MAP523/DPS923 Midterm Test Winter 2016 (PART A)

Midterm Test PART A Friday March 11, 2016 @ 8:00 A.M.
SUBMIT using web submission form (link).

This question is worth 35 MARKS (35%)

PLEASE READ THE INSTRUCTIONS BELOW CAREFULLY!

Submission LINK: web submission form.
Password: X23-161-TEST
File Name: userid_PARTA.zip (where 'userid' is your my.seneca user id)

Specifications:
For this question you are required to create a Single View iOS application
using the Swift programming language.

Project name: userid_PARTA (where 'userid' is your my.seneca user id)
Device to use: iPhone 5
Simulator OS: 9.1 (Deployment Target)
Storyboard device size: 4-inch (Simulated Metrics Pane)

Once your project is completed, you must create an archive with a file extension of *.zip and a file name as indicated above (using your student user ID).

PART A Section 1: (10 MARKS)

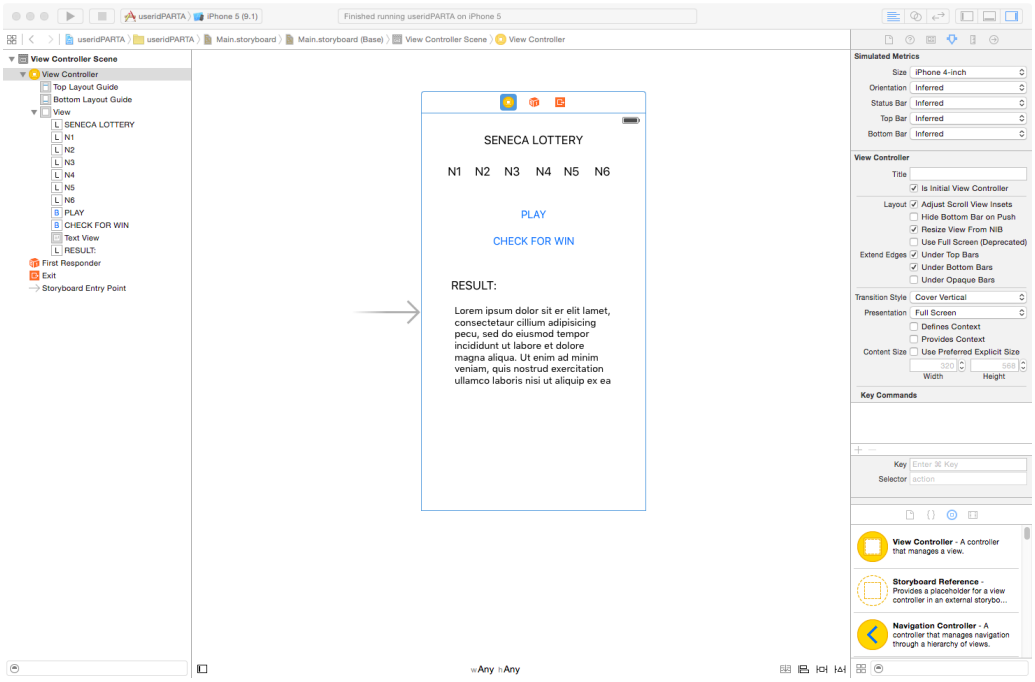
In the Main.storyboard create a scene with a title: "SENECA LOTTERY" (Label),
6 additional Labels with titles of: "N1" to "N6", a Button with title: "PLAY",
another button with title: "CHECK FOR WIN", a "RESULT" Label, and a TextView.

You are also required to create the appropriate outlets for the 6 labels and
the TextView, as well as actions for the buttons in the ViewController.swift class
(see below).

NOTE: The 6 Labels ("N1" to "N6") can be created as a "Collection" (an array) when
creating the Outlets for them!
The first one is created as a "Collection" and the others are dragged on top
of the existing outlet!
To loop through the collection, you could use:

```
for labels in yourLabelCollection {
    ...
}
```

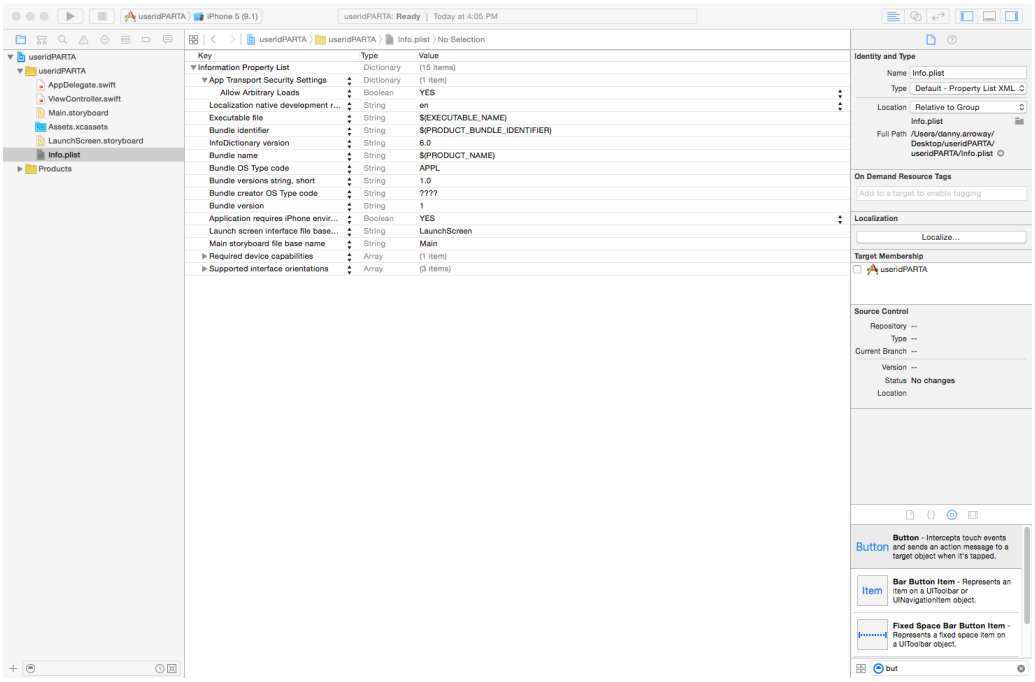
(This is not a requirement however)



PART A Section 2: (5 MARKS)

Modify the application's info.plist file to allow for network connections from anywhere. Recall, this is accomplished by adding the following keys:

```
<key>NSAppTransportSecurity</key>
    <dict>
        <key>NSAllowsArbitraryLoads</key>
        <true/>
    </dict>
```



PART A Section 3: (10 MARKS)

For the "PLAY" button, the app should generate 6 random numbers (stored in an array) from 1 to 49 such that no 2 values repeat and then display the numbers in sorted order (from low to high) in the Labels "N1" to "N6" respectively.

NOTE: To convert an Int to a String in Swift, you can use the

```
String(format: "...") string function.
```

NOTE: To check if duplicates are in a set (Array), you can use the:

```
Array(Set(yourArray)) function.
```

PART A Section 4: (10 MARKS)

For the "CHECK FOR WIN" button, the app should retrieve an array of numbers stored in a plist file at the following URL:
https://scs.senecac.on.ca/~danny.abesdris/numbers.plist

The contents of the plist can be read by into an NSArray object by using the following 2 functions:

```
NSURL(string: anyValidURLString)
NSArray(contentsOfURL: NSURL)
```

NOTE: If (for any reason) you cannot read the plist from the link above, then simply copy the values from the link provided into an array (**-3 mark deduction**).

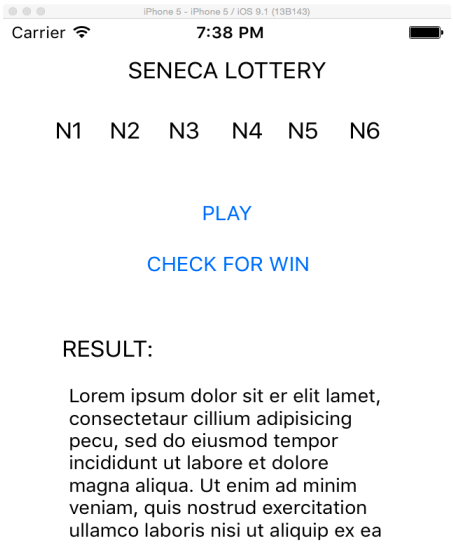
Once the values have been obtained, this function determines how many numbers in the randomly drawn values match the numbers in the numbers.plist file (displaying the result in the TextView).
The result output should be in the form:
Matched N numbers (N1, N2, N3 ...) (where N = number of matches and N1, N2, N3, etc are the values of the actuals numbers that matched).

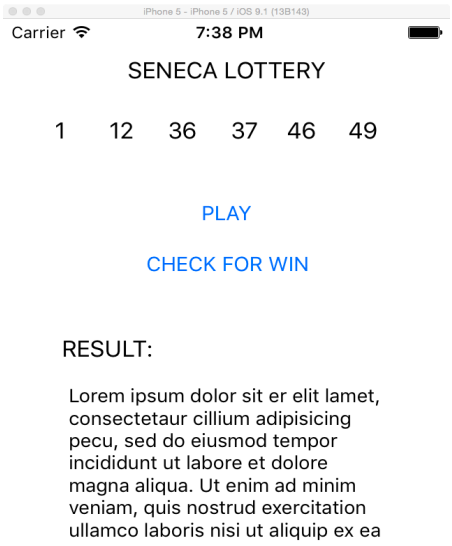
HINT: In Swift 2.1, the function:

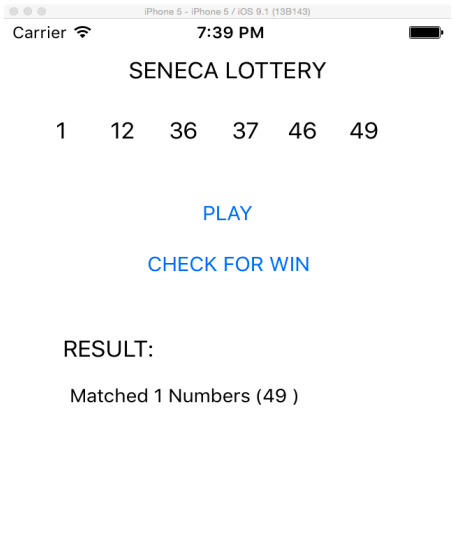
```
myArray.containsObject( )
```

can be used to determine if a value is present in an array!

App final output should look similar to:








Made on a
Mac

BEST VIEWED WITH
AnyBrowser