Problem 1: [16 points]

a) The book says you’re not supposed to compare two objects (Strings, for example), with the \texttt{==} operator. Why not? How should you compare them instead?

b) I gave you a rule of thumb: If a variable \textit{can} be declared as a local variable, it \textit{should} be made a local rather than an instance variable. Why is that the case?
public int mystery(Die d) {
    int a = 1;
    int b = 0;
    while(a != b) {
        a = b;
        b = d.roll();
        System.out.println("Rolled a "+b);
    }
    return b;
}

Problem 2: [20 points]

a) In English, describe what the mysterious method above does. (Try to focus on what it does rather than a step-by-step description of how it works.) To get in the right frame of mind, think about what you would write as a comment for the method.

b) Will mystery get stuck in an infinite loop if the Die instance only has 1 side? Explain.
import java.util.ArrayList;
import java.util.Random;
public class Election {
    private String[] candidates; // Array of candidate names
    private int[] votes; // Num votes for each candidate

    ... // Methods deleted to save space
}

Problem 3: [20 points]

The code above shows the start of the Election class we used in lab. Below, define a new method, voteRandomly, that could be added to the class. It should select one of the candidates in the election at random and increase their vote count by one. For full credit, your method should be capable of selecting any of the candidates in the election. (I’ve added the import statement for the Random class for you, above.)
Problem 4: [20 points]

Below, define a **toString** method that could also be added to the *Election* class. It should return a String containing the names of all of the candidates in the election. (It doesn’t need to include their vote totals.) For full credit, the string should contain \n characters such that each candidate’s name would appear on its own line if the returned String were printed.
Problem 5: [24 points]

Below, define a `getCandidatesWhoGot` method that could also be added to the `Election` class and used to find the names of candidates receiving a specific vote count. It should take a single parameter, the specified number of votes, and create and return an `ArrayList` containing the names of all candidates receiving that number of votes. (If none of the candidates receive the specified number, return an empty list.)