**Question 1**

One way to enable your app to trigger navigation from one fragment to the next is to use a LiveData value to indicate whether or not to trigger navigation.

What are the steps for using a LiveData value, called gotoBlueFragment, to trigger navigation from the red fragment to the blue fragment? Select all that apply:

* 🡺 In the ViewModel, define the LiveData value gotoBlueFragment.
* 🡺 In the RedFragment, observe the gotoBlueFragment value. Implement the observe{} code to navigate to BlueFragment when appropriate, and then reset the value of gotoBlueFragment to indicate that navigation is complete.
* 🡺 Make sure your code sets the gotoBlueFragment variable to the value that triggers navigation whenever the app needs to go from RedFragment to BlueFragment.
* Make sure your code defines an onClick handler for the View that the user clicks to navigate to BlueFragment, where the onClick handler observes the goToBlueFragment value.

**Question 2**

You can change whether a Button is enabled (clickable) or not by using LiveData. How would you ensure that your app changes the UpdateNumber button so that:

* The button is enabled if myNumber has a value greater than 5.
* The button is not enabled if myNumber is equal to or less than 5.

Assume that the layout that contains the UpdateNumber button includes the <data> variable for the NumbersViewModel as shown here:

<data>

<variable

name="NumbersViewModel"

type="com.example.android.numbersapp.NumbersViewModel" />

</data>

Assume that the ID of the button in the layout file is the following:

android:id="@+id/update\_number\_button"

What else do you need to do? Select all that apply.

* 🡺 In the NumbersViewModel class, define a LiveData variable, myNumber, that represents the number. Also define a variable whose value is set by calling Transform.map() on the myNumber variable, which returns a boolean indicating whether or not the number is greater than 5.  
    
  Specifically, in the ViewModel, add the following code:

val myNumber: LiveData<Int>

val enableUpdateNumberButton = Transformations.map(myNumber) {

myNumber > 5

}

* 🡺 In the XML layout, set the android:enabled attribute of the update\_number\_button button to NumberViewModel.enableUpdateNumbersButton.

android:enabled="@{NumbersViewModel.enableUpdateNumberButton}"

* 🡺 In the Fragment that uses the NumbersViewModel class, add an observer to the enabled attribute of the button.  
    
  Specifically, in the Fragment, add the following code:

*// Observer for the enabled attribute*

viewModel.enabled.observe(this, Observer<Boolean> { isEnabled ->

myNumber > 5

})

* In the layout file, set the android:enabled attribute of the update\_number\_button button to "Observable".