

Divisors:

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

Reverse:

```
public class Reverse {  
    public static void main(String[] args) {  
        String s = args[0];  
        String reverse = "";  
  
        for (int i = 0; i < s.length(); i++) {  
            char ch = s.charAt(i);  
            reverse = ch + reverse;  
        }  
        int midIndex = s.length() / 2;  
        char midChar = reverse.charAt(midIndex);  
        System.out.println(reverse);  
        System.out.println("The middle character is " + midChar);  
    }  
}
```

In Order:

```
public class InOrder {  
    public static void main(String[] args) {  
        int c = 0;  
        String result = "";  
        while (true) {  
            int randomNumber = (int) (Math.random() * 10);  
            if (randomNumber > c) {  
                result = result + randomNumber;  
                c = randomNumber;  
  
            } else if (randomNumber <= c) {  
                break;  
            }  
        }  
        System.out.println(result);  
    }  
}
```

Perfect:

```
public class Perfect {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int sum = 1;
        String result = n + " is a perfect number since " + n + " = " +
1;
        for (int i = 1; i <= n; i++) {
            if (n % i == 0 && i != 1 && n != i) {
                result = result + " + " + i;
                sum += i;
            }
        }

        if (sum == n) {
            System.out.println(result);
        } else {
            System.out.println(n + " is not a perfect number");
        }
    }
}
```

DamkaBoard:

```
public class DamkaBoard {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int c = 1;
        for (int i = 1; i <= n; i++) {

            if ((c % 2) == 0) {
                for (int j = 1; j <= n; j++) {
                    System.out.print(" *");
                }
            } else {
                for (int j = 1; j <= n; j++) {
                    System.out.print("* ");
                }
            }
            System.out.println();
            c++;
        }
    }
}
```

OnOfEachStats:

```
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]);
        // Initializes a random numbers generator with the given seed
value
        Random generator = new Random(seed);
        int twoChild = 0;
        int threeChild = 0;
        int fourChildOrMore = 0;
        int totalChildren = 0;

        for (int i = 1; i <= T; i++) {
            int boyCounter = 0;
            int girlCounter = 0;
            int total = 0;

            while (true) {
                if (boyCounter > 0 && girlCounter > 0) {
                    break;
                }
                double rnd = generator.nextDouble();
                if (rnd > 0.5) {
                    total++;
                    girlCounter++;
                } else {
                    total++;
                    boyCounter++;
                }
            }
            totalChildren += total;
            if (total == 2) {
                twoChild++;
            } else if (total == 3) {
                threeChild++;
            } else if (total >= 4) {
                fourChildOrMore++;
            }
        }
    }
}
```

```
        double average = (double) totalChildren / T;
        int common = Math.max(twoChild, Math.max(threeChild,
fourChildOrMore));

        System.out.println("Average: " + average + " children to get at
least one of each gender.");
        System.out.println("Number of families with 2 children: " +
twoChild);
        System.out.println("Number of families with 3 children: " +
threeChild);
        System.out.println("Number of families with 4 or more children:
" + fourChildOrMore);

        String mostCommonChildren = "";
        if (common == fourChildOrMore) {
            mostCommonChildren += "4 or more.";
        } else if (common == threeChild) {
            mostCommonChildren += "3.";
        } else if (common == twoChild) {
            mostCommonChildren += "2.";
        }
        System.out.println("The most common number of children is " +
mostCommonChildren);
    }
}
```