Lab Exercise 5

Control Flow

for

Exercise 5.1

Write a `for` loop that prints the values 1 through 5, each on a separate line, without using an array.

Exercise 5.2

Write a `for` loop that prints the values in array `[1, 2, 3, 4, 5]`, each value on a separate line.

Exercise 5.3

Write a `for` loop that, without iterating over the array, prints the values in array `[2, 6, 11, 19, 25]`, each value on a separate line. I.e. use an index.

Exercise 5.4

Write a `for` loop that prints the values in array `[2, 6, 11, 19, 25]`, each value *and its index in the array* on a separate line.

while

Exercise 5.5

Write a `while` loop that increments a counter variable until it reaches 10.

Exercise 5.6

Write a `repeat...while` loop that increments a counter until it reaches 10.

if

Exercise 5.7

Using `if`, print "It's too hot" if the temperature is 30 degrees or above. Print "It's too cold" if the temperature is less than 0 degrees. Finally print "It's tolerable" for any other temperature.

Exercise 5.8

Write a single `if` statement that converts a String to an Int and then checks if that Int is 1337. If it is, print "The value is 1337".
Solution

Switch

Exercise 5.9

Given the variable `value`:
```
let value: Int = 1337
```
Write a `switch` that prints "elite" if the value is 1337, "the meaning of life" if the value is 42, and "some number" otherwise.

Exercise 5.10

Given the variable `value`:
```
let value: Int = 1337
```
Write a `switch` that prints "a number we care about" if the value is 42, 1337, or 4711 using a single case. Print "who cares" otherwise.

Exercise 5.11

Given the variable `animal`:
```
let animal: String = "tiger"
```
Write a `switch` (using `fallthrough`) that prints "Animal is a tiger" and also prints "Animal is a cat" if `animal` is "tiger". It should also print "Animal is a cat" if `animal` is "cat". The line that prints "Animal is a cat" may only exist once in the code. If animal is not a cat or a tiger, print "Animal is some other type of animal".

Exercise 5.12

Given the variable `distance`:
```
let distance: UInt = 10
```
Write a `switch` using interval matching that...

1. Prints "Here" if distance is 0.
2. Prints "Immediate vicinity" if distance is less than 5 but more than 0.
3. Prints "Near" if distance is between 5 and 15, including 15.
4. Prints "Kind of far" if distance is more than 15 and less or equal to 40.
5. Prints "Far" if distance exceeds 40.

Exercise 5.13

Given the variable `vector3D`:
```
let vector3D: (x: Int, y: Int, z: Int) = (x: 3, y: 2, z: 5)
```

Write a `switch` that prints the `y` value if the vector has a `z` value of 5 or an `x` value of 12.

Exercise 5.14

Given the variable `vector3D`:

```
let vector3D: (x: Int, y: Int, z: Int) = (x: 3, y: 2, z: 6)
```

Write a `switch` that prints the `x` value if the vector has a `z` value that is equal to the `y` value multiplied by 3.

guard

Exercise 5.15

Rewrite the following function using guard statements. You can use 2 guard statements or 1 compound guard statement. Actually, try both. :-)

```
func printIfPositiveInteger(number: String) {
    if let value = Int(number) {
        if value > 0 {
            print(value)
        }
    }
}

printIfPositiveInteger(number: "abc")
printIfPositiveInteger(number: "-10")
printIfPositiveInteger(number: "10")
```

Solution

```
// ----- With 2 guards -----

func printIfPositiveInteger2(number: String) {
    guard let value = Int(number) else { return }
    guard value > 0 else { return }

    print(value)
}

printIfPositiveInteger2(number: "abc")
printIfPositiveInteger2(number: "-10")
printIfPositiveInteger2(number: "20")

// ------- With compound guard --------

func printIfPositiveInteger3(number: String) {
    guard let value = Int(number), value > 0 else { return }

    print(value)
}
```