ASSIGNMENT 2

Compiler used - <https://www.onlinegdb.com/online_swift_compiler>

**Exercise 15**.

Assign a tuple with two values in it to a constant named

player.

let name: String = "John"

let num1 = 5

**Exercise 16.**

OK, now you have a player tuple. Decompose (i.e. split) the

number and the name into two constants named number and

name. This could be done in at least three ways.

var playerofhockey = (name: "Henry", number: 25)

print(playerofhockey.name)

print(playerofhockey.number)

ANSWER:

Henry

25

**Exercise 17.**

Can a constant have an optional type?

Yes, a constant can have an optional type

let value: Int? = nil

print(value)

let otherValue: Int? = 6

print(otherValue)

ANSWER:

Returned error

**Exercise 18.**

Why doesn’t this work? What needs to be added to value on

the second line for this to work?

let value: Int? = 17

let banana: Int = value

ANSWER: MIssing Default value

**Exercise 19:**

If value in the previous exercise had been nil, what would

have happened if you force unwrapped value?

let value: Int? = nil

let banana: Int = value!

ANSWER: There would be an error as unexpectedly nil found while unwrapping an Optional value.

**Exercise 20:**

What would be a better way to assign value to the banana

constant?

**Exercise 21:**

Write a Swift program to compute the sum of the two

integers. If the values are equal return the triple their sum.

var num1 = 243

var num2 = 243

var sum1 = num1 + num2

print("Expression: 190 + 243, Result:", sum1)

if sum1 = = sum1 {

print (sum1 + sum1 + sum1)

**Exercise 22:**

Write a Swift program to check if 5 appears as either the first

or last element in a given array of integers. The array length

should be 1 or more.

func check\_first\_last(\_ arra: [Int]) -> Bool {

guard arra.count > 0 else

{

return false

}

if arra.first == arra.last

{

return true

} else

{

return false

}

}

print(check\_first\_last([1, 2, 3]))

print(check\_first\_last([1, 2, 3, 1]))

print(check\_first\_last([1, 2, 2, 1]))

print(check\_first\_last([1]))

**Exercise 23:**

Write a Swift program to create a new array with the elements

in reverse order of a given array of integers.

import Swift

// Creating an array of number

// Here the array is of float type

var newNumber = [23.034, 1.90, 9.1, 32.34, 560.44, 21.23]

print("Array before reversing:", newNumber)

// Reverse the newNumber array

// Using reverse() function

newNumber.reverse()

// Displaying the final result

print("Array after reversing:", newNumber)

**Exercise 24:**

Write a Swift program to rotate the elements of an array of

integers to left direction. Therefore {1, 2, 3} yields {2, 3, 1}

func rotateLeft(arrToRotate: inout [Int], positions: Int){

if arrToRotate.count == 0 || positions == 0 || positions > arrToRotate.count{

print("invalid")

return

}

arrToRotate = arrToRotate.dropFirst(positions) + arrToRotate.dropLast(arrToRotate.count-positions)

}

var numbers : [Int] = [1, 2, 3, 4, 5]

rotateLeft(arrToRotate: &numbers, positions:2)

print(numbers) //prints [3, 4, 5, 1, 2]

**Exercise 25:**

Write a Swift program to compute the sum of all the elements

of a given array of integers and length 4.

import Swift

let numbers = [1, 12, 2, 9, 27]

let total = numbers.reduce(0, +)

print (total)

**Exercise 26:**

Write a Swift program to compute and return the absolute

difference of n and 51, if n is over 51 return double the

absolute difference.

func diff\_51(x: Int) -> Int {

if x > 51

{

return (x - 51) \* 2

}

else

{

return 51 - x

}

}

print(diff\_51(x: 45))

print(diff\_51(x: 61))

print(diff\_51(x: 21))

Exercise 27.

Write a Swift program to accept two integer values and return

true if one is negative and one is positive. Return true only if

both are negative.

func test\_num(x: Int, y: Int) -> Bool {

if x > 0 && y < 0

{

return true

}

else if x < 0 && y > 0

{

return true

}

else if x < 0 && y < 0

{

return true

}

else

{

return false

}

}

print(test\_num(x:12, y:-5))

print(test\_num(x:-12, y:5))

print(test\_num(x:-12, y:-5))

print(test\_num(x:12, y:5))

Exercise 29:

Write a Swift program that return true if either of two given

integers is in the range 10..30 inclusive.

func in1030(a: Int, b: Int) -> Bool {

if a >= 10 && a <= 30

{

return true

}

else if b >= 10 && b <= 30

{

return true

}

else

{

return false

}

}

print(in1030(a: 15, b: 40))

print(in1030(a: 55, b: 9))

print(in1030(a: 11, b: 25))

**Exercise 30.**

Write a Swift program to change the first and last character

of a given string.

func diff\_51(x: Int) -> Int {

if x > 51

{

return (x - 51) \* 2

}

else

{

return 51 - x

}

}

print(diff\_51(x: 45))

print(diff\_51(x: 61))

print(diff\_51(x: 21))