To the right are abbreviations the you may use. You must write everything else in full.

Abbreviation	Stands for
S.o.p	System.out.print
S.o.pln	System.out.println
I.pI	Integer.parseInt
JOP.sID	JOptionPane.showInputDialog

Short Java API descriptions (all methods are public):

```
class Integer:
  static int parseInt(String s) // = s's value, as an int.
class Double:
  static double parseDouble(String s) // = s's value, as a double.
class Boolean:
  static boolean parseBoolean(String s) // = s's value, as a boolean.
class String:
  String substring(int i, int j) // = the letters between i (inclusive) and j (non-inclusive).
  String substring(int i) // = the letters from i (inclusive) to the end.
  int indexOf(String s) // = the index of s in this String; -1 if s is not a substring.
  int indexOf(String s, int i) // = index of s in this String after index i; -1 if s not found.
  int length() // = the number of letters in this String.
class Math:
  static double abs(double x) // absolute value of x
class JOptionPane:
  static String showInputDialog(String m) // get input from the user, prompting with m.
```

Question 1. [10 MARKS]

Below is (very stripped down) java code for class `DayPlan` and `Agenda2`. In the space provided below each `System.out.println` statement in `Agenda2`, write the output you expect to see.

```
import java.util.Date;
/**
 * DayPlan models the behaviour of a
 * single day's plan or schedule.
 */
public class DayPlan {
  private int numEvents= 0;
  private String eventList= "";
  private Date date;
  private static int numDayPlans= 0;

  public DayPlan() {
    date= new Date();
    numDayPlans= numDayPlans + 1;
  }

  public static int getNumDayPlans() {
    return numDayPlans;
  }

  public void addEvent(String event) {
    eventList= eventList + event + "\n";
    numEvents= numEvents + 1;
  }

  public int getNumEvents() {
    return numEvents;
  }
}
```

```
import java.util.Date;
/** Agenda2 test drives a DayPlan
 */
public class Agenda2 {
  public static void testDayPlan() {
    int i= 3, j= i;
    i= 4;
    System.out.println("i: " + i + " j: " + j);

    i: 4 j: 3

    int k= 5;
    System.out.println("i == k: " + (i == k));

    i == k: false

    DayPlan dp1= new DayPlan();
    DayPlan dp2= dp1;
    System.out.println("Number of day plans: " +
                       DayPlan.getNumDayPlans());

    Number of day plans: 1

    dp1= new DayPlan();
    DayPlan dp3= dp2;
    dp1.addEvent("wake up.");
    dp1.addEvent("brush teeth.");
    dp2.addEvent("sleep.");
    System.out.println("dp1 has: " + dp1.getNumEvents() +
                       " events, " + "dp2 has: " +
                       dp2.getNumEvents() + " events, " +
                       "dp3 has: " + dp3.getNumEvents() +
                       " events.");

    dp1 has: 2 events, dp2 has: 1 events, dp3 has: 1 events

    dp3= new DayPlan();
    dp2= new DayPlan();
    System.out.println("Number of day plans: " +
                       DayPlan.getNumDayPlans());

    Number of day plans: 4

    System.out.println("dp2 == dp3: " + (dp2 == dp3));

    dp2 == dp3: false
  }
}
```

Question 2. [10 MARKS]

Fill in the missing body for method `setDateAndGender` of the class `License` below. Each `License` has a 17-character identifier which starts (at position 0) with an upper-case letter indicating the initial of the license-holder's surname. The last two digits of the identifier indicate the day of the month the license-holder was born. Digits at positions 10 and 12 indicate the year the license holder was born. If the license holder is male, the digits at positions 13 and 14 indicate the month of birth, otherwise these two digits indicate the month of birth **plus 50**. For example, a male license holder with date-of-birth 75/08/13 might have identifier "X1234-67897-50813", whereas a female with the same date-of-birth would have "X1234-67897-55813".

```
/**
 * a class to store and manipulate a driver's License. */
public class License {

  /**
   * License identifier, set to ridiculous default value. */
  private String identifier= "X1234-67890-23456";

  /**
   * Encode the date-of-birth (dob) and gender into identifier.  You may assume that
   * dob is an 8-character String with the format "yy/mm/dd" (not including the
   * quotes) where digits "yy" indicate year, "mm" indicate month, and "dd" indicate
   * day of the month.  You may assume that gender is a 1-character String,
   * either "M" (indicating male) or "F" (indicating female). */
  public void setDateAndGender(String dob, String gender) {

    identifier= identifier.substring(0,15) + dob.substring(6);
    identifier= identifier.substring(0,10) + dob.substring(0,1) + "-" +
      dob.substring(1,2) + identifier.substring(13);
    if (gender.equals("F")) {
      int m= Integer.parseInt(dob.substring(3,5));
      m = m + 50;
      identifier= identifier.substring(0,13) + m +
        identifier.substring(15);
    }
    else {
      identifier= identifier.substring(0,13) +
        dob.substring(3,5) + identifier.substring(15);
    }

  }

}
```

Question 3. [10 MARKS]

Fill in the body of `goZinta`, an instance method of class `MoreMath`. The idea of `goZinta` is to perform elementary-school division: "i goes into j q times with remainder r," and return a `String` description.

```
import javax.swing.*;
/**
 * MoreMath does a little more math.
 */
public class MoreMath {

  /**
   * goZinta prompts the user for two positive integers, int1 and int2.  If int1 is bigger
   * than int2, return "int1 doesn't go into int2 at all." (of course, replacing int1
   * and int2 by the user-supplied values).  Otherwise, if int1 divides int2 exactly
   * q times, return "int1 goes into int2 exactly q times" (replacing q by its value).
   * Otherwise return "int1 goes into int2 q times with remainder r" (replacing r by
   * the remainder after division). */
  public String goZinta() {
    int int1= Integer.parseInt(JOptionPane.showInputDialog("Enter first integer: "));
    int int2= Integer.parseInt(JOptionPane.showInputDialog("Enter second integer: "));
    if (int1 > int2) {
      return "" + int1 + " doesn't go into " + int2 + " at all.";
    }
    else if (int2 % int1 == 0) {
      return "" + int1 + " goes into " + int2 + " exactly " +
        int2/int1 + " times.";
    }
    else {
      return "" + int1 + "  goes into " + int2 + " " + int2/int1 +
        " times with remainder " + int2 % int1;
    }

  }

}
```

Total Marks = 30