

Name: \_\_\_\_\_

# Swift: Functions

## Part 1: Writing an Is Prime Function

A previous homework assignment was to write a program that determined if an input number is prime or not. Now, you should write this as a function, which has an Int parameter and returns a Bool. Feel free to look at your homework code for this and copy it, but make sure to put it into a function properly.

```
func isPrime(number: Int) -> Bool {  
    // Many possible answers here, just test your code to see if it's  
    right.  
    // This is probably the simplest solution  
  
    if number <= 1 {  
        return false  
    }  
  
    for divisor in 2..  
number {  
        if number % divisor == 0 {  
            return false  
        }  
    }  
  
    return true  
}
```

## Part 2: Refactoring Code

Below I have written a program that prints out primes forever, or until they get as big as  $2^{63}$  at least. However, it can be re-written in a way that is structurally superior (refactoring), using the `isPrime` function from Part 1. Assuming the `isPrime` function is defined properly, rewrite my code to take advantage of it. Your code should be **much** shorter than mine.

```
for possiblePrime in 2...Int.max { // Int.max == 9223372036854775807
    // First I check if possiblePrime is actually prime
    var isPrime = true
    for divisor in 2..<possiblePrime {
        if possiblePrime % divisor == 0 {
            isPrime = false
            break
        }
    }

    if isPrime {
        print(possiblePrime)
    }
}
```

```
for possiblePrime in 2...Int.max { // Int.max == 9223372036854775807
    if isPrime(number: possiblePrime) {
        print(possiblePrime)
    }
}
```