

Assignment 1 (100 points)
Due Sept 9 11:59 pm
Objectives

1. Get started using Xcode.
2. Learn basic Swift

Programing Problems (10 points each)

1. A palindrome is a string that reads the same forwards and backwards. We will call a number a palindrome if it reads the same forwards and backwards. So 191 is a palindrome. Write a function that returns the number of palindrome numbers between 0 and n.
2. If we list all the natural numbers below 20 that are multiples of 3 or 5 but not multiples both of 3 and 5, we get 3, 5, 6, 9, 10, 12, 18. The sum of these multiples is 63. Write a function, `sumMultiples3_5`, that returns the sum of the multiples of 3 and 5 less than N. So

`sumMultiples3_5(20) == 63`

3. Write a function `patternCount` with two arguments. The first arguments is a string, lets call it `text`, and the second argument is also a string, call it `pattern`. The function `patternCount` returns the number of times the pattern occurs in the text. For example:

`patternCount(string: "abababa", pattern: "aba") == 3`
`patternCount(string: "aaaaa", pattern: "aa") == 4`
`patternCount(string: "Abcde", pattern: "abc") == 0`

4. Write a Swift function, call it `popularClasses`, that has one argument an array of sets of courses and returns the intersection of all the sets in the array. In the code below `popularClasses` would return `{"CS237"}`.

`let studentA: Set = ["CS101", "CS237", "CS520"]`
`let studentB: Set = ["CS101", "Math245", "CS237"]`
`let studentC: Set = ["CS237", "CS560"]`
`popularClasses([studentA, studentB, studentC])`

5. Write a Swift function, call it `average`, that has one argument an array of `Ints` and returns an optional double, which is the average of the inputs `ints`. If the input array is empty return the optional value `nil`.
6. Write a Swift function `average2` which is the same as `average` in #4 except that the input is an array of optional `ints`.

7. Write a Swift function `digitDistribution` with one argument an array of numbers integers and returns a dictionary of the distribution of digits in the array of numbers. So with an input of `[112, 24, 15]` `digitDistribution` will return `[1: 3, 2: 2, 4: 1, 5: 1]`.
8. Write a Swift function `digitDistribution` that has two arguments, a string and an `Int`. The function does the same thing as `digitDistribution` in problem 7 except that the dictionary that it returns only contains the digits that occur at least as many times as the second argument.

```
digitDistribution(numbers: [112, 24, 15] , count: 1) returns [1: 3, 2: 2, 4: 1, 5: 1]
digitDistribution(numbers: [112, 24, 15] , count: 2) returns [1: 3, 2: 2]
digitDistribution(numbers: [112, 24, 15] , count: 3) returns [1: 3]
digitDistribution(numbers: [112, 24, 15] , count: 4) returns [:]
```

9. Write a Swift function `digitDistribution2` that has the same arguments as `digitDistribution` in problem 8 and returns the same result. However give the second argument a default value of 2 so we can call the function with one or two arguments as shown below.

```
digitDistribution2(numbers: [112, 24, 15] , count: 3) returns [1: 3]
digitDistribution2(numbers: [112, 24, 15] ) returns [1: 3, 2: 2]
```

10. Write a Swift function `digitDistribution3` that has one argument an `Int`, which has the same role as the second argument of `digitDistribution`. `digitDistribution3` returns a function. The return function has one argument an array of integers. When evaluated the returned function returns the dictionary of the distribution of digits in the array of numbers. But as in problem 8 it only contains the digits that occur as many times as the argument to `digitDistribution3`. See examples below.

```
let testA = digitDistribution3(2)
testA(numbers: [112, 24, 15]) returns [1: 3, 2: 2]
testA(numbers: [111, 42, 4005]) returns [1:3, 0:2, 4:2]
let testB = digitDistribution3(3)
testB(numbers: [111, 42, 4005]) returns [1:3]
```

What to Turn in

Create a Xcode Playground for your assignment 1. Answer all questions in the single playground. Use a comment to separate each questions. Zip up the playground and turn in your zipped file using assignment 1 link on blackboard.

Late Penalty

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eighth day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.

