Swift Assignment #5
Palindromes and a Stack

CS275 Fall 2021
15 points
due Tuesday, Oct. 12th, 11:59 pm

1 Palindromes

A palindrome is a string that reads the same backwards and forwards, ignoring spaces and punctuation. For example: “Madam, I’m Adam”

1.1 Checking for a palindrome

Write this function:

```
func isPalindrome(_ s: String) -> Bool
```

This will take a string and return `true` if the string—ignoring spaces and punctuation—is a palindrome; and `false` otherwise. The function should also look at only the alphabetic characters, and it should ignore case. So for example:

```
print(isPalindrome("Madam, I’m Adam"))
true
```

Do not just reverse the string and see whether the reversed string is equal to the non-reversed string. Use a loop.

Hint: use `filter(_:)` and `lowercased()`.

1.2 A stack

We can define a stack data structure in Swift this way:

```
struct Stack<Element> {
    var items = [Element]()

    mutating func push(_ newItem: Element) {
        items.append(newItem)
    }

    mutating func pop() -> Element? {
        guard !items.isEmpty else {
            return nil
        }
        return items.removeLast()
    }
}
```

This uses a feature of Swift called *generics*, which we'll discuss later in the semester.

Then, for example:

```
var stack = Stack<Character>()
stack.push("a")
stack.push("b")
stack.push("c")
while let val = stack.pop() {
    print("value is \(val)")
}
value is c
value is b
value is a
```

Write a second palindrome-checker function that uses a stack:

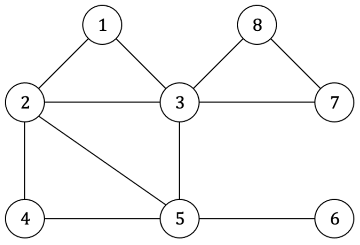
```
isPalindromeStack(_ s: String) -> Bool
```

In your function, you'll have a line that compares one character to a second character. Before this line, print out the two characters that your are comparing.

2 Graduate Students

Graduate students, and undergraduates for bit of extra credit: implement a depth-first search on an undirected graph, using your `Stack` structure. Define a `Node` class and a `Graph` class. Define `func dfs(_ node: Node)` on `Graph`. Every time you discover a child node in the DFS tree, print out the child node and the parent node, for example: `1 -> 2`.

Here's a sample graph to use:



Here's my output from a DFS traversal of this graph, starting at node 1:

```
tree: 1 -> 3
tree: 3 -> 8
tree: 8 -> 7
tree: 3 -> 5
tree: 5 -> 6
tree: 5 -> 4
tree: 4 -> 2
```

You might not get the same traversal that I get, but your output should show a depth-first traversal!

3 Testing Your Code

Make sure you're handling edge cases correctly. I will test your functions pretty rigorously.

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-five.netid.swift`, using your netid.