I have neither given or received, nor have I tolerated others’ use of unauthorized aid.

Joseph Leveille

Chosen Problem #1: Pig

Why: This project involves the random function and user input, so it was a good introduction to some useful functionalities in Go.

Solution Code:

//Joe Leveille PS7 project 1 - Pig Dice Game

package main

import "fmt"

import "math/rand"

func main(){

win,currPlay, sum, currRoll:=0,0,0,0

var score [2] int

score[0],score[1]= 0, 0

choice := ""

for win == 0 {

fmt.Println("Player ",currPlay+1,", it is your turn")

if sum > 1 {

fmt.Println("Would you like to roll again? (Y/N)")

fmt.Scan(&choice)

if choice == "N" || choice == "n" {

score[currPlay] += sum

sum = 0

if score[currPlay] >= 100 {

win = currPlay +1

continue

}

currPlay = (currPlay + 1) % 2

continue

}

}

currRoll = rand.Intn(6) + 1

fmt.Println("You roll: ",currRoll)

if currRoll == 1 {

fmt.Println("That ends your turn!")

sum = 0

currPlay = (currPlay + 1) % 2

continue

} else{

sum += currRoll

fmt.Println("Your current sum for this turn: ", sum)

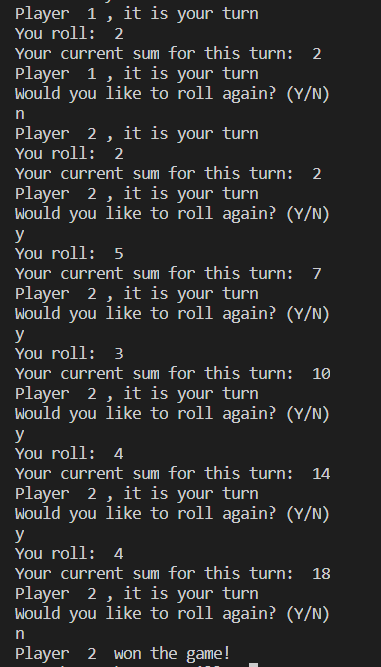
}

}

fmt.Println("Player ", win, " won the game!")

}

Output Screenshot: (at the end of a long game)



Chosen Problem #2: Hawaiian Word Pronunciation

Why: Seemed to be a simple problem that gave me practice using strings, if statements and for loops in Go.

Solution Code:

package main

import "fmt"

func main(){

wordIn := ""

repeat := true

rep := ""

//fmt.Println(s.Contains("tester", "e"), " poop", test)

for repeat{

fmt.Println("Input a word to Hawaiian-ify: ")

fmt.Scan(&wordIn)

//TODO: check for each letter combo, printing

for i := 0; i<len(wordIn);i++{

//example usage: fmt.Print(string(wordIn[i])," ")

if string(wordIn[i]) == "w"{

//check w cases

continue

} else if string(wordIn[i]) == "p" || string(wordIn[i])=="k"||string(wordIn[i])=="h"||

string(wordIn[i]) == "l" || string(wordIn[i]) == "m"||string(wordIn[i])=="n"||

string(wordIn[i]) == "'"||string(wordIn[i]) == " "{

fmt.Print(string(wordIn[i]))

continue

} else if string(wordIn[i]) == "a" {

if i == len(wordIn)-1{

fmt.Print("ah")

} else if string(wordIn[i+1]) == "i"||string(wordIn[i+1]) == "e"{

fmt.Print("eye")

i++

} else if string(wordIn[i+1]) == "o"||string(wordIn[i+1]) == "u"{

fmt.Print("ow")

i++

} else{

fmt.Print("ah")

}

} else if string(wordIn[i]) == "e"{

//check e cases

if i == len(wordIn)-1{

fmt.Print("eh")

} else if string(wordIn[i+1]) == "i"{

fmt.Print("ay")

i++

} else if string(wordIn[i+1]) == "u"{

fmt.Print("eh-oo")

i++

} else{

fmt.Print("eh")

}

} else if string(wordIn[i]) == "i"{

//check i case

if i == len(wordIn)-1{

fmt.Print("ee")

} else if string(wordIn[i+1]) == "u"{

fmt.Print("ew")

i++

} else{

fmt.Print("ee")

}

} else if string(wordIn[i]) == "o"{

//check o cases

if i == len(wordIn)-1{

fmt.Print("oh")

} else if string(wordIn[i+1]) == "i"{

fmt.Print("oy")

i++

} else if string(wordIn[i+1]) == "u"{

fmt.Print("ow")

i++

} else {

fmt.Print("oh")

}

} else if string(wordIn[i]) == "u"{

//check u case

if i == len(wordIn)-1{

fmt.Print("oo")

} else if string(wordIn[i+1]) == "i"{

fmt.Print("ooey")

i++

} else{

fmt.Print("oo")

}

} else {

fmt.Println("\n\nInvalid character present!")

break

}

if i < len(wordIn)-1 && string(wordIn[i+1])!=" " {

fmt.Print("-")

}

}

//TODO: check for each illegit letter

fmt.Println("\nWould you like to try another word? (Y/N)")

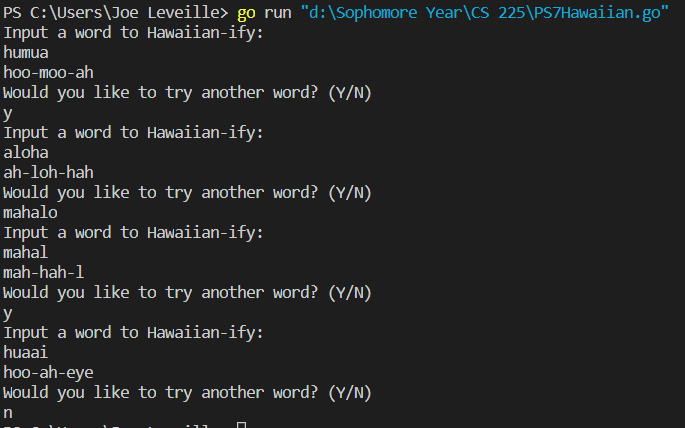
fmt.Scan(&rep)

if rep == "N"||rep=="n"||rep=="no" {repeat = false;}

}

}

Output Screenshot:



Chosen Problem #3: Presdigitation

Why: This involved coming up with an interesting, but decently simple algorithm, so it was a fun little project that forced some usage of the math & rand libraries, as well as knowing the uses of slices vs arrays.

Solution Code:

package main

import "fmt"

import "math/rand"

import "math"

func main(){

//Read in which digit (MSD is 1st, then 2nd down to LSD) to look at

dig, length, j:= 0,0, 0

fmt.Println("Which digit should be counted?(1-4)")

fmt.Scan(&dig)

//Produce random data set to evaluate

fmt.Println("How many numbers should be used?")

fmt.Scan(&length)

nums := make([]int, length)

var digs [10] int

digs[0],digs[1],digs[2],digs[3],digs[4],digs[5],digs[6],digs[7],digs[8],digs[9]= 0,0,0,0,0,0,0,0,0,0

for i:=0;i<length;i++{

nums[i] = rand.Intn(1000)

//Testing: fmt.Println("New Num: ",nums[i])

temp:=nums[i]

if float64(nums[i]) > math.Pow10(dig-1) {

//Testing: ///fmt.Println("Reached point 1")

for j=1;temp>int(math.Pow10(dig));j++ {

temp = temp/10

}

digs[temp % 10]++

}

}

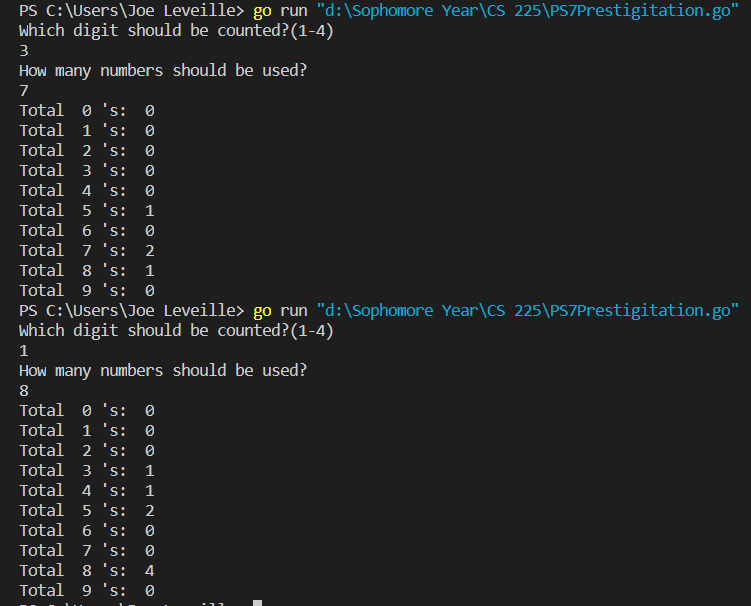
for i:=0;i<10;i++{

fmt.Println("Total ",i,"'s: ", digs[i])

}

}

Output Screenshot:



First test does not total to 7 digits because not all 7 numbers have 3 digits, so they do not increment any of the counts.