Homework 2

Divisors

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

            }

        }
    }
}
```

Reversing a string

```
public class Reverse {
     public static void main (String[] args){
         String input = args[0];
         System.out.println();
         for (int i = input.length() - 1; i >= 0; i--) {
             System.out.print(input.charAt(i));
         }
         System.out.println();

                 int middle = (input.length()-1)/2;

                  char mid = input.charAt(middle);

                  System.out.println("The middle
character is " + mid);
                 System.out.println(" ");

     }
}
```

Lucky streak

```
public class InOrder {
     public static void main (String[] args) {
     int  nam1 = (int)(Math.random()*10);
     int  nam2 = (int)(Math.random()*10);
     int temp =  nam1;
     System.out.print(nam1);

     while(temp <=  nam2){
          System.out.print(" " +  nam2);
          temp =  nam2;
           nam2 = (int)(Math.random()*10);

     }


     }
}
```

Perfect Numbers

```
public class Perfect {
    public static void main (String[] args) {
    int nam = Integer.parseInt(args[0]);
    int sam = 0;
        for(int i = 1; i < nam ; i++){
            if (nam % i== 0){
                sam = i + sam ;


            }

        }
        if (sam == nam) {
            System.out.print(nam +  " " +"is a perfect
number since " + nam +" = 1"  );
                    for(int i = 2; i <= (nam/2) ; i++){
                        if (nam % i== 0){
                            System.out.print(" + " + i );

                        }

                    }
                }
                else
            System.out.println(nam + " is not a perfect
number");
            }

        }
```

Damka Board

```
public class DamkaBoard {
    public static void main(String[] args) {
        int x = Integer.parseInt(args[0]);
        for (int i = 0; i < x; i++) {
            for (int j = 0; j < x*2; j++) {
                System.out.print((i+j)%2 == 0 ? "*":" ");
            }
            System.out.println();
        }
     }
}
```

One of Each

```
public class OneOfEach {
     public static void main (String[] args) {
        boolean boyborn = false;
        boolean girlborn = false;
        int Children = 0;

        while (!(boyborn && girlborn)) {
            if (Math.random() <= 0.5) {
                System.out.print("b"+" ");
                boyborn = true;
            } else {
                System.out.print("g ");
                girlborn = true;
            }
            Children++;
        }
        System.out.println();
        System.out.println("You made it... and you now have
" + Children + " children.");
    }
}
```

One of Each Stats1

```
public class OneOfEachStats1 {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Please provide a single
integer as a command-line argument.");
            return;
        }

        int T = Integer.parseInt(args[0]);

        double totalChildren = 0;
        int familiesWith2Children = 0;
        int familiesWith3Children = 0;
        int familiesWith4OrMoreChildren = 0;
        int mostCommonNumber = 0;
        int maxCount = 0;

        for (int i = 0; i < T; i++) {
            boolean boyBorn = false;
            boolean girlBorn = false;
            int Children = 0;

            while (!(boyBorn && girlBorn)) {
                if (Math.random() < 0.5) {
                    boyBorn = true;
                } else {
                    girlBorn = true;
                }
                Children++;
            }

            totalChildren += Children;

            if (Children == 2) {
```

```
                familiesWith2Children++;
            } else if (Children == 3) {
                familiesWith3Children++;
            } else if (Children >= 4) {
                familiesWith4OrMoreChildren++;
            }

            if (Children > maxCount) {
                mostCommonNumber = Children;
                maxCount = Children;
            }
        }
        double averageChildren = totalChildren / T;

        System.out.println("Average: " + averageChildren +
" 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);
        System.out.println("The most common number of
children is " + mostCommonNumber + ".");
    }
}
```

One of Each Stats

```
import java.util.Random;
public class OneOfEachStats {
    public static void main (String[] args) {
        int T = Integer.parseInt(args[0]);
        int times = Integer.parseInt(args[1]);
        Random generator = new Random(times);
        double rnd;
```

```
        double totalChildren = 0;
        int familiesWith2Children = 0;
        int familiesWith3Children = 0;
        int familiesWith4OrMoreChildren = 0;
        String mostCommonNumber="";

        for (int i = 0; i < T; i++) {
            boolean boyBorn = false;
            boolean girlBorn = false;
            int Children = 0;
            while (!(boyBorn && girlBorn)) {
                rnd = generator.nextDouble();
                if (rnd > 0.5) {
                    boyBorn = true;
                } else {
                    girlBorn = true;
                }
                Children++;
            }
            totalChildren += Children;
            if (Children == 2) {
                familiesWith2Children++;
            } else if (Children == 3) {
                familiesWith3Children++;
            } else if (Children >= 4) {
                familiesWith4OrMoreChildren++;
            }

            if(familiesWith2Children >
familiesWith3Children){
                mostCommonNumber = "2";
            } else if (familiesWith3Children >
familiesWith4OrMoreChildren){
                mostCommonNumber = "3";
```

```
            } else{
                mostCommonNumber = "4 or more";
            }
        }
        double averageChildren = totalChildren / T;

        System.out.println("Average: " +
averageChildren + " 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);
        System.out.println("The most common number of
children is " + mostCommonNumber + ".");


    }
}
```