

GoLang Problem Solving (Additional)

By Suteerth Subramaniam

- [Main Repo](#)
- [Golang Exercism Problem Solving](#)

Anagrams

Iteration 1 Latest Published
Submitted via Editor, a minute ago

anagram.go cases_test.go

```
1 package anagram
2 import (
3     "unicode"
4     "strings"
5 )
6 func Detect(subject string, candidates []string) []string {
7     freqMap := make(map[rune]int)
8     var res []string
9     for _, c := range subject {
10         freqMap[unicode.ToLower(c)]++
11     }
12     for _, s := range candidates {
13         if s == subject || strings.ToLower(subject) == strings.ToLower(s) {
14             continue
15         }
16         testerMap := make(map[rune]int)
17         for _, c := range s {
18             testerMap[unicode.ToLower(c)]++
19         }
20         flag := true
21         for k,v := range testerMap {
22             if freqMap[k] != v {
23                 flag = false
24                 break
25             }
26         }
27         if flag {
28             res = append(res, s)
29         }
30     }
31     return res
32 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Two Fer

Iteration 1 Latest
Submitted via Editor, a few seconds ago

two_fer.go

```
1 package twofer
2
3 func ShareWith(name string) string {
4     if name == "" {
5         return "One for you, one for me."
6     } else {
7         return "One for " + name + ", one for me."
8     }
9 }
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

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

Get mentoring

Raindrops

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

raindrops.go cases_test.go

```
1 package raindrops
2 import "fmt"
3 func Convert(number int) string {
4     var res string = ""
5     if number % 3 == 0 {
6         res += "Pling"
7     }
8     if number % 5 == 0 {
9         res += "Plang"
10    }
11    if number % 7 == 0 {
12        res += "Plong"
13    }
14    if res == "" {
15        return fmt.Sprintf("%d",number)
16    } else {
17        return res
18    }
19 }
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

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

Get mentoring

Collatz Conjecture

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

collatz_conjecture.go cases_test.go

```
1 package collatzconjecture
2 import "errors"
3 func CollatzConjecture(n int) (int, error) {
4     steps := 0
5     if n <= 0 {
6         return 0, errors.New("Only positive integers are allowed.")
7     }
8     for ; n != 1 ; steps++ {
9         if n % 2 == 0 {
10             n /= 2
11         } else {
12             n *= 3
13             n += 1
14         }
15     }
16     return steps, nil
17 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Gigaseconds

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

gigasecond.go cases_test.go

```
1 package gigasecond
2
3 import "time"
4
5 // What's the time after 1 gigasecond.
6 func AddGigasecond(t time.Time) time.Time {
7     return t.Add(time.Second * 1e9)
8 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Flatten Array

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

flatten_array.go	cases_test.go
------------------	---------------

```
1 package flatten
2
3 func Flatten(nested interface{}) []interface{} {
4     arr := []interface{}{}
5     // Take advantage of type assertions
6     // If the element is of slice type, open it up and add the elements
7     switch i := nested.(type) {
8         // It is a 1D thing
9         case []interface{}:
10             for _, val := range i {
11                 arr = append(arr, Flatten(val)...)
12             }
13         case interface{}:
14             arr = append(arr, i)
15     }
16     return arr
17 }
```

Analysis Tests [Get mentoring](#)



No auto suggestions? Try human mentoring.

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

Linked List

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

linked_list.go	cases_test.go
----------------	---------------

```
47     if dll.isEmpty() {
48         return int(0), errors.New("err empty list")
49     }
50     headVal := dll.head.Value
51     dll.head = dll.head.next
52     // After shifting, the head is NIL
53     if dll.head == nil {
54         dll.tail = nil
55     } else {
56         dll.head.prev = nil
57     }
58     return headVal, nil
59 }
60 func (dll *List) Unshift(v interface{}) {
61     node := &Node{Value: v}
62     if dll.isEmpty() {
63         dll.head = node
64         dll.tail = node
65         return
66     }
67     node.next = dll.head
68     dll.head.prev = node
69     dll.head = node
70 }
71 func (dll *List) Push(v interface{}) {
72     node := &Node{Value: v}
73     if dll.isEmpty() {
74         dll.head = node
75         dll.tail = node
76         return
77     }
78     node.prev = dll.tail
79     dll.tail.next = node
80     dll.tail = node
```

Analysis Tests [Get mentoring](#)



No auto suggestions? Try human mentoring.

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

Binary Search Tree

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

```
18 // Insert inserts an int into the BinarySearchTree.
19 // Inserts happen based on the rules of a binary search tree
20 func (bst *BinarySearchTree) Insert(i int) {
21     if i <= bst.data {
22         if bst.left == nil {
23             bst.left = NewBst(i)
24         } else {
25             bst.left.Insert(i)
26         }
27     } else {
28         if bst.right == nil {
29             bst.right = NewBst(i)
30         } else {
31             bst.right.Insert(i)
32         }
33     }
34 }
35 }
36
37 // SortedData returns the ordered contents of BinarySearchTree as an []int.
38 // The values are in increasing order starting with the lowest int value.
39 // A BinarySearchTree that has the numbers [1,3,7,5] added will return the
40 // []int [1,3,5,7].
41 func (bst *BinarySearchTree) SortedData() []int {
42     var inorder []int
43     if bst.left != nil {
44         inorder = append(inorder, bst.left.SortedData()...)
45     }
46     inorder = append(inorder, bst.data)
47     if bst.right != nil {
48         inorder = append(inorder, bst.right.SortedData()...)
49     }
50     return inorder
51 }
52 }
```

Passed | ⏺

Analysis Tests ⚙️



No auto suggestions? Try human mentoring.

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

Get mentoring

Say

Iteration 2 Latest Published Submitted via Editor, a minute ago Passed

```
48     return sayPart(n, ""), true
49 }
50
51 func sayPart(n int64, prefix string) string {
52     // Going from top to bottom
53     for _, p := range parts {
54         // value is 1000 and name is "thousand"
55         // n = 1056 but p.value = 1000000 will make order as 0 and rem as n
56         order, rem := n/p.value, n%p.value
57         if order == 0 {
58             // The order of magnitude is not matching
59             continue
60         }
61         if p.value < 100 {
62             // Role of prefix is to add the space or the hyphen
63             // If we are at 90, then p.name is ninety and the prefix is going to be hyphen for
64             return prefix + p.name + sayPart(rem, "-")
65         }
66         // Add the order like thousand which is 1-9 giving us results like nine thousand and th
67         // we have the sayPart(rem, " ") which basically makes us add a space in the front
68         return sayPart(order, prefix) + " " + p.name + sayPart(rem, " ")
69     }
70     /*
71     Dry RUN for 19876
72     19876 / 1000 gives order = 19 and rem = 876
73     sayPart(19, "") + " " + thousand + sayPart(876, " ")
74     "19 thousand " + sayPart(876, " ")
75     876 / 100 gives order = 8 and rem = 76
76     sayPart(8, " ") + hundred + sayPart(76, " ")
77     "19 thousand 8 hundred " + sayPart(76, " ")
78 */
79     return ""
80 }
81
82
```

Analysis Tests 

No auto suggestions? Try human mentoring.

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

Get mentoring

Pascal Triangle

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

```
pascals_triangle.go
```

```
1 package pascal
2
3 func Triangle(n int) [][]int {
4     var pascal [][]int
5     row1 := []int{1}
6     row2 := []int{1,1}
7     if n == 1 {
8         return append(pascal, row1)
9     }
10    if n == 2 {
11        return append(pascal, row1, row2)
12    }
13    pascal = append(pascal, row1, row2)
14    for i := 3 ; i <= n ; i++ {
15        curr := make([]int, i)
16        for j := 0 ; j < len(pascal[i-2]) - 1 ; j++ {
17            curr[j+1] = pascal[i-2][j] + pascal[i-2][j+1]
18        }
19        curr[0] = 1
20        curr[len(curr)-1] = 1
21        pascal = append(pascal, curr)
22    }
23    return pascal
24 }
```

Analysis Tests 

No auto suggestions? Try human mentoring.

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

Get mentoring

Binary Search

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

binary_search.go cases_test.go

```
1 package binarysearch
2 import "sort"
3 func SearchInts(list []int, key int) int {
4     sort.Ints(list)
5     left := 0
6     right := len(list) - 1
7     for ; left <= right; {
8         mid := (left + right) / 2
9         if list[mid] == key {
10             return mid
11         }
12         if list[mid] < key {
13             left = mid + 1
14         } else {
15             right = mid - 1
16         }
17     }
18     return -1
19 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Leap

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

leap.go cases_test.go

```
1 package leap
2
3 func IsLeapYear(year int) bool {
4     if year % 4 != 0 {
5         return false
6     }
7     if year % 100 == 0 && year % 400 != 0 {
8         return false
9     }
10    return true
11 }
12
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Hamming

Iteration 2 Latest Published
Submitted via Editor, a minute ago

hamming.go cases_test.go

```
1 package hamming
2
3 import "errors"
4
5 func Distance(a, b string) (int, error) {
6     if len(a) != len(b) {
7         return 0, errors.New("Length of DNA strands is not the same.")
8     }
9     hammingDistance := 0
10    for i := range b {
11        if a[i] != b[i] {
12            hammingDistance++
13        }
14    }
15    return hammingDistance, nil
16 }
```

Analysis Tests [...](#)



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Passed

Grep

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

for _, file := range files {
 fileContent, _ := os.ReadFile(file)
 // Going through every line which is splitted with the newline character.
 for index, line := range strings.Split(string(fileContent), "\n") {
 // Ignore Empty lines
 if line == "" {
 continue
 }
 // Before applying any flags
 processedLine, processedPattern := line, pattern
 if caseInsensitive {
 // Convert all to lower case
 processedLine, processedPattern = strings.ToLower(line), strings.ToLower(pattern)
 }
 var match bool
 if exactMatch {
 match = (processedLine == processedPattern)
 } else {
 // Contain method to search for the pattern in the line.
 match = strings.Contains(processedLine, processedPattern)
 }
 // inverseMatch and match are opposites
 if (match && !inverseMatch) || (!match && inverseMatch) {
 // If we need lines with at least one match
 var whetherPrependLineOrNot string
 if atleastOneLineMatch {
 // We only need to output name of files in case of atleastOneLineMatch flag
 result = append(result, file)
 break
 } else if prependLineNumber {
 // Add a line number as per the prepend flag
 whetherPrependLineOrNot = fmt.Sprintf("%d:%s", index+1, line)
 } else {
 whetherPrependLineOrNot = line
 }
 }
 }
}

Analysis Tests [...](#)



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Passed

Scrabble

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

scrabble_score.go cases_test.go

```
1 package scrabble
2
3 import "strings"
4
5 func Score(word string) int {
6     scoreMap := map[rune]int{
7         'A': 1, 'E': 1, 'I': 1, 'O': 1, 'U': 1, 'L': 1, 'N': 1, 'R': 1, 'S': 1, 'T': 1,
8         'D': 2, 'G': 2,
9         'B': 3, 'C': 3, 'M': 3, 'P': 3,
10        'F': 4, 'H': 4, 'V': 4, 'W': 4, 'Y': 4,
11        'K': 5,
12        'J': 8, 'X': 8,
13        'Q': 10, 'Z': 10,
14    }
15
16    res := 0
17    for _, c := range strings.ToUpper(word) {
18        score, ok := scoreMap[c]
19        if ok {
20            res += score
21        }
22    }
23    return res
24 }
```

Analysis Tests [...](#)



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Passed

Complex Numbers

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

ComplexNumbers.go

```
43     return Number{
44         r: n.r * factor,
45         i: n.i * factor,
46     }
47 }
48
49 func (n1 Number) Divide(n2 Number) Number {
50     denominator := n2.r * n2.r + n2.i * n2.i
51     if denominator == 0 {
52         panic("Division by zero")
53     }
54     realPart := (n1.r*n2.r + n1.i*n2.i) / denominator
55     imaginaryPart := (n1.i*n2.r - n1.r*n2.i) / denominator
56     return Number{r: realPart, i: imaginaryPart,}
57 }
58
59
60 func (n Number) Conjugate() Number {
61     return Number{r: n.r, i: (-1)*n.i}
62 }
63
64 func (n Number) Abs() float64 {
65     return math.Sqrt(n.r*n.r + n.i*n.i)
66 }
67
68 func (n Number) Exp() Number {
69     realPart := math.Exp(n.r)
70     cosB := math.Cos(n.i)
71     sinB := math.Sin(n.i)
72     return Number{
73         r: realPart * cosB,
74         i: realPart * sinB,
75     }
76 }
```

Analysis Tests [...](#)



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Passed

Wordy

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

```
4     "regexp"
5     "strconv"
6 )
7
8 func Answer(s string) (int, bool) {
9     match, _ := regexp.MatchString(`What is -?\d+(?: (?:plus|minus|divided by|multiplied by) -?`)
10    if !match {
11        return 0, false
12    }
13
14    rel := regexp.MustCompile(`(plus|minus|divided|multiplied)`)
15    operators := rel.FindAllString(s, -1) // Find all the occurrences of the operations
16
17    re2 := regexp.MustCompile(`-\?\d+`)
18    numbers := re2.FindAllString(s, -1) // Find all the occurrences of the numbers
19
20    if len(operators) != len(numbers)-1 { // For 'n' operators there are 'n - 1' operations
21        return 0, false
22    }
23    sum, _ := strconv.Atoi(numbers[0])
24    for i, o := range operators {
25        n, _ := strconv.Atoi(numbers[i + 1])
26        switch o {
27            case "plus":
28                sum += n
29            case "minus":
30                sum -= n
31            case "divided":
32                sum /= n
33            case "multiplied":
34                sum *= n
35        }
36    }
37    return sum, true
38 }
```

● Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Spiral Matrix

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

```
spiral_matrix.go
1 package spiralmatrix
2
3 func SpiralMatrix(size int) [][]int {
4     xDir := []int{0, 1, 0, -1}
5     yDir := []int{1, 0, -1, 0}
6
7     r := 0
8     c := 0
9     step := 0
10
11    res := make([][]int, size)
12    for i := range res {
13        res[i] = make([]int, size)
14    }
15
16    for i := 0 ; i < size * size ; i++ {
17        res[r][c] = i + 1
18        newR, newC := r + xDir[step], c + yDir[step]
19        if newR < 0 || newR >= size || newC < 0 || newC >= size || res[newR][newC] != 0 {
20            step = (step + 1) % 4 // Change direction
21        }
22        r += xDir[step]
23        c += yDir[step]
24    }
25    return res
26 }
27
```

● Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Isogram

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

isogram.go cases_test.go

```
1 package isogram
2
3 import "strings"
4
5 func IsIsogram(word string) bool {
6     word = strings.ToLower(word)
7     freq := make(map[rune]int)
8     for _, c := range word {
9         if c == ' ' || c == '-' {
10             continue
11         }
12         val, exists := freq[c]
13         if !exists {
14             freq[c] = 1
15         } else {
16             if val == 1 {
17                 return false
18             }
19         }
20     }
21     return true
22 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Difference of Squares

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

difference_of_squares.go

```
1 package diffsquare
2
3 func SquareOfSum(n int) int {
4     sum := (n*(n + 1)) / 2
5     return sum * sum
6 }
7
8 func SumOfSquares(n int) int {
9     return (n * (n+1) * (2*n+1))/6
10 }
11
12 func Difference(n int) int {
13     return SquareOfSum(n) - SumOfSquares(n)
14 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Luhn

Iteration 1 [Latest](#) [Published](#)

Submitted via Editor, a few seconds ago

[Run](#) [Edit](#) [View raw](#) [Copy](#) [Share](#)

luhn.go cases_test.go

```
1 package luhn
2
3 import "strings"
4
5 func Valid(id string) bool {
6     id = strings.ReplaceAll(id, " ", "")
7     n := len(id)
8     if n < 2 {
9         return false
10    }
11    sum := 0
12    for i := n - 1; i >= 0; i-- {
13        num := id[i]
14        if num < '0' || num > '9' {
15            return false
16        }
17        digit := int(num - '0')
18        if (n - i) % 2 == 0 {
19            digit *= 2
20            if digit > 9 {
21                digit -= 9
22            }
23        }
24        sum += digit
25    }
26    return sum % 10 == 0
27 }
```

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



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Resistor Color

Iteration 1 Latest Published

Submitted via Editor, a minute ago

Passed

resistor_color.go cases_test.go

```
1 package resistorcolor
2
3 // Colors returns the list of all colors.
4 func Colors() []string {
5     return []string{"black", "brown", "red", "orange", "yellow", "green", "blue", "violet", "gr
6 }
7
8 // ColorCode returns the resistance value of the given color.
9 func ColorCode(color string) int {
10    colors := Colors()
11    for i, v := range colors {
12        if color == v {
13            return i
14        }
15    }
16    return -1
17 }
```

Analysis Tests

ooo



No auto suggestions? Try human mentoring.

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

Get mentoring

Space Age

Iteration 1 Latest Published
Submitted via Editor, 4 minutes ago

space_age.go cases_test.go

```
1 package space
2
3 type Planet string
4
5 var orbitalPeriod = map[Planet]float64{
6     "Mercury": 0.2408467, "Venus": 0.61519726, "Earth": 1.0, "Mars": 1.8808158,
7     "Jupiter": 11.862615, "Saturn": 29.447498, "Uranus": 84.016846, "Neptune": 164.79132,
8 }
9
10 func Age(seconds float64, planet Planet) float64 {
11     if orbit, exists := orbitalPeriod[planet]; exists {
12         years := seconds / 31557600.0
13         return years / orbit
14     } else {
15         return -1.0
16     }
17 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Series

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

series.go

```
1 package series
2
3 func All(n int, s string) []string {
4     if len(s) < n {
5         return nil
6     }
7     var result []string
8     runes := []rune(s)
9     for i := 0; i <= len(s)-n; i++ {
10         result = append(result, string(runes[i:n+i]))
11     }
12     return result
13 }
14
15 func UnsafeFirst(n int, s string) string {
16     return All(n, s)[0]
17 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Prime Factors

Iteration 1 Latest Submitted via Editor, a few seconds ago

prime_factors.go cases_test.go

```
1 package prime
2
3 func Factors(n int64) []int64 {
4     res := make([]int64, 0)
5     // Theory used: A number is a product of its prime factors only
6     // 12 = 2 x 2 x 3
7     for i := int64(2); n > 1; { // Start off with 2, until we reach n which will directly divide
8         if n % i == 0{
9             res = append(res, i)
10            n /= i
11        } else {
12            i++
13        }
14    }
15    return res
16 }
```

Passed

Analysis Tests

No auto suggestions? Try human mentoring.

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

Get mentoring



Armstrong Numbers

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

armstrong_numbers.go cases_test.go

```
1 package armstrong
2
3 import (
4     "strconv"
5     "math"
6 )
7
8 func IsNumber(n int) bool {
9     if n < 9 {
10         return true
11     }
12     numString:= strconv.Itoa(n)
13     order := len(numString)
14     sum := 0
15     for _, digit := range numString {
16         sum += int(math.Pow(float64(digit - '0'), float64(order)))
17     }
18     return sum == n
19 }
```

Passed

Analysis Tests

No auto suggestions? Try human mentoring.

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

Get mentoring



D&D Character

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

```
20 func Modifier(score int) int {
21     return int(math.Floor(float64(score - 10) / 2.0))
22 }
23
24 // Ability uses randomness to generate the score for an ability
25 func Ability() int {
26     diceRolls := make([]int, 4)
27     for i := 0; i < 4; i++ {
28         diceRolls[i] = rand.Intn(6) + 1
29     }
30     sort.Ints(diceRolls)
31     return diceRolls[1] + diceRolls[2] + diceRolls[3]
32 }
33
34 // GenerateCharacter creates a new Character with random scores for abilities
35 func GenerateCharacter() Character {
36     strength := Ability()
37     dexterity := Ability()
38     constitution := Ability()
39     intelligence := Ability()
40     wisdom := Ability()
41     charisma := Ability()
42
43     hitpoints := 10 + Modifier(constitution)
44
45     return Character{
46         Strength: strength,
47         Dexterity: dexterity,
48         Constitution: constitution,
49         Intelligence: intelligence,
50         Wisdom: wisdom,
51         Charisma: charisma,
52         Hitpoints: hitpoints,
53     }
54 }
55 }
```

Analysis Tests



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

ETL

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

```
etl.go | etl.go
1 package etl
2
3 import "strings"
4
5 func Transform(input map[int][]string) map[string]int {
6     var etl = make(map[string]int)
7     for pts, letters := range input {
8         for _, letter := range letters {
9             etl[strings.ToLower(letter)] = pts
10        }
11    }
12    return etl
13 }
```

Analysis Tests



No auto suggestions? Try human mentoring.

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

[Get mentoring](#)

Nucleotide Count

Iteration 1 Latest
Submitted via Editor, a few seconds ago

Passed

nucleotide_count.go cases_test.go

```
1 package dna
2
3 import (
4     "strings"
5     "fmt"
6 )
7
8 // Histogram is a mapping from nucleotide to its count in given DNA.
9 // Choose a suitable data type.
10 type Histogram map[rune]int
11
12 // DNA is a list of nucleotides. Choose a suitable data type.
13 type DNA string
14 // Counts generates a histogram of valid nucleotides in the given DNA.
15 // Returns an error if d contains an invalid nucleotide.
16
17 // Counts is a method on the DNA type. A method is a function with a special receiver argument.
18 // The receiver appears in its own argument list between the func keyword and the method name.
19 // Here, the Counts method has a receiver of type DNA named d.
20 func (d DNA) Counts() (Histogram, error) {
21     var h Histogram = Histogram{'A': 0, 'C': 0, 'G': 0, 'T': 0}
22     d = DNA(strings.ToUpper(string(d)))
23     for _, v := range d {
24         _, ok := h[v]
25         if !ok {
26             return nil, fmt.Errorf("Invalid nucleotide %v", v)
27         }
28         h[v]++
29     }
30     return h, nil
31 }
32
```

Analysis Tests

No auto suggestions? Try human mentoring.

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

Get mentoring

DNA Transcription

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

Passed

rna_transcription.go cases_test.go

```
1 package strand
2
3 func ToRNA(dna string) string {
4     transcribe := map[rune]rune{'G':'C', 'C':'G', 'T':'A', 'A':'U'}
5     res := ""
6     for _, v := range dna {
7         res += string(transcribe[v])
8     }
9     return res
10 }
```

Analysis Tests

No auto suggestions? Try human mentoring.

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

Get mentoring

Strain

Iteration 1 Latest
Submitted via Editor, a minute ago

strain.go

```
1 package strain
2
3 func Keep[AnyCollection any](arr []AnyCollection, predicate func(AnyCollection) bool) []AnyCollection {
4     var res []AnyCollection
5     for _, v := range arr {
6         if predicate(v) {
7             res = append(res, v)
8         }
9     }
10    return res
11 }
12
13 func Discard[AnyCollection any](arr []AnyCollection, predicate func(AnyCollection) bool) []AnyCollection {
14     var res []AnyCollection
15     for _, v := range arr {
16         if !predicate(v) {
17             res = append(res, v)
18         }
19     }
20    return res
21 }
22 // You will need typed parameters (aka "Generics") to solve this exercise.
23 // They are not part of the Exercism syllabus yet but you can learn about
24 // them here: https://go.dev/tour/generics/1
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

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

Get mentoring

Pangram

Iteration 1 Latest
Submitted via Editor, a few seconds ago

pangram.go cases_test.go

```
1 package pangram
2
3 import "strings"
4
5 func IsPangram(input string) bool {
6     freq := make([]int, 26)
7     input = strings.ToLower(input)
8     for _, c := range input {
9         if c < 'a' || c > 'z' {
10             continue
11         }
12         freq[int(c - 'a')]++
13     }
14     for _, v := range freq {
15         if v == 0 {
16             return false
17         }
18     }
19     return true
20 }
```

Passed

Analysis Tests



No auto suggestions? Try human mentoring.

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

Get mentoring

Reverse String

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

reverse_string.go cases_test.go

```
1 package reverse
2
3 func Reverse(input string) string {
4     runes := []rune(input)
5     for i, j := 0, len(runes)-1; i < j; i, j = i+1, j-1 {
6         runes[i], runes[j] = runes[j], runes[i]
7     }
8     return string(runes)
9 }
10
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Darts

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

darts.go cases_test.go

```
1 package darts
2
3 func Score(x, y float64) int {
4     // For a circle :  $x^2 + y^2 = r^2$ 
5     x2 := x*x
6     y2 := y*y
7     if x2 + y2 <= 1.0 {
8         return 10
9     }
10    if x2 + y2 > 1.0 && x2 + y2 <= 25.0 {
11        return 5
12    }
13    if x2 + y2 > 25.0 && x2 + y2 <= 100.0 {
14        return 1
15    }
16    return 0
17 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Bob

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

```
6 )
7
8 func isSilence(remark string) bool {
9     return remark == ""
10 }
11
12 func isShouting(remark string) bool {
13     hasLetters := strings.IndexFunc(remark, unicode.IsLetter) >= 0 // First letter index or -1
14     isUpcased := strings.ToUpper(remark) == remark
15     return hasLetters && isUpcased
16 }
17
18 func isQuestion(remark string) bool {
19     return strings.HasSuffix(remark, "?")
20 }
21
22 func hasYelledAQuestion(remark string) bool {
23     return isShouting(remark) && isQuestion(remark)
24 }
25
26 func Hey(remark string) string {
27     remark = strings.Trim(remark, " \t\n\r")
28     switch {
29     case isSilence(remark):
30         return "Fine. Be that way!"
31     case hasYelledAQuestion(remark):
32         return "Calm down, I know what I'm doing!"
33     case isShouting(remark):
34         return "Whoa, chill out!"
35     case isQuestion(remark):
36         return "Sure."
37     default:
38         return "Whatever."
39     }
40 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Grains

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

```
grains.go   cases_test.go
```

```
1 package grains
2
3 import (
4     "errors"
5 )
6
7 func Square(number int) (uint64, error) {
8     if number <= 0 || number > 64 {
9         return 0, errors.New("invalid square number")
10    }
11
12    return 1 << (number - 1), nil // Bitwise left shift for power of 2
13 }
14
15 func Total() uint64 {
16     var sum uint64 = 0
17     for i := 1; i <= 64; i++ {
18         sq, err := Square(i)
19         if err != nil {
20             continue
21         }
22         sum += sq
23     }
24     return sum
25 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Sum of multiples

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

sum_of_multiples.go cases_test.go

```
1 package summultiples
2
3 func SumMultiples(limit int, divisors ...int) int { // find sum of multiples of given divisors
4     // limit is the excluded upper limit
5     // starting from
6     sum := 0
7     for i:= 1 ; i < limit ; i++ {
8         // add the number if its a multiple of at least one of the divisors
9         for _, d := range divisors {
10             if d != 0 && i%d == 0{
11                 sum += i
12                 break // to add each multiple only once
13             }
14         }
15     }
16     return sum
17 }
18 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

ISBN Verifier

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

isbn_verifier.go cases_test.go

```
1 package isbn
2
3 import "strings"
4
5 func IsValidISBN(isbn string) bool {
6     isbn = strings.ReplaceAll(isbn, "-", "")
7     if len(isbn) != 10 {
8         return false
9     }
10    sum := 0
11    for i, c := range isbn {
12        var digit int
13        switch c {
14            case '0', '1', '2', '3', '4', '5', '6', '7', '8', '9':
15                digit = int(c - '0')
16            case 'X':
17                if i != 9 { // Check if X is in the last position
18                    return false
19                }
20                digit = 10
21            default:
22                return false // Invalid character
23            }
24        sum += digit * (10 - i)
25    }
26    return sum % 11 == 0
27 }
```

Passed

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Triangle

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

```
1 package triangle
2
3 import "math"
4
5 type Kind int
6
7 const (
8     NaT = 0 // not a triangle
9     Equ = 1 // equilateral
10    Iso = 2 // isosceles
11    Sca = 3 // scalene
12 )
13
14 func KindFromSides(a, b, c float64) Kind {
15     var k Kind
16     switch {
17         case math.IsNaN(a) || math.IsNaN(b) || math.IsNaN(c):
18             k = NaT
19             // Cover -Inf and Inf both
20             case math.IsInf(a, 1) || math.IsInf(b, 1) || math.IsInf(c, 1) || math.IsInf(a, -1) || m
21             k = NaT
22             case a <= 0 || b <= 0 || c <= 0:
23                 k = NaT
24             case a + b < c || a + c < b || b + c < a:
25                 k = NaT
26             case a == b && a == c:
27                 k = Equ
28             case (a == b && a != c) || (a == c && a != b) || (b == c && a != b):
29                 k = Iso
30             case a != b && a != c:
31                 k = Sca
32     }
33     return k
34 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Word Count

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

```
word_count.go    cases_test.go
```

```
1 package wordcount
2
3 import (
4     "regexp"
5     "strings"
6 )
7
8 type Frequency map[string]int
9
10 func WordCount(phrase string) Frequency {
11     freq := Frequency{}
12     re := regexp.MustCompile(`\w+(''\w+')?`) // Either a word or a word followed by apostrophe and
13     words := re.FindAllString(strings.ToLower(phrase), -1)
14     for _, word := range words {
15         freq[word]++
16     }
17     return freq
18 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Sieve

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

Passed

sieve.go cases_test.go

```
1 package sieve
2
3 func Sieve(limit int) []int {
4     primes := make([]bool, limit+1)
5     for i := 2 ; i <= limit ; i++ {
6         primes[i] = true
7     }
8     for p := 2 ; p*p <= limit ; p++ {
9         if primes[p] == true {
10             for i := p * p ; i <= limit; i += p {
11                 primes[i] = false
12             }
13         }
14     }
15     var primeNumbers []int
16     for p := 2 ; p <= limit ; p++ {
17         if primes[p] {
18             primeNumbers = append(primeNumbers, p)
19         }
20     }
21     return primeNumbers
22 }
```

Analysis Tests ...



No auto suggestions? Try human mentoring.

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

Get mentoring

Simple Linked List

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

Passed

Analysis Tests ...

```
36 func (l *List) Push(element int) {
37     newNode := &Element{Value: element}
38
39     if l.head == nil {
40         l.head = newNode
41         l.tail = newNode // Initialize tail for the first element
42         return
43     }
44     l.tail.next = newNode // Add to the end of the list using the tail
45     l.tail = newNode      // Update tail
46 }
47
48 func (l *List) Pop() (int, error) {
49     if l == nil || l.head == nil {
50         return 0, errors.New("empty list")
51     }
52
53     val := l.tail.Value.(int) // Type assertion to int. Handle non-int values if needed.
54     if l.head == l.tail {    // Only one element
55         l.head = nil
56         l.tail = nil
57         return val, nil
58     }
59     current := l.head
60     for current.next != l.tail {
61         current = current.next
62     }
63     current.next = nil
64     l.tail = current
65
66     return val, nil
67 }
68
69 func (l *List) Array() []int {
70     if l == nil || l.head == nil {
```

No auto suggestions? Try human mentoring.

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

Get mentoring

Forth

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 Forth exercise and start writing better Go.

Get mentoring

```
func Forth(lines []string) ([]int, error) {
    macros := make(map[string]string) // defined macros
    for i := range lines {
        line := strings.ToLower(lines[i])
        if line[0] == ';' { // This refers to us defining a macro
            parts := strings.SplitN(line[2:len(line)-2], " ", 2) // Extract name and definition
            macros[parts[0]] = expandMacros(parts[1], macros) // Store macro which can be nested
        } else {
            lines[i] = expandMacros(line, macros) // Expand any macros in the line.
        }
    }
    stack := make([]int, 0, 8)
    // Process the last line (the actual Forth program).
    for _, word := range strings.Fields(lines[len(lines)-1]) { // Split the line into words
        if op, ok := ops[word]; ok { // Check if it's a builtin op
            argsIdx := len(stack) - op.numArgs // Calculate the starting index of arguments
            if argsIdx < 0 { // Check for stack underflow
                return nil, errors.New("stack underflow")
            }
            if op.isValid != nil && !op.isValid(stack[argsIdx:]) { // Validate arguments
                return nil, errors.New("invalid arguments")
            }
            stack = append(stack[:argsIdx], op.run(stack[argsIdx:]))... // Apply op
        } else if num, err := strconv.Atoi(word); err == nil { // try number parsing
            stack = append(stack, num) // Push the number into stack
        } else {
            return nil, errors.New("invalid word") // Invalid word
        }
    }
    return stack, nil // Return the final stack and any error
}
```

50 % Completion

You're 50.4% through the Go track. Over halfway there! 🚀

Completed 71 In-progress 0 Available 70 Locked 0 Total Exercises 141

71
Exercises completed >

34
Concepts learnt >

Bo Co Co Ty Co

34
Concepts mastered >

Sl Ra Ar Fi Re

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

 PROLIFIC PADAWAN	 LAUDABLE LEARNER	 LOCKED	 LOCKED	 LOCKED	 LOCKED
 FUNDAMENTAL					