1. Here’s my code:

```java
class Profile {
    public static void main(String[] args) {
        String input = args[0];
        String memory = "";
        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);
            if (contains(memory, c) == 0) {
                System.out.println( c + " occurs " + contains(input, c) + " times " );
                memory += input.charAt(i);
            }
        }
    }

    static int contains(String input, char c) {
        int count = 0;
        for (int i = 0; i < input.length(); i++)
            if (input.charAt(i) == c)
                count += 1;
        return count;
    }
}
```

```
-bash-3.2$ javac Profile.java
-bash-3.2$ java Profile banana
b occurs 1 times
a occurs 3 times
n occurs 2 times
-bash-3.2$
```
2. Here’s my code:

```
class Second {
    static void simplify(int a, int b) {
        int m = Math.min(Math.abs(a), Math.abs(b));
        System.out.print(a + " / " + b + " = ");
        for (int i = 2; i < m; i++)
            while (a % i == 0 && b % i == 0) {
                a /= i;
                b /= i;
            }
        System.out.println(a + " / " + b);
    }
    public static void main(String[] args) {
        Second.simplify(6, 9);
        Second.simplify(-4, 6);
    }
}
```

```
-javac Second.java
-java Second
6/9 = 2/3
-4/6 = -2/3
```
3. Here’s my code:

```bash
cat Fraction.java
```
System.out.println(" Sub: " + f + " - " + g + " = " + f.minus(g));
System.out.println(" Mul: " + f + " * " + g + " = " + f.times(g));
System.out.println(" Div: " + f + " / " + g + " = " + f.divideBy(g));
System.out.println("Test of predicates: ");
System.out.println(" 1. Does " + f + " equal " + g + "? ");
System.out.println(" 2. Is " + h + " an integer? ");
System.out.println(" 3. Does " + i + " equal " + j + "? ");
System.out.println(" 4. Is 5/8 greater than 2/3? The answer is: ");
System.out.println((new Fraction(5, 8)).greaterThan(new Fraction(2, 3)));

}
4.1 Here’s my code:

```bash
$ cat Jugador\textsuperscript{1}.java

```class Jugador extends Player {
    Jugador(String name) {
        super(name);
    }
    boolean isWinner() {
        return score() % 7 == 0;
    }
}
```

```bash
$ cat Juego\textsuperscript{2}.java

```import java.util.*;

class Juego {
    public static void main(String[] args) {
        ArrayList<Player> players = new ArrayList<Player>();
        int n = Integer.parseInt(args[0]);
        for (int i = 0; i < n; i++)
            players.add(new Jugador("Player " + i));
        Game a = new Game(players);
        a.start();
    }
}
```

```bash
$ javac Juego.java
$ java Juego 5
Player 0: [2] false
Player 1: [2] false
Player 2: [6] false
Player 3: [1] false
Player 4: [3] false
Player 0: [2, 1] false
Player 1: [2, 5] true
$ javac Juego.java
$ java Juego 5
Player 0: [6] false
Player 1: [1] false
Player 2: [5] false
Player 3: [3] false
Player 4: [1] false
Player 0: [6, 2] false
Player 1: [2, 1] false
Player 2: [5, 2] true
$ javac Juego.java
$ java Juego 3
Player 0: [6] false
Player 1: [1] false
Player 2: [2] false
Player 0: [6, 4] false
Player 1: [1, 3] false
Player 2: [2, 4] false
Player 0: [6, 4, 3] false
Player 1: [1, 3, 4] false
Player 2: [2, 4, 5] false
```

\textsuperscript{1} Jugador means Player in Spanish.
\textsuperscript{2} Juego means Game in Spanish
4.2 Here’s my code:

```java
import java.util.*;

class Jatek extends Game {
    Jatek(ArrayList<Player> players) {
        super(players);
    }
    void start() {
        boolean gameOn = true;
        while (gameOn) {
            for (Player p : players) {
                p.add(d.shake());
                if (p.isWinner())
                    gameOn = false;
                System.out.println(p);
            }
        }
    }
}

public static void main(String[] args) {
    ArrayList<Player> players = new ArrayList<Player>();
    int n = Integer.parseInt(args[0]);
    for (int i = 0; i < n; i++)
        players.add(new Player("Player " + i));
    Game a = new Jatek(players);
    a.start();
}
```

Jatek means Game in Hungarian.
Player 1: [2, 3, 3] false
Player 2: [6, 6, 4] false
Player 0: [5, 3, 1, 4] false
Player 1: [2, 3, 3, 5] false
Player 2: [6, 6, 4, 2] false
Player 0: [5, 3, 1, 4, 2] false
Player 1: [2, 3, 3, 5, 5] false
Player 2: [6, 6, 4, 2, 2] false
Player 0: [5, 3, 1, 4, 2, 3] false
Player 1: [2, 3, 3, 5, 5, 6] true
Player 2: [6, 6, 4, 2, 2, 4] true

-bash-3.2$