//

// main.swift

// Lab2

//

// Created by JOE FOY (000037286) on 9/4/15.

// Copyright (c) 2015 JOE FOY (000037286). All rights reserved.

// Revision 0: 9/4/15, initial release

// 1: 8/13/16 modified for year 2016

//

// Exploration 1: Run the code to see that it works. Notice that three selections

// discriminant,convert mph, convert fps will not work until you implement

// your pseudocode.

// Exploration 2: Write the discriminant function

// Exploration 3: Run the code, but enter a number that is not between 0 and 6 for the

// selection. Notice what happens

// Exploration 4: Run and select slope. Notice that if the answer should be a decimal

// number, this will not work correctly.

// Exploration 5: In function "factorial", comment out the line that updates count.

// Notice what happens

// Exploration 6: Finish the mph\_to\_fps function

// Exploration 7: Finish the fps\_to\_mph function

// Exploration 8: The menu is a series of lines inside "main". Write a function titled

// "menu" that does the same thing, move the code into the function, and

// have the function return an integer to the calling program

// Exploration 9: (LATER, AFTER CLASSROOM EXPLANATION) Replace the multiple if-statements

// with a switch-case block

// Exploration 10: Add two more unit conversions of your choice to the menu, also write

// their functions and test your code

//

//

import Foundation

import Swift

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Function strinput returns a String which it reads from the Console

//

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func strinput() -> String {

let keyboard = NSFileHandle.fileHandleWithStandardInput()

let inputData = keyboard.availableData

let strData = NSString(data: inputData, encoding: NSUTF8StringEncoding)!

return strData.stringByTrimmingCharactersInSet(NSCharacterSet.newlineCharacterSet())

} // end of function strinput

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Function intinput returns an integer which it reads from the Console

//

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func intinput() -> Int32 {

let keyboard = NSFileHandle.fileHandleWithStandardInput()

let inputData = keyboard.availableData

let strData = NSString(data: inputData, encoding: NSUTF8StringEncoding)!

strData.stringByTrimmingCharactersInSet(NSCharacterSet.newlineCharacterSet())

return strData.intValue

} // end of function intinput

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// Function doubleinput returns an integer which it reads from the Console

//

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func doubleinput() -> Double {

let keyboard = NSFileHandle.fileHandleWithStandardInput()

let inputData = keyboard.availableData

let strData = NSString(data: inputData, encoding: NSUTF8StringEncoding)!

strData.stringByTrimmingCharactersInSet(NSCharacterSet.newlineCharacterSet())

var dvalue: Double = 0

dvalue = strData.doubleValue

return dvalue

} // end of function intinput

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func Pythagorean(a:Double, b:Double) ->(mag:Double, ang:Double)

{

let PI = 3.14159

let magnitude = sqrt(a\*a+b\*b)

var angle = atan(b/a)

angle = angle \* 180/PI

return(magnitude, angle)

} // end of function Pythagorean

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func slope()

{

var rise, run: Int32

var slope:Double

print("Slope Calculation")

print("Enter rise value: ", terminator: "")

rise = intinput()

print("Enter run value: ", terminator: "")

run = intinput()

// Exploration 3: this calculation will not work correctly if slope is a decimal number

print("\n Slope is: \(rise/run)\n")

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Exploration 2: Finish this function

func discriminant()

{

var a, b, c, disc: Int32

print("Discriminant Calculator")

} // end of discriminant

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func factorial()

{

var num, fvalue, count: Int32

print("Factorial Calculation")

print("Enter value: ", terminator: "")

num = intinput()

fvalue = 1

// while variable must be initialized

count = 1

while (count <= num)

{

fvalue = fvalue \* count

count++ // this line updates the value of count

} // end of while

print("\nFactorial is: \(fvalue)")

} // end of factorial

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Exploration 6: Finish this function

func mph\_to\_fps()

{

var mph: Int32

let FEETINMILE = 5280.0

let SECINHOUR = 3600.0

var converted: Double

print("Miles per hour to feet per second converter")

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Exploration 7: Finish this function

func fps\_to\_mph()

{

var fps: Int32

let FEETINMILE = 5280.0

let SECINHOUR = 3600.0

var converted: Double

print("Feet per second to miles per hour converter")

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

func arithmetic\_sum()

{

var upper, sum, count: Int32

print("Arithmetic Sum")

print("Calculates sum of numbers from 1 to an upper limit")

print("Enter upper limit: ", terminator: "")

upper = intinput()

sum = 0

for count = 1; count <= upper; count++

{

sum = sum + count

} // end of for

print("\nSum is: \(sum)")

} // end of arithmetic\_sum

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var done: Bool = false

var x, y, z, a, b, c: Int32

var select:Int32 = 0

while !done

{

//

// For Exploration 8, move these into a new function "menu", which returns an integer

// assign the integer to variable “select” so that the rest of the code works

print("\nSecond Swift Program")

print("Operation Selection")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Slope 1")

print("Discriminant 2")

print("Factorial 3")

print("Convert mph to fps 4")

print("Convert fps to mph 5")

print("Arithmetic sum 6")

print("Quit 0")

print("Enter selection: ")

select = intinput()

// For Exploration 9, replace the following statements with a switch-case block

if select == 0 {done = true}

if select == 1 {slope()}

if select == 2 {discriminant()}

if select == 3 {factorial()}

if select == 4 {mph\_to\_fps()}

if select == 5 {fps\_to\_mph()}

if select == 6 {arithmetic\_sum()}

if ((select < 0) || (select > 6)) {print("Must enter an integer between 0 and 6")}

} // end of while