1. Divisors

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

2. Reversing a string- Reverse

```
public class Reverse {
    public static void main (String[] args){
        String word=(args[0]);
        String reversed= "";
        int length = word.length();

for (int n = length - 1; n >= 0; n--) {
    reversed += word.charAt(n);
}

System.out.println(reversed);
int middleindex = (length -1) / 2;
char middlechar = word.charAt(middleindex);
System.out.println("The middle character is "+ middlechar);

    }
}
```

3. Lucky streak- InOrder

```
public class InOrder {
    public static void main (String[] args) {
        int i=0;
        int prevNum= (int) (Math.random()*10);
        while (i<10) {
            System.out.print(i+ " ");
            i++;
        if (prevNum < i) {
            break;
        }
        }
    }
}
```

4. Perfect Numbers- Perfect

```
public class Perfect {
    public static void main (String[] args) {
        int N= Integer.parseInt(args[0]);
        int SumDiv=0;
        StringBuilder divisorString = new StringBuilder();

        for (int divisor= 1; divisor <=N / 2; divisor++)
      if (N%divisor==0){
       if (divisor != N) {
          if (divisorString.length() > 0) {
      divisorString.append(" + ");

         }
     }
     SumDiv += divisor;
     divisorString.append(divisor);
     }

     if (SumDiv==N) {
      System.out.println(N +" is a perfect number since " + N +" = " + divisorString);
     }
     else {
      System.out.println(N + " is not a perfect number");
    }
}
}
```

5. Damka Board

```
public class DamkaBoard {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int i = 0;
        int j;

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

6. One Of Each

```
public class OneOfEach {
         public static void main (String[] args) {
           int boycount= 0;
           int girlcount=0;
           int totalchildren=0;

     do {
           if (Math.random() < 0.5) {
              System.out.print("b ");
              boycount++;
               totalchildren++;
       }
                   else if (Math.random() > 0.5) {
                     System.out.print("g ");
                     girlcount++;
                     totalchildren++;
       }
               } while (boycount < 1 || girlcount < 1);

         System.out.println("\nYou made it... and you now have  "+ totalchildren + " children.");
         }
}
```

7. One of each stats- OneOfEachStats1

```
public class OneOfEachStats1 {
  public static void main(String[] args) {
    int T = Integer.parseInt(args[0]);
    int totalChildren = 0;
    int famof2 = 0;
    int famof3 = 0;
    int famof4ormore = 0;

    int[] familyCounts = new int[T];

    for (int i = 0; i < T; i++) {
      int boycount = 0;
      int girlcount = 0;
      int totalchildren = 0;

      do {
        if (Math.random() < 0.5) {
          boycount++;
          totalchildren++;
        } else if (Math.random() > 0.5) {
          girlcount++;
          totalchildren++;
        }
      } while (boycount < 1 | | girlcount < 1);

      familyCounts[i] = totalchildren;
      totalChildren += totalchildren;
```

```
            if (totalchildren == 2) {
                famof2++;
            } else if (totalchildren == 3) {
                famof3++;
            } else if (totalchildren >= 4) {
                famof4ormore++;
            }
        }

        double averageChildren = (double) 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: " + famof2);
        System.out.println("Number of families with 3 children: " + famof3);
        System.out.println("Number of families with 4 or more children: " + famof4ormore);

        int mode = calculateMode(familyCounts);
        System.out.println("The most common number of children is " + (mode >= 4 ? "4 or more" : mode) + ".");
    }

    private static int calculateMode(int[] array) {
        int maxValue = 0, maxCount = 0;

        for (int i = 0; i < array.length; ++i) {
            int count = 0;
            for (int j = 0; j < array.length; ++j) {
                if (array[j] == array[i])
                    ++count;
            }
        if (count > maxCount)
    maxCount = count;
    maxValue = array[i];
     }
  }
     return maxValue;
   }
}
```

8. One of each stats (final version)

```
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]);
                // Initailizes a random numbers generator with the given seed value
    Random generator = new Random(seed);
    int totalChildren = 0;
    int famof2 = 0;
    int famof3 = 0;
    int famof4ormore = 0;

    int[] familyCounts = new int[T];

    for (int i = 0; i < T; i++) {
      int boycount = 0;
      int girlcount = 0;
      int totalchildren = 0;

      do {
        double rnd = generator.nextDouble();
         if (rnd < 0.5) {
           boycount++;
           totalchildren++;
         } else if (rnd > 0.5) {
           girlcount++;
           totalchildren++;
         }
```

```
            } while (boycount < 1 || girlcount < 1);


            familyCounts[i] = totalchildren;

            totalChildren += totalchildren;


            if (totalchildren == 2) {

                famof2++;

            } else if (totalchildren == 3) {

                famof3++;

            } else if (totalchildren >= 4) {

                famof4ormore++;

            }

        }


        double averageChildren = (double) 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: " + famof2);

        System.out.println("Number of families with 3 children: " + famof3);

        System.out.println("Number of families with 4 or more children: " + famof4ormore);


        int mode = calculateMode(familyCounts);

        System.out.println("The most common number of children is " + (mode >= 4 ? "4 or more" : mode) + ".");

    }


    private static int calculateMode(int[] array) {

        int maxValue = 0, maxCount = 0;


        for (int i = 0; i < array.length; ++i) {

            int count = 0;
            for (int j = 0; j < array.length; ++j) {
                if (array[j] == array[i])
                    ++count;
            }

            if (count > maxCount) {
                maxCount = count;
                maxValue = array[i];
            }
        }

        return maxValue;
    }
}
```