1. (2 points) (For this question, u*se the **GregorianCalendar** class:*
https://docs.oracle.com/javase/8/docs/api/java/util/GregorianCalendar.html)

Java API has the **GregorianCalendar** class in the **java.util** package, which you can use to obtain the year, month, and day of a date. The no-arg constructor constructs an instance for the current date, and **get(GregorianCalendar.YEAR)**, **get(GregorianCalendar.MONTH)**, and **get(GregorianCalendar.DAY_OF_MONTH)** return the year, month, and day.

Write a program to perform two tasks:

- Display the current year, month, and day.
- The **GregorianCalendar** class has the **setTimeInMillis(long)**, which can be used to set a specified elapsed time since January 1, 1970. Set the value to **1234567898765L** and display the year, month, and day.

2. (1 point) Here are four different contracts with preconditions and postconditions.

```
static int find1(int[] a, int val)
  requires: val occurs exactly once in a
  effects:  returns index i such that a[i] = val
```

```
 static int findStronger2(int[] a, int val)
  requires: val occurs at least once in a
  effects:  returns index i such that a[i] = val
```

```
 static int findStronger3(int[] a, int val)
  requires: val occurs at least once in a
  effects:  returns lowest index i such that a[i] = val
```

```
static int find4(int[] a, int val)
  requires: nothing
  effects:  returns index i such that a[i] = val,
              or -1 if no such i
```

Draw a diagram that shows the relationships among the contracts.

3. (1 point) In the following code, radius is private in the Circle class, and myCircle is an object of the Circle class. Does the highlighted code cause any problems? If so, explain why.

```
public class Circle {
  private double radius = 1;

  /** Find the area of this circle */
  public double getArea() {
    return radius * radius * Math.PI;
  }
  public static void main(String[] args) {
    Circle myCircle = new Circle();
    System.out.println("Radius is " + myCircle.radius);
  }}
```

4. (1 point) Consider these different specifications for the isValidShape function:

public static boolean isValidShape(char shape)

Spec A:

/**

* @param shape any character

* @return true iff shape is a lowercase aPredefinedType letter

*/

Spec B:

/**

* @param shape an English alphabetic character Spec B

* @return true iff shape is a lowercase aPredefinedType letter

*/

Spec C:

/**

* @param shape a lowercase English alphabetic character Spec C

* @return true

*/

Compare these specifications.