

Homework 2

1. Divisors

```
public class Divisors {  
    public static void main (String[] args) {  
        int y = Integer.parseInt(args[0]);  
        int x = 1 ;  
        while ( x <= y ){  
            if ( y % x == 0 )  
                {System.out.println(x);}  
                x++ ;}  
        }  
    }  
}
```

2. Reversing a string

```
public class Reverse {  
    public static void main(String[] args) {  
  
        String a = args[0];  
        String r = "";  
  
        for (int i = a.length() - 1; i >= 0; i--) {  
            r += a.charAt(i);  
        }  
  
        System.out.println(r);  
  
        if (a.length() > 0) {  
            char middleCharacter = a.charAt(a.length() / 2);  
            System.out.println("The middle character is " + middleCharacter);  
        }  
    }  
}
```

3. Lucky streak

```
public class InOrder {  
    public static void main(String[] args) {  
        int n = (int) (Math.random() * 10);  
        System.out.print(n);  
        while (true) {  
            int n2 = (int) (Math.random() * 10);  
  
            if (n2 >= n) {  
                System.out.print(" " + n2);  
  
                n = n2 ;  
            } else {  
                break;  
            }  
        }  
        System.out.println();  
    }  
}
```

4. Perfect Numbers

```
public class Perfect {
    public static void main (String[] args) {
        int x = Integer.parseInt(args[0]);
        int y = 0;
        String Print = "";
        for (int i = 1; i <= (x-1); i++) {
            if ((x % i) == 0) {
                y = y + i;
                if (i != 1) {
                    Print = Print + " + " + i;
                } else {
                    Print = "" + i;
                }
            }
        }
        Boolean n = true;
        n = (y == x) ? true : false;
        if (n==true) {
            System.out.println(x + " is a perfect number since " + x + " = " +
Print);
        } else {
            System.out.println(x + " is not a perfect number");
        }
    }
}
```

5. Damka Board

```
public class DamkaBoard {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if ((i + j) % 2 == 0) {
                    System.out.print("* ");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

6. One of Each

```
public class OneOfEach {
    public static void main(String[] args) {
        boolean hasBoy = false;
        boolean hasGirl = false;
        int childrenCount = 0;

        // Keep simulating until there is at least one boy and one girl
        while (!hasBoy || !hasGirl) {
            if (Math.random() < 0.5) {
                System.out.print("b ");
                hasBoy = true;
            } else {
                System.out.print("g ");
                hasGirl = true;
            }
            childrenCount++;
        }

        System.out.println("\nYou made it... and you now have " + childrenCount + "
children.");
    }
}
```

7. One of Each Stats

```
public class OneOfEachStats1 {
    public static void main(String[] args) {
        int T = Integer.parseInt(args[0]);
        int totalChildren = 0;
        int twoChildren = 0;
        int threeChildren = 0;
        int fourOrMoreChildren = 0;
        int mode = 0;
        int modeCount = 0;

        for (int i = 0; i < T; i++) {
            boolean hasBoy = false;
            boolean hasGirl = false;
            int childrenCount = 0;

            while (!hasBoy || !hasGirl) {
                if (Math.random() < 0.5) {
                    hasBoy = true;
                } else {
                    hasGirl = true;
                }
                childrenCount++;
            }

            totalChildren += childrenCount;

            if (childrenCount == 2) {
                twoChildren++;
                if (twoChildren > modeCount) {
                    mode = 2;
                    modeCount = twoChildren;
                }
            } else if (childrenCount == 3) {
                threeChildren++;
                if (threeChildren > modeCount) {
                    mode = 3;
                    modeCount = threeChildren;
                }
            } else if (childrenCount >= 4) {
                fourOrMoreChildren++;
                if (fourOrMoreChildren > modeCount) {
                    mode = 4;
                    modeCount = fourOrMoreChildren;
                }
            }
        }
    }
}
```

```
double average = (double) totalChildren / T;

System.out.println("Average: " + average + " children to get at least one of each
gender.");
System.out.println("Number of families with 2 children: " + twoChildren);
System.out.println("Number of families with 3 children: " + threeChildren);
System.out.println("Number of families with 4 or more children: " +
fourOrMoreChildren);
System.out.println("The most common number of children is: " + (mode == 4 ?
"4 or more" : mode));
}
}
```


8. One of Each Stats

```
import java.util.Random;
```

```
/**
 * Computes some statistics about families in which the parents decide
 * to have children until they have at least one child of each gender.
 * The program expects to get two command-line arguments: an int value
 * that determines how many families to simulate, and an int value
 * that serves as the seed of the random numbers generated by the program.
 * Example usage: % java OneOfEachStats 1000 1
 */
public class OneOfEachStats {
    public static void main (String[] args) {
        // Gets the two command-line arguments
        int T = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);

        // Initializes a random numbers generator with the given seed value
        Random generator = new Random(seed);

        int totalChildren = 0;
        int twoChildren = 0;
        int threeChildren = 0;
        int fourOrMoreChildren = 0;
        int mode = 0;
        int modeCount = 0;

        for (int i = 0; i < T; i++) {
            boolean hasBoy = false;
            boolean hasGirl = false;
            int childrenCount = 0;

            while (!hasBoy || !hasGirl) {
                // This statement will generate a random value in the range [0,1),
                // just like you had in the previous version, except that the
                // randomization will be based on the given seed.
                double rnd = generator.nextDouble();

                if (rnd < 0.5) {
                    hasBoy = true;
                } else {
                    hasGirl = true;
                }
                childrenCount++;
            }

            totalChildren += childrenCount;

            if (childrenCount == 2) {
```

```

        twoChildren++;
        if (twoChildren > modeCount) {
            mode = 2;
            modeCount = twoChildren;
        }
    } else if (childrenCount == 3) {
        threeChildren++;
        if (threeChildren > modeCount) {
            mode = 3;
            modeCount = threeChildren;
        }
    } else if (childrenCount >= 4) {
        fourOrMoreChildren++;
        if (fourOrMoreChildren > modeCount) {
            mode = 4;
            modeCount = fourOrMoreChildren;
        }
    }
}

```

```

double average = (double) totalChildren / T;

```

```

    System.out.println("Average: " + average + " children to get at least one of each
gender.");

```

```

    System.out.println("Number of families with 2 children: " + twoChildren);
    System.out.println("Number of families with 3 children: " + threeChildren);
    System.out.println("Number of families with 4 or more children: " +
fourOrMoreChildren);
    System.out.println("The most common number of children is: " + (mode == 4 ?
"4 or more" : mode));
}
}

```