Due Tuesday, March 17, 2020

Random Problem Set

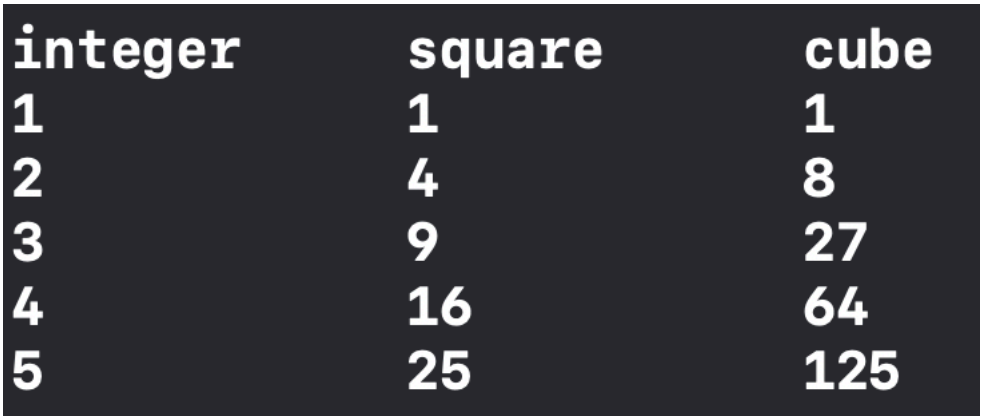
Do as many of these as makes sense to you to do. They vaguely progress from easy to hard. They should only need variables/if-else statements/for or while loops to complete.

```
func print1to4() {
    // this function should print "1 2 3 4" three different ways
    // method #1 using a single print statement and no loop
    // method #2 using a four print statements
    // method #3 using a single print statemnt and a loop
    // In comments in the function, discuss the pros/cons to each approach
}

func printStatsFor(number1 : Int, number2 : Int, number3 : Int) {
    // this function should take in 3 numbers and print the following
    // sum of the three numbers
    // average of the three numbers
    // product of the three numbers
    // smallest of the three numbers
    // largest of the three numbers
}

func isOdd(number : Int) -> Bool {
    // this functions should return true if the number is odd, and false if the number provided is even
}
```

func **printSquaresAndCubesTable(to number: Int)** {

// this function should calculate and print the squares from 1 to number.

// Ex print with number = 5



integer	square	cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125

// Note! Don't use any functions to find the powers, just use the arithmatic we've covered so far

}

func **raise(_ number : Int, toThePowerOf exponent: Int)** -> Int {

// this function should take **number** and multiply it to itself **exponent** times

// raise(2, toThePowerOf: 3) should return 8

// raise(1, toThePowerOf: 20) should return 1

// raise(10, toThePowerOf: 3) should return 1000

}

```
func isAPalindrome(number: Int) -> Bool {
    // this function should take in a 5 digit integer, and return true if the number would be the same if the digits were reversed, and false if it would make a new number
    // ex:
    // isAPalindrome(number: 12321) should return true
    // isAPalindrome(number: 55555) should return true
    // isAPalindrome(number: 12361) should return false
}

func binaryToInt(_ binaryString : String) -> Int {
    // this function should take in a string of 1s and 0s and turn it into the equivalent decimal number
    // If you're not farmiliar with binary, this is a great time to learn about it!
    // here are a few examples to get you started
    // binaryToInt("0") -> 0
    // binaryToInt("1") -> 1
    // binaryToInt("10") -> 2
    // binaryToInt("11") -> 3
    // binaryToInt("100") -> 4
    // binaryToInt("101") -> 5
    // binaryToInt("110") -> 6
    // binaryToInt("111") -> 7
    // binaryToInt("1000") -> 8
    // binaryToInt("1001") -> 9
    // binaryToInt("1010") -> 10
```

```
    // binaryToInt("1011") -> 11
    // binaryToInt("1100") -> 12
    // binaryToInt("1101") -> 13
    // binaryToInt("1110") -> 14
    // binaryToInt("1111") -> 15
}

func printACheckerboard() {
    // print the following pattern using only one of each of these statements:
print("*", terminator: ""), print(" ", terminator: ""), print()
```

```
}

func printTriangles(height : Int) {
    //print the patterns shown here ->
    // (shown are height = 5)
}
```