Divisors.java

```java
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.java

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

InOrder.java

```
public class InOrder {
    public static void main (String[] args) {
    int FirstTry = (int)(Math.random() * (10));
    int Attempts = 0;
    System.out.print(+FirstTry+" ");
    do{
        Attempts = (int)(Math.random() * (10));
        if(Attempts>=FirstTry){
            System.out.print(+Attempts+" ");     // A loop enters only when the second is
                                                 .          larger than the first
            FirstTry = Attempts;
        }
    }while(FirstTry <= Attempts);

    }
}
```

DamkaBoard.java

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

Perfect.java

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

    }
}
```

OneOfEachStats.java

```
public class OneOfEachStats {
   public static void main (String[] args) {
      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);
      String EndOfsentence = ".";
      String ans= "";
      int ConterFor2 = 0;
      int ConterFor3 = 0;
      int ConterFor4 = 0;
      int colcoltion = 0;
      char boy = 'b';
      char girl = 'g';
      int r = 0;
      for (; r < T; ++r) {
      ans="";
      boolean helper = true;
      while (helper){
            double rand = generator.nextDouble();
            ans+= (rand<0.5) ? boy : girl;
                  colcoltion++;
            for (int i = 0; i < ans.length()-1; i++) {
               if(ans.length()>1){
                  if (ans.charAt(i) != ans.charAt(i+1)){
                        helper=false;
                  }
               }
            }
      }
         if(ans.length()==2 )
               ConterFor2++;
         if(ans.length()==3 )       ///Loops that help associate
              ConterFor3++;         // with the relevant family quantity
         if(ans.length()>=4)
              ConterFor4++;
     }
 double Average=((double)colcoltion/(double)T);
 int chack = Math.max(ConterFor3, ConterFor2);
 int max = Math.max(chack,ConterFor4);



 if(max==ConterFor4 ){
```

```
        EndOfsentence = " or more.";
            max = 4;
        }else if
        (max==ConterFor2)
            max = 2;
        else
            max = 3;
        if(r==T){
                System.out.println("Average: "+Average+" children to get at least one of
each gender."+"\n"+
                        "Number of families with 2 children: "+ConterFor2+"\n"+
                        "Number of families with 3 children: "+ConterFor3+"\n"+
                        "Number of families with 4 or more children: "+ConterFor4+"\n"+
                        "The most common number of children is "+max+EndOfsentence);
        }
    }
    }
```