Divisors

```java
public class Divisors {
    public static void main (String[] args) {
        int x = Integer.parseInt(args[0]);
        if (x == 0){ // If the input is zero, print zero.
            System.out.println(0);
        }else{
            for(int k = 1; k <= (x / 2); k++){//Check if x has divisors from 1
to x:2.
                if(x % k == 0){
                    System.out.println(k);// Print the divisor.
                }
            }
            System.out.println(x); //  Every number is a divisor of himself.
        }
    }
}
```

Reverse

```
public class Reverse {
    public static void main (String[] args){
        String str = args[0];
        String strRev = "";// The new reversed string.
        char midChar = 'a';//The middle char.
        for(int i = 0;i < str.length();i++){
            strRev = strRev + str.charAt(str.length() - (i+1));//add the chars
in reversed to the new string.
            if(str.length() / 2 == i){//Find the mid char and set it.
                midChar = strRev.charAt(i);
            }
        }
        System.out.println(strRev);
        System.out.println("The middle character is " + midChar);
    }
}
```

InOrder

```
public class InOrder {
    public static void main (String[] args) {
        double currNum = 0;
        double nextNum = Math.random() * 10; //Set the first number.
        do{
            currNum = nextNum;
            System.out.print((int)currNum+ " ");
            nextNum = Math.random() * 10;//Generate a new number.
        }while(currNum <= nextNum); //Check if we have a non decreasing
sequence.
    }
}
```

Perfect

```
public class Perfect {
    public static void main (String[] args) {
        int ifPerfect = Integer.parseInt(args[0]);
        String str = ifPerfect + " is a perfect number since " + ifPerfect + "
= 1";
        int sum = 1;
        for(int i = 2;i <= ifPerfect / 2;i++){
            if(ifPerfect % i == 0){
                str =str + " + " + i;
                sum = sum + i;
            }
        }if(sum == ifPerfect){
            System.out.print(str);
        }else{
            System.out.print(ifPerfect + " is not a perfect number");
        }

    }
}
```

DamkaBoard

```java
public class DamkaBoard {
    public static void main(String[] args) {
        int damkaSize = Integer.parseInt(args[0]);
        String damkaLine = "*";
        for(int i = 0;i < damkaSize - 1; i++){//create a base to each damka
line.
            damkaLine = damkaLine + " *";
        }
        for(int i = 0;i < damkaSize;i++){ // print rows.
            if(i%2 == 1){ //if the row number is uneven print it with space at
the begging.
                System.out.println(" " + damkaLine);
            }else{
                System.out.println(damkaLine + " ");
            }
        }
    }
}
```

OneOfEachStats

```
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]);
        Random generator = new Random(seed);
        double kids =0;//0 for a boy, 1 for a girl.
        int twoChild = 0; // count how many families has two childs.
        int threeChild = 0; // count how many families has three childs.
        int fourChild = 0; // count how many families has four childs.
        double avgChild = 0;// avarage of how many children to get at least one
of each gender.
        int mostCommon = 0;// determines the most common number of children in
families.
        for(int i = 0;i < T;i++){
            boolean boy = false;
            boolean girl = false;
            String family="";
            while(!boy || !girl){
                kids = generator.nextDouble(); //0 is a boy and 1 is a girl.
                if(kids < 0.5){
                family = family + "b";
                boy = true;
                }
                if(kids >= 0.5){
                    family = family + "g";
                    girl = true;
                }
                if(!boy || !girl){
                    family = family + " ";
                }
            }
            int familySize = ((family.length() / 2) + 1); // determines how
many children are.
            avgChild = avgChild + familySize;
            if(familySize == 2){
                twoChild += 1;
            }
            else if(familySize == 3){
                threeChild += 1;
            }
            else{
                fourChild += 1;
            }
        }
        avgChild = avgChild / T;
        mostCommon = Math.max(Math.max(twoChild,threeChild),fourChild);// check
which group is the biggest.
        System.out.println("Average: " + avgChild + " 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: " +
fourChild);
        if(mostCommon == twoChild){
            System.out.println("The most common number of children is 2.");
        }else if(mostCommon == threeChild){
            System.out.println("The most common number of children is 3.");
        }else{
            System.out.println("The most common number of children is 4 or
more.");
        }
        }
    }
```