Problem 1

```
import stanford.karel.*;
public class KareltheRobot extends SuperKarel {
	public void run(){
		for(int i = 0; i < 4; i++){
			while (frontIsClear()) {
				putBeeper();
				move();
			}
		turnCorner();
		}
	}
private void turnCorner(){
	turnAround();
	move();
	turnRight();
}

}
```

Problem 2

(2a)

- 5.0 / 4 – 4/5 = 1.25 – 0 = 1.25
- 7 < 9 -5 && 3 % 0 == 3
  (7 < 4 : false) && (3 % 0 == 3 : true)
  ∴ false
- “B” + 8 + 4
  B84

(2b)

1. Mystery (2, 6)
   num1 = Unknown(2, 6)
2. num3 = 2 + 6 = 8
   num2 = 6 + 8 * 2 = 22
   Unknown(2, 6) = 22 =num1
3. num2 = Unknown(6, 22)
4. Unknown(6,22)
   num3 = 28
   num2 = 22 + 28*2 = 22 + 56 = 78
   return(num2)
   ∴Mystery(2,6) = 78

1. Mystery(3, 5)
   num1 = Unknown(3,5)
2. num3 = 3 + 5 =8

num2 = 5 + 16 = 21
Unknown(3, 5) = 21 = num1

3. num2 = Unknown(5, 21)
4. num3 = 5 + 21 = 26
   num2 = 21 + 26*2 = 21 + 52 =73
   return(num2)
   ∴Mystery(3,5) = 73
   Answer :
   The 1st number is: 78
   The 2nd number is: 73

Problem3 : Simple Java programs

```
import acm.program.*;

public class SecondLargest extends ConsoleProgram {
	public void run() {
		int largest = 0;
		int secondLargest = 0;
		int value = 0;

		while (true){
			value = readInt("? ");

			if (value ==0) {
				break;
			} else if (value >= largest) {
				secondLargest = largest;
				largest = value;
			}
		}
		println("The largest value is " + largest);
		println("The second largest value is" + secondLargest);

	}
}
```

Problem 4

```
import java.awt.event.MouseEvent;

import acm.program.*;
import acm.graphics.*;
public class FrogHop extends GraphicsProgram{
	public static final int SQSIZE = 75;
	public static final int NCOLS = 7;
	public static final int NROWS = 3;
	public static final int APPLICATION_WIDTH = SQSIZE * NCOLS;
	public static final int APPLICATION_HEIGHT = SQSIZE * NROWS;
	private static GImage frog = null;
	private static int currentPositionX = SQSIZE*(NCOLS-1)/2;
	private static int currentPositionY = SQSIZE*(NROWS-1);
	public void run() {
```

```
            frog = new GImage("frog.gif");
            add(frog, currentPositionX, currentPositionY);
            addMouseListeners();
      }

      //place where the mouse stands at a point(x1,y1) and frog stands at a
point(x2,y2)
      //|x1-x2|>|y1-y2|, then the mouse moves right or left,
      //|x1-x2|<|y1-y2|, then the mouse move up or down
      public void mousePressed(MouseEvent e) {
            double x = e.getX();
            double y = e.getY();

            if (Math.abs(x-currentPositionX) > Math.abs(y-currentPositionY)){
                  if (x-currentPositionX < 0){
                        if (currentPositionX - SQSIZE >= 0) {
                              frog.move(-SQSIZE, 0);
                              currentPositionX -= SQSIZE;
                        }
                  } else if (x-currentPositionX > 0){
                        if (currentPositionX + SQSIZE < SQSIZE * NCOLS) {
                              frog.move(SQSIZE, 0);
                              currentPositionX += SQSIZE;
                        }
                  }
            }else {
                  if (y-currentPositionY < 0){
                        if (currentPositionY - SQSIZE >= 0) {
                              frog.move(0, -SQSIZE);
                              currentPositionY -= SQSIZE;
                        }
                  } else if (y-currentPositionY > 0){
                        if (currentPositionY + SQSIZE < SQSIZE * NROWS) {
                              frog.move(0, SQSIZE );
                              currentPositionY += SQSIZE;
                        }
                  }
            }
      }




}
```

Problem 5

```
      public String removeDoubledLetters(String str){
            String tempStr = "";
            tempStr += str.charAt(0);
            for(int i=1; i<str.length(); i++){
                  if(str.charAt(i) != str.charAt(i-1)){
                  tempStr += str.charAt(i);
                  }
            }
```

```
        return tempStr;
    }
```