Swift Assignment #3
Functions and Closures

CS275 Fall 2021
15 points
due Tuesday, Sept. 28th, 11:59 pm

1 Functions

You'll write several functions for this assignment that operate on sets and returns tuples.

1.1 Primes and composites

Write this function:

```
func categorizeIntegers(in vals: Set<Int>)
      -> (prime: Set<Int>, composite: Set<Int>, other: Set<Int>)
```

This will take an input set of integers and categorize them by whether or not each is prime. For the purpose of this function, consider any integer less than two (including negative integers) to be neither prime nor composite (they will fall into the "other" category). To do this, write also the following function:

```
func isPrime(_ val: Int) -> Bool?
```

The function returns `true` if the input value is greater or equal to two and is prime; returns `false` if the input value is greater or equal to two and is not prime; and returns `nil` if the input value is less than two.

1.2 Healthy foods

Write a function to take a set of foods (in the form of strings) and sort them into healthy and unhealthy foods. Specifically, write this function:

```
func sortFood(in food: Set<String>) -> (healthy: Set<String>, unhealthy: Set<String>)
```

The sorting criterion is the following: if the name of the food contains "sugar", then it's unhealthy; otherwise, it's healthy. So for example:

```
let myFoods: Set = ["apples", "chocolate frosted sugar bombs", "cauliflower",
                    "olives", "refined sugar", "sugar beets", "fritos", "sugar peas",
                    "fried dough", "broccoli", "broccoli au sucre", "broccoli au lapin"]
print("results, using simple discriminator:")
var results = sortFood(in: myFoods)
print("my healthy foods: \(results.healthy)")
print("my unhealthy foods: \(results.unhealthy)")
```

will produce the following:

```
results, using simple discriminator:
my healthy foods: ["cauliflower", "olives", "broccoli au lapin", "fried dough", "fritos",
  "apples", "broccoli au sucre", "broccoli"]
my unhealthy foods: ["sugar beets", "refined sugar", "sugar peas",
 "chocolate frosted sugar bombs"]
```

This clearly is not producing the right results:

- fried dough, fritos, and broccoli au sucre are not healthy
- sugar beets and sugar peas *are* healthy foods

To do a better job of discriminating between healthy and unhealthy foods, write this function:

```
func sortFoodSmart(in food: Set<String>,
                   with sortFunction: (String) -> Bool)
                       -> (healthy: Set<String>, unhealthy: Set<String>)
```

This will take a function (or a closure) that returns `true` if a food is healthy and `false` if a food is unhealthy.

Write a helper function with this behavior:

- sugar beets, sugar peas, and sugar snap peas are healthy
- fritos are unhealthy
- any food having "fried" in the name is unhealthy
- any food having "broccoli" in the name is healthy, unless it also has "sucre" in the name

Then use your function to with `sortFoodSmart(in:with:)`.

Note: there's actually no such thing as broccoli au sucre. There might or might not be a real dish named broccoli au lapin.

2 Testing your functions

Here is a sequence of calls to make:

```
let myFoods: Set = ["apples", "chocolate frosted sugar bombs", "cauliflower", "olives",
          "refined sugar", "sugar beets", "fritos", "sugar peas", "fried dough", "broccoli",
          "broccoli au sucre", "broccoli au lapin"]
print("results, using simple discriminator:")
var results = sortFood(in: myFoods)
print("my healthy foods: \(results.healthy)")
print("my unhealthy foods: \(results.unhealthy)")

print()
print("results, using smart discriminator")
results = sortFoodSmart(in: myFoods, with: isHealthy)
print("my healthy foods: \(results.healthy)")
print("my unhealthy foods: \(results.unhealthy)")

let wifeFoods: Set = ["bamba", "sugar snap peas", "eggs", "broccoli", "fried chicken"]
```

```
print()
print("Sarah's groceries, using smart discriminator:")
results = sortFoodSmart(in: wifeFoods, with: isHealthy)
print("Sarah's healthy foods: \(results.healthy)")
print("Sarah's unhealthy foods: \(results.unhealthy)")
```

And here's the output you should get:

```
results, using simple discriminator:
my healthy foods: ["cauliflower", "olives", "broccoli au lapin", "fried dough", "fritos",
 "apples", "broccoli au sucre", "broccoli"]
my unhealthy foods: ["sugar beets", "refined sugar", "sugar peas",
 "chocolate frosted sugar bombs"]

results, using smart discriminator
my healthy foods: ["apples", "broccoli", "broccoli au lapin", "sugar beets",
 "cauliflower", "olives", "sugar peas"]
my unhealthy foods: ["chocolate frosted sugar bombs", "broccoli au sucre",
 "fried dough", "fritos", "refined sugar"]

Sarah's groceries, using smart discriminator:
Sarah's healthy foods: ["broccoli", "eggs", "bamba", "sugar snap peas"]
Sarah's unhealthy foods: ["fried chicken"]
```

3 Graduate Students

No additional work for graduate students.

4 What to Submit

Do all of these in a single Xcode Playground. Submit just your Swift file. You can see your .swift file in your Playground directory, with the name `Contents.swift`. Rename it `assignment-three.netid.swift`, using your netid.