animal Magic exercise and start writing better Go.

Get mentoring

Bird Watcher

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

bird_watcher.go

```
1 package birdwatcher
2
3 // TotalBirdCount return the total bird count by summing
4 // the individual day's counts.
5 func TotalBirdCount(birdsPerDay []int) int {
6     var totalBirdCount int = 0;
7     for i := 0 ; i < len(birdsPerDay) ; i++ {
8         totalBirdCount += birdsPerDay[i]
9     }
10    return totalBirdCount
11 }
12
13 // BirdsInWeek returns the total bird count by summing
14 // only the items belonging to the given week.
15 func BirdsInWeek(birdsPerDay []int, week int) int {
16     var weeklyBirdCount int = 0;
17     for i := (week*7) - 1 ; i >= (week*7) - 7 ; i-- {
18         weeklyBirdCount += birdsPerDay[i]
19     }
20     return weeklyBirdCount
21 }
22
23 // FixBirdCountLog returns the bird counts after correcting
24 // the bird counts for alternate days.
25 func FixBirdCountLog(birdsPerDay []int) []int {
26     for i := 0 ; i < len(birdsPerDay) ; i++ {
27         if i % 2 == 0 {
28             birdsPerDay[i]++;
29         }
30     }
31     return birdsPerDay
32 }
33
```

Passed

Analysis Tests

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Bird Watcher exercise and start writing better Go.

Get mentoring

Interest is Interesting

Iteration 1 Latest
Submitted via Editor, a few seconds ago

```
2 // InterestRate returns the interest rate for the provided balance.
3 func InterestRate(balance float64) float32 {
4     switch {
5         case balance < 0:
6             return float32(3.213)
7         case balance >=0 && balance < 1000:
8             return float32(0.5)
9         case balance >= 1000 && balance < 5000:
10            return float32(1.621)
11        default:
12            return float32(2.475)
13    }
14 }
15 }
16
17 // Interest calculates the interest for the provided balance.
18 func Interest(balance float64) float64 {
19     var rate float64 = float64(InterestRate(balance))/100.0
20     return balance * rate
21 }
22
23 // AnnualBalanceUpdate calculates the annual balance update, taking into account the interest rate.
24 func AnnualBalanceUpdate(balance float64) float64 {
25     return balance + Interest(balance)
26 }
27
28 // YearsBeforeDesiredBalance calculates the minimum number of years required to reach the desired balance.
29 func YearsBeforeDesiredBalance(balance, targetBalance float64) int {
30     var projectedBalance float64 = balance
31     var years int = 0
32     for ; projectedBalance < targetBalance ; years++ {
33         projectedBalance = AnnualBalanceUpdate(projectedBalance)
34     }
35     return years
36 }
```

Passed | 

 Analysis  Tests 



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Interest is Interesting exercise and start writing better Go.

[Get mentoring](#)

Lasagna Master

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
4 func PreparationTime(layers []string, minutes int) (time int) {
5     if minutes == 0 {
6         minutes = 2
7     }
8     time = len(layers) * minutes
9     return
10 }
11 // TODO: define the 'Quantities()' function
12 func Quantities(layers []string) (int, float64) {
13     var noodleCount int = 0
14     var sauceCount float64 = 0.0
15     for i := 0 ; i < len(layers) ; i++ {
16         switch layers[i] {
17             case "noodles":
18                 noodleCount++
19             case "sauce":
20                 sauceCount++
21             default:
22                 continue
23         }
24     }
25     return noodleCount*50, sauceCount*0.2
26 }
27
28 // TODO: define the 'AddSecretIngredient()' function
29 func AddSecretIngredient(friendsList, myList []string) {
30     myList[len(myList) - 1] = friendsList[len(friendsList) - 1]
31 }
32 // TODO: define the 'ScaleRecipe()' function
33 func ScaleRecipe(quantities []float64, portions int) []float64 {
34     scaledQuantities := make([]float64, len(quantities))
35     copy(scaledQuantities, quantities)
36     for i := 0 ; i < len(scaledQuantities) ; i++ {
37         scaledQuantities[i] = ((scaledQuantities[i] / 2.0) * float64(portions))
38     }
}
```

Passed |  Analysis |  Tests | 



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Lasagna Master exercise and start writing better Go.

[Get mentoring](#)

Booking up for Beauty

Iteration 1 Latest Published
Submitted via Editor, a minute ago

```
5 // Schedule returns a time.Time from a string containing a date.
6 // Layout is used to specify how the string is to be parsed
7 func Schedule(date string) time.Time {
8     layout := "1/02/2006 15:04:05"
9     t, _ := time.Parse(layout, date)
10    return t
11 }
12
13 // HasPassed returns whether a date has passed.
14 func HasPassed(date string) bool {
15     layout := "January 2, 2006 15:04:05"
16     t, _ := time.Parse(layout, date)
17     if t.Compare(time.Now()) == -1 {
18         return true
19     } else {
20         return false
21     }
22 }
23
24 // IsAfternoonAppointment returns whether a time is in the afternoon.
25 func IsAfternoonAppointment(date string) bool {
26     layout := "Monday, January 2, 2006 15:04:05"
27     t, _ := time.Parse(layout, date)
28     if t.Hour() >= 12 && t.Hour() < 18{
29         return true
30     } else {
31         return false
32     }
33 }
34
35 // Description returns a formatted string of the appointment time.
36 func Description(date string) string {
37     layout := "1/2/2006 15:04:05"
38     t, _ := time.Parse(layout, date)
39     // Format is used to display the string in a particular format as per the requirement
```

Passed | ⏺

Analysis Tests ⚙️



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Booking up for Beauty exercise and start writing better Go.

Get mentoring

Gross Store

Iteration 1 Latest Published Submitted via Editor, a minute ago Passed

```
30     if !exists {
31         bill[item] = units[unit]
32     }
33     return true
34 }
35
36 // RemoveItem removes an item from customer bill.
37 func RemoveItem(bill map[string]int, item, unit string) bool {
38     _, itemExists := bill[item]
39     if !itemExists {
40         return false
41     }
42     _, unitExists := units[unit]
43     if !unitExists {
44         return false
45     }
46     bill[item] -= units[unit]
47     if bill[item] < 0 {
48         // Reverting the changes if the operation resulted in negative quantity.
49         bill[item] += units[unit]
50         return false
51     }
52     if bill[item] == 0 {
53         delete(bill, item)
54     }
55     return true
56 }
57
58 // GetItem returns the quantity of an item that the customer has in his/her bill.
59 func GetItem(bill map[string]int, item string) (int, bool) {
60     _, itemExists := bill[item]
61     if !itemExists {
62         return 0, false
63     }
64     return bill[item], true
65 }
```

Analysis Tests

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Gross Store exercise and start writing better Go.

[Get mentoring](#)

Chessboard

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
11     for _, square := range cb[file] {
12         if square {
13             squareOccupied += 1
14         }
15     }
16     return squareOccupied
17 }
18
19 // CountInRank returns how many squares are occupied in the chessboard,
20 // within the given rank.
21 func CountInRank(cb Chessboard, rank int) int {
22     if rank < 1 || rank > 8 {
23         return 0
24     }
25     var squareOccupied int = 0
26     for _, file := range cb {
27         if file[rank-1] {
28             squareOccupied += 1
29         }
30     }
31     return squareOccupied
32 }
33
34 // CountAll should count how many squares are present in the chessboard.
35 func CountAll(cb Chessboard) int {
36     count := 0
37     for _, file := range cb {
38         for range file {
39             count++
40         }
41     }
42     return count
43 }
44
45 // CountOccupied returns how many squares are occupied in the chessboard.
```

Passed | ⏱

Analysis Tests ...



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Chessboard exercise and start writing better Go.

Get mentoring

Election Day

Iteration 1 Latest Published
Submitted via Editor, a minute ago

```
13 func VoteCount(counter *int) int {
14     if counter != nil {
15         return *counter
16     } else {
17         return 0
18     }
19 }
20
21 // IncrementVoteCount increments the value in a vote counter.
22 func IncrementVoteCount(counter *int, increment int) {
23     *counter = *counter + increment
24 }
25
26 // NewElectionResult creates a new election result.
27 func NewElectionResult(candidateName string, votes int) *ElectionResult {
28     var newResult *ElectionResult = &ElectionResult{
29         Name: candidateName,
30         Votes: votes,
31     }
32     return newResult
33 }
34
35 // DisplayResult creates a message with the result to be displayed.
36 func DisplayResult(result *ElectionResult) string {
37     return fmt.Sprintf("%s (%d)", result.Name, result.Votes)
38 }
39
40 // DecrementVotesOfCandidate decrements by one the vote count of a candidate in a map.
41 func DecrementVotesOfCandidate(results map[string]int, candidate string) {
42     _, exists := results[candidate]
43     if exists {
44         results[candidate]--
45     }
46 }
47
```

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Election Day exercise and start writing better Go.

Get mentoring

Elon's Toys

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
elons_toys.go   car.go
```

```
1 package elon
2 import "fmt"
3 // TODO: define the 'Drive()' method
4 func (car *Car) Drive() {
5     if car.battery >= car.batteryDrain {
6         car.distance += car.speed
7         car.battery -= car.batteryDrain
8     }
9 }
10 // TODO: define the 'DisplayDistance() string' method
11 func (car *Car) DisplayDistance() string {
12     return fmt.Sprintf("Driven %d meters", car.distance)
13 }
14 // TODO: define the 'DisplayBattery() string' method
15 func (car *Car) DisplayBattery() string {
16     return fmt.Sprintf("Battery at %d%%", car.battery)
17 }
18 // TODO: define the 'CanFinish(trackDistance int) bool' method
19 func (car *Car) CanFinish(trackDistance int) bool {
20     var distance int = car.distance
21     if car.battery/car.batteryDrain * car.speed + distance >= trackDistance {
22         return true
23     } else {
24         return false
25     }
26 }
```

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Elon's Toys exercise and start writing better Go.

Get mentoring

Logs, Logs, Logs!

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

logs_logs_logs.go

```
1 package logs
2 import "strings"
3 import "unicode/utf8"
4 // Application identifies the application emitting the given log.
5 func Application(log string) string {
6     for _, c := range log {
7         switch c {
8             case '!' :
9                 return "recommendation"
10            case 'Q' :
11                 return "search"
12            case '*' :
13                 return "weather"
14         }
15     }
16     return "default"
17 }
18
19 // Replace replaces all occurrences of old with new, returning the modified log
20 // to the caller.
21 func Replace(log string, oldRune, newRune rune) string {
22     return strings.ReplaceAll(log, string(oldRune), string(newRune))
23 }
24
25 // WithinLimit determines whether or not the number of characters in log is
26 // within the limit.
27 func WithinLimit(log string, limit int) bool {
28     return utf8.RuneCountInString(log) <= limit
29 }
```

Passed |

Analysis Tests



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Logs, Logs, Logs! exercise and start writing better Go.

Get mentoring

Census

Iteration 1 Latest Published
Submitted via Editor, a minute ago

Passed

[Analysis](#) [Tests](#) [...](#)



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Census exercise and start writing better Go.

[Get mentoring](#)

```
11 // NewResident registers a new resident in this city.
12 func NewResident(name string, age int, address map[string]string) *Resident {
13     var newResidentPtr *Resident
14     newResidentPtr = &Resident{
15         Name: name,
16         Age: age,
17         Address: address,
18     }
19     return newResidentPtr
20 }
21
22 // HasRequiredInfo determines if a given resident has all of the required information.
23 func (r *Resident) HasRequiredInfo() bool {
24     if r.Name == "" || len(r.Address) == 0 {
25         return false
26     } else {
27         for k, v := range r.Address {
28             if v == "" || k != "street" {
29                 return false
30             }
31         }
32         return true
33     }
34 }
35
36 // Delete deletes a resident's information.
37 func (r *Resident) Delete() {
38     r.Name = ""
39     r.Age = 0
40     r.Address = nil
41 }
42
43 // Count counts all residents that have provided the required information.
44 func Count(residents []*Resident) int {
45     var count int = 0
46 }
```

Parsing Log Files

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
11 }
12
13 func CountQuotedPasswords(lines []string) int {
14     re := regexp.MustCompile(`(?i)".*password.*"`)
15     // (?i) is case insensitive flag
16     // .* Matches any character (.) zero or more times (*)
17     count := 0
18     for _, line := range lines {
19         if re.MatchString(line) {
20             count++
21         }
22     }
23     return count
24 }
25
26 func RemoveEndOfLineText(text string) string {
27     return regexp.MustCompile(`\n`).ReplaceAllString(text, "")
28     // \d+ Matches one or more digits (0-9).
29 }
30
31 func TagWithUserName(lines []string) []string {
32     var tagged []string
33     re := regexp.MustCompile(`User\s+([A-Za-z0-9]+)` // Highlight all username strings
34     // \s+ Matches one or more whitespace characters (spaces, tabs, newlines)
35     // () defines a capturing group
36     for _, line := range lines {
37         taggedLogs := re.FindStringSubmatch(line)
38         // First occurrence of User...
39         // The first element of the slice is the entire matched substring
40         // Basically ["The quick", "The", "quick"]
41         if taggedLogs != nil {
42             tagged = append(tagged, "[USR] "+taggedLogs[1]+" "+line)
43         } else {
44             tagged = append(tagged, line)
45         }
46 }
```

Passed |

Analysis Tests ...

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Parsing Log Files exercise and start writing better Go.

[Get mentoring](#)

Meteorology

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
26 }
27
28 // Add a String method to the Temperature type
29 func (t Temperature) String() string {
30     return fmt.Sprintf("%d %s", t.degree, t.unit.String())
31 }
32
33 type SpeedUnit int
34
35 const (
36     KmPerHour     SpeedUnit = 0
37     MilesPerHour SpeedUnit = 1
38 )
39
40 // Add a String method to SpeedUnit
41 func (s SpeedUnit) String() string {
42     switch s {
43         case KmPerHour:
44             return "km/h"
45         case MilesPerHour:
46             return "mph"
47     }
48     return ""
49 }
50
51 type Speed struct {
52     magnitude int
53     unit       SpeedUnit
54 }
55
56 // Add a String method to Speed
57 func (s Speed) String() string {
58     return fmt.Sprintf("%d %s", s.magnitude, s.unit.String())
59 }
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
948
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
978
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2239
2240
2241
22
```

Sorting Room

Iteration 1 Latest Published
Submitted via Editor, a few seconds ago

```
35     fancyNumber, ok := fnb.(FancyNumber)
36     if !ok {
37         return 0
38     }
39     num, _ := strconv.Atoi(fancyNumber.n)
40     return num
41 }
42
43 // DescribeFancyNumberBox should return a string describing the FancyNumberBox.
44 func DescribeFancyNumberBox(fnb FancyNumberBox) string {
45     fancyNumber, ok := fnb.(FancyNumber)
46     if !ok {
47         return "This is a fancy box containing the number 0.0"
48     }
49     num, _ := strconv.Atoi(fancyNumber.n)
50     return fmt.Sprintf("This is a fancy box containing the number %.1f", float64(num))
51 }
52
53
54 // DescribeAnything should return a string describing whatever it contains.
55 func DescribeAnything(i interface{}) string {
56     switch v := i.(type) {
57     case int:
58         return DescribeNumber(float64(v))
59     case float64:
60         return DescribeNumber(v)
61     case NumberBox:
62         return DescribeNumberBox(v)
63     case FancyNumberBox:
64         return DescribeFancyNumberBox(v)
65     default:
66         return "Return to sender"
67     }
68 }
69 }
```

● Passed | ⏪

Analysis Tests ⚙️



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Sorting Room exercise and start writing better Go.

Get mentoring

The Farm

Iteration 1 Latest Submitted via Editor, a few seconds ago

```
5 func DivideFood(fc FodderCalculator, cows int) (float64, error) {
6     totalFodder, okFodder := fc.FodderAmount(cows)
7     factor, okFactor := fc.FatteningFactor()
8     if okFodder != nil || okFactor != nil {
9         if okFodder == nil {
10             return 0, okFactor
11         } else {
12             return 0, okFodder
13         }
14     }
15     totalFodder *= factor
16     return float64(totalFodder)/float64(cows), nil
17 }
18 // TODO: define the 'ValidateInputAndDivideFood' function
19 func ValidateInputAndDivideFood(fc FodderCalculator, cows int) (float64, error) {
20     if cows > 0 {
21         return DivideFood(fc, cows)
22     } else {
23         return 0, errors.New("invalid number of cows")
24     }
25 }
26 // TODO: define the 'ValidateNumberOfCows' function
27 func ValidateNumberOfCows(cows int) error {
28     if cows < 0 {
29         return errors.New(fmt.Sprintf("%d cows are invalid: there are no negative cows", cows))
30     } else if cows == 0 {
31         return errors.New(fmt.Sprintf("%d cows are invalid: no cows don't need food", cows))
32     } else {
33         return nil
34     }
35 }
36 // Your first steps could be to read through the tasks, and create
37 // these functions with their correct parameter lists and return types.
38 // The function body only needs to contain 'panic("")'.
39 //
```

Passed |

Analysis Tests



No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on The Farm exercise and start writing better Go.

Get mentoring

Expenses

Iteration 1 Latest Published
Submitted via Editor, a minute ago

```
37 // and false otherwise.
38 func ByCategory(c string) func(Record) bool {
39     return func(r Record) bool {
40         return c == r.Category
41     }
42 }
43
44 // TotalByPeriod returns total amount of expenses for records
45 // inside the period p.
46 func TotalByPeriod(in []Record, p DaysPeriod) float64 {
47     total := 0.0
48     Callback := ByDaysPeriod(p)
49     for _, record := range in {
50         if Callback(record) {
51             total += record.Amount
52         }
53     }
54     return total
55 }
56
57 // CategoryExpenses returns total amount of expenses for records
58 // in category c that are also inside the period p.
59 // An error must be returned only if there are no records in the list that belong
60 // to the given category, regardless of period of time.
61 func CategoryExpenses(in []Record, p DaysPeriod, c string) (float64, error) {
62     CallbackCategory := ByCategory(c)
63     filteredByCategory := Filter(in, CallbackCategory)
64     // Match all categories first
65     if len(filteredByCategory) == 0 {
66         return 0, errors.New("unknown category entertainment")
67     }
68     // Get total by period which can be 0 if none of the record are between the given period.
69     return TotalByPeriod(filteredByCategory, p), nil
70 }
71
```

Passed |

Analysis Tests

No auto suggestions? Try human mentoring.

Get real 1-to-1 human mentoring on the Expenses exercise and start writing better Go.

Get mentoring

Completed Learning Path

You're 19.9% through the Go track. That's a great start! 🚀

Completed 28 In-progress 0 Available 113 Locked 0 Total Exercises 141

28
Exercises completed >

34
Concepts learnt >

Co Nu St Po Va

34
Concepts mastered >

Ra Ty Pa Ru Fl

Go Trophy Cabinet
Unlock for achievements as you progress through the track. [Learn more](#)

 PROLIFIC PADAWAN	 LOCKED	 LOCKED	 LOCKED	 LOCKED	 LOCKED
 FUNDAMENTAL					