Storyboards and Navigation

As today exercise we can design an app with multiple view controllers and complex navigation.

Let's imagine we want to create an utility app to solve different types of math problems. The user can navigate the problems using a **UITabBarController** to switch between them. Each individual panel in the tabBar is a **UINavigationController** each one with two individual view controller, connected by a segue: the first one has the UI components that allow the user to insert the necessary input data; by triggering a "Solve!" button a second view controller will be pushed to navigation and this second view controller will compute the solution of the problem and somehow display the results.

Some input validation will always be required, depending on the specific problem (i.e. empty fields, negative numbers, zeroes, etc) before navigating to the next controller: you can either perform the validation before you trigger the segue and then trigger the segue manually only after the validation succeeds, using *performSegue(withIdentifier:sender:)* or alternatively trigger the segue every time the button is pressed but provide the delegate method *shouldPerformSegue(withIdentifier:sender:)* to prevent the segue to occur if validation fails.

> **optional**
> you can also use the **UIAlertController** to display a simple error message when validation fails.

Examples of problems to be solved included in your tabs, in rough order of complexity:

1. one tab can be used to solve a second degree equation, by providing the coefficients a, b, and c
   > **optional**
   > if the validation fails because the first coefficient is missing you can show a special error message notifying the user that the provided equation is a first degree equation and asking if he prefers to fix it by providing a new value for it or if he wants to go forward and solve the first degree equation anyway. You can again use **UIAlertController,** and to spice it up you can try using it with a preferred style .actionSheet to see the difference between a regular popup

2. one tab can be used to find the length of the third side of a right triangle by providing the other two sides, using the Pythagorean Theorem
   > **optional**

the view controller containing the solution might also try to draw the triangle. You can do so by adding a triangular **CAShapeLayer** to any view

3. one tab can be used to print all prime numbers in a given range, using the Sieve of Eratosthenes

   optional
   for large number this operation might take significant time, optionally you can display a **UIActivityIndicator** to notify the user computation is ongoing and/or use *DispatchQueue.global().async* to do the expensive operation on background)

4. one tab can use a Monte-Carlo simulation to compute the distribution of the sum of a set of dices containing any given number of dices having 4-faces, 6-faces, 8-faces, 10-faces, 12-faces and 20-faces. To generate a random number you can use *arc4random_uniform()*

   optional
   similarly to the previous case this simulation can take several seconds, depending how many runs you do. You can show a **UIActivityIndicator** and use *DispatchQueue.global().async* to do the expensive operation on background.