Divisors

```
public class Divisors {
    public static void main(String[] args) {
        //// Put your code here

        // Receiving number
        int x = Integer.parseInt(args[0]);

        // Determining the smallest divisor
        int d = 1;

        // Checking if the divisor is less than or equal to the number received
        while (d <= x) {

            // Checking if d is a divisor of x
            if (x % d == 0) {

                // Prints the divisor number
                System.out.println(d);
            }

            // Increases the divisor number by 1
            d += 1;
        }

    }
}
```

Reverse

```
public class Reverse {
    public static void main(String[] args) {
        //// Put your code here

        // Receiving string
        String str = args[0];

        // Prints the string received in reverse order
        for (int i = str.length(); i > 0; i--) {
            System.out.print(str.charAt(i - 1));
        }

        // Prints the middle character
        System.out.println("");
        System.out.println("The middle character is " + str.charAt((str.length() - 1) / 2));

    }
}
```

InOrder

```
public class InOrder {
    public static void main(String[] args) {
        //// Write your code here

        // Rolls a number between 0 (inclusive) to 10 (not inclusive)
        int x = (int) (10 * Math.random());

        // Determination of variable
        int y = 0;

        do {
            // Prints thr number that rolled
            System.out.print(x + " ");

            // Changes the last number to the new last number
            y = x;

            // Rolls a number between 0 (inclusive) to 10 (not inclusive)
            x = (int) (10 * Math.random());
        }
        // Checks if the rolled number is less than or equal to the last number
printed
        while (x >= y);
    }
}
```

Perfect

```
public class Perfect {
    public static void main(String[] args) {
        //// Put your code here

        // Receiving number
        int N = Integer.parseInt(args[0]);

        // A string that say that the number N is a perfect number
        String str = N + " is a perfect number since " + N + " = 1";

        // Determining the smallest divisor (not including 1)
        int d = 2;

        // Sum of the divisor
        int sum = 1;

        // Checking if the divisor is less than the number received
        while (d < N) {

            // Checking if d is a divisor of x
            if (N % d == 0) {

                // Adds the divisor to the string
                str += " + " + d;

                // Adds the divisor to the sum of the divisors
                sum += d;
            }
            // Increases the divisor number by 1
            d += 1;
        }

        // Checks if N is a perfect number and prints the answer
        if (sum == N) {
            System.out.println(str);
        } else {
            System.out.println(N + " is not a perfect number");
        }
    }
}
```

DamkaBoard

```
public class DamkaBoard {
    public static void main(String[] args) {
        //// Put your code here

        // Receiving number
        int n = Integer.parseInt(args[0]);

        // Set the starting line number
        int line = 0;

        // Checks that the line is not bigger than the received number
        while (n > line) {

            // Checks if is a even line
            if (line % 2 == 0) {

                // Prints n times *
                for (int i = 0; i < n; i++) {

                    System.out.print("* ");
                }
            } else {

                // Prints n times *
                for (int i = 0; i < n; i++) {

                    System.out.print(" *");
                }
            }
            // Skip to the next line
            System.out.println();

            line++;
        }
    }
}
```

OneOfEachStats

```
import java.util.Random;

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

        // Determination of variables
        int familiesWith2Children = 0;
        int familiesWith3Children = 0;
        int familiesWith4OrMoreChildren = 0;
        double sumOfAllTheChildren = 0;

        for (int t = 0; t < T; t++) {
            boolean isBoy = false;
            boolean isGirl = false;
            double birth = generator.nextDouble();
            int sumOfChildren = 1;

            // Checks if it is a girl or a boy
            if (birth < 0.5) {
                isGirl = true;

                // Continues to give birth until a boy is received
                while (!isBoy) {
                    birth = generator.nextDouble();
                    if (birth >= 0.5) {
                        isBoy = true;
                    }
                    sumOfChildren++;
                }
            } else {
                isBoy = true;

                // Continues to give birth until a girl is received
                while (!isGirl) {
                    birth = generator.nextDouble();
                    if (birth < 0.5) {
                        isGirl = true;
                    }
                    sumOfChildren++;
                }
            }

            // Adds the number of the children in the current family to the counter
that
            // counts the total number of all children
            sumOfAllTheChildren += sumOfChildren;

            // Adds 1 to the group that counts the number of times that number of
children
            // has appeared
            if (sumOfChildren == 2) {
                familiesWith2Children++;
            } else if (sumOfChildren == 3) {
                familiesWith3Children++;
            } else {
                familiesWith4OrMoreChildren++;
            }
```

```
    }

    // Prints the average to get at least one children of each gender.
    System.out.println(
        "Average: " + sumOfAllTheChildren / (double) T + " children to get at least one of each gender.");
    System.out.println("Number of families with 2 children: " + familiesWith2Children);
    System.out.println("Number of families with 3 children: " + familiesWith3Children);
    System.out.println("Number of families with 4 or more children: " + familiesWith4OrMoreChildren);

    // Checks and prints the most common numder of children
    int max = familiesWith2Children;
    if (max < familiesWith3Children) {
      max = familiesWith3Children;
    }
    if (max < familiesWith4OrMoreChildren) {
      max = familiesWith4OrMoreChildren;
      System.out.println("The most common number of children is 4 or more.");
    } else if (max == familiesWith2Children) {
      System.out.println("The most common number of children is 2.");
    } else {
      System.out.println("The most common number of children is 3.");
    }
  }
}
```