

Divisors.java

```
public class Divisors {  
    public static void main (String[] args) {  
        int num = Integer.parseInt(args[0]);  
        System.out.println(1);  
        for(int i = 2; i <= num ; i++){  
            if((num%i) == 0){  
                System.out.println(i);  
            }  
        }  
    }  
}
```

Reverse.java

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

InOrder.java

```
public class InOrder {  
    public static void main (String[] args) {  
        boolean inOrder = true;  
        int prev = -1;  
        while (inOrder){  
            int current = (int) (10*Math.random());  
            if (current >= prev){  
                System.out.print(current + " ");  
                prev = current;  
            }else {  
                inOrder=false;  
            }  
        }  
    }  
}
```

DamkaBoard.java

```
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%2) == 0){
                    System.out.print(" * ");
                } else if(j==0) {
                    System.out.print(" * ");
                } else {
                    System.out.print(" * ");
                }
            }
            System.out.println();
        }
    }
}
```

Perfect.java

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

OneOfEachStats.java

```
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);
        int TotalChild = 0;
        int TwoChild = 0;
        int ThreeChild = 0;
        int FourOrMoReChild = 0;
        for (int i=0 ; i<T; i++) {
            boolean boy = false;
            boolean girl = false;
            int numOfChild = 0;
            while (!(boy && girl)) {
                if (Math.random() < 0.5) {
                    boy = true;
                } else {
                    girl = true;
                }
                numOfChild++;
            }
            TotalChild += numOfChild;
            if (numOfChild == 2) {
                TwoChild++;
            } else if (numOfChild == 3) {
                ThreeChild++;
            } else {
                FourOrMoReChild++;
            }
        }
        double Average = (double) TotalChild/T;
        String max = "2 children";
        if (ThreeChild >= TwoChild && ThreeChild >= FourOrMoReChild){
            max = "3 children";
        } else if (FourOrMoReChild >= ThreeChild && FourOrMoReChild >= TwoChild)
        {
            max = "4 or more children";
        }
        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: " +
FourOrMoReChild);
        System.out.println("The most common number of children is " + max + ".");
    }
}
```