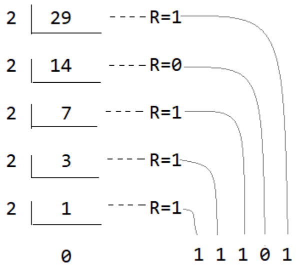
Activity 7-1

Convert Decimal to Binary

1. Write a recursive program to convert a given decimal number into binary.
2. Work through manually how to convert decimal number, say 29, into binary.

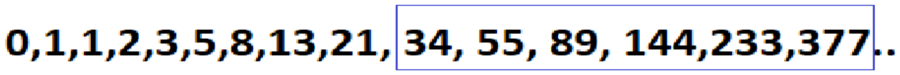


3. Identify the recursion – base case and generic case.
4. There is challenge of collecting all the remainders and form the proper binary.
   In the case of decimal 29, the correct answer is 11101.
   The first remainder collected is the MSB 1 (not the LSB).
5. Derive the algorithm and code it.

Activity 7-2

Fibonacci numbers

1. Given the 1st and 2nd numbers anywhere in the sequence, find the ***nth*** Fibonacci number starting from the given 1st number.
   For instance: Given 1st =34, 2nd = 55; Hence, 5th from 34 should be 377


0,1,1,2,3,5,8,13,21, 34, 55, 89, 144,233,377..

2. Modify the algorithm given in slide and code the recursive method with this signature:

int FibNumFrom (first, second, n)

Activity 7-3

Prime Number

1. A prime number is a number that can only be divided by itself and 1 without remainders.
2. Method 1 – write a method to determine is a given integer is a Prime Number using iterations.
3. Method 2 – solve it by recursive programming.
4. Assuming that input number is always >1. (1 is not a Prime)

(Maths: Fist determine the square root of the number. Then, check if it is divisible by any of the prime numbers that are smaller than the square root, it is not Prime).

(Our algorithm: Fist determine the square root of the number. Then, check if it is divisible by any of the ~~prime numbers~~ integers that are smaller than the square root, it is not Prime. Of course excluding 1).

Activity 7-4

Linear Binary Search

1. Given a sorted array with some integers.
2. Write a recursive method to check if a *key* (search key) exists in the array using binary search algorithm. Return the index of the *key* if found, or -1 if not found.

Activity 7-5

Big O

What is the Big-O for the code below?

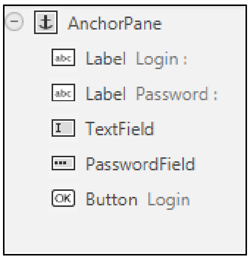
```
public static void main(String[] args) {

        int x = 64;
        int n = x;

        while (n>1){
          n = (int)Math.floor(n/2);

           for (int j=1; j<=x; j++) {
               System.out.println (j);
           }
        }
    }
```

Activity 7-6

Login and 2nd Window

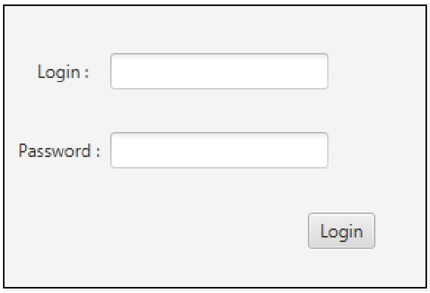
1. Create a JavaFX project with the following GUI.

Hierarchy

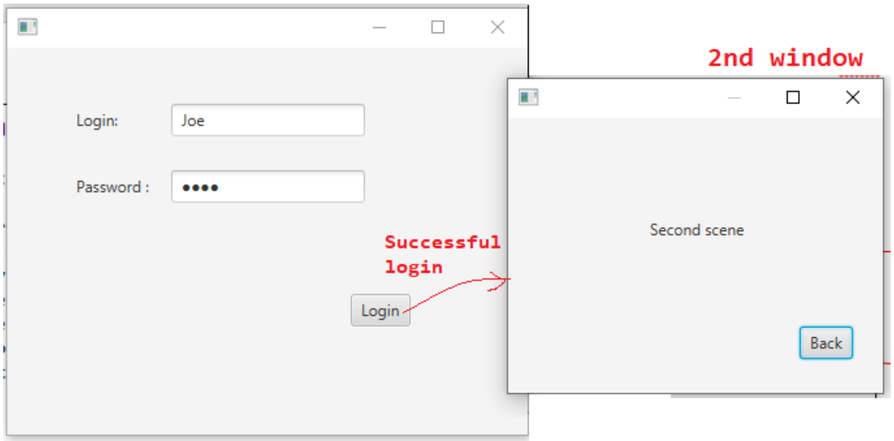


(controls)

GUI



2. Clicking the Login button will check if the Login and Password are correct. (Hardcode correct Login and Joe and Password as 1234)
3. If the login is successful, a 2nd window will appear.



4. In the 2nd window, there is a Label and a Back Button.
5. Clicking the Back button will close the 2nd window.
6. Create **Sample2.fxml** by copy-n-paste from **Sample.fxml**
7. Create **Sample2controller.java** by copy-n-paste **SampleController.java.**
8. Open **Sample2.fxml** using **SceneBuilder** to fix the GUI for the 2nd Window accordingly.

9. The following sample method helps to open the 2nd Window:

```
public void startSecondWindow() {
   try {
      AnchorPane root =
      (AnchorPane)FXMLLoader.load(getClass().getResource("Sample2.fxml"));
      Scene scene2 = new Scene(root,400,400);
      scene2.getStylesheets().add(getClass().getResource("application.css"
      ).toExternalForm());
      Stage Window2 = new Stage();
      Window2.initModality(Modality.APPLICATION_MODAL);
      Window2.setScene(scene2);
      Window2.show();

   } catch(Exception e) {
      e.printStackTrace();
   }
}
```

10. The following sample code helps to close the 2nd Window:

```
@FXML
 void onBackClicked(ActionEvent event) {

      Stage window2 = (Stage)btnBack.getScene().getWindow();
      window2.close();
 }
```