```
public class Divisors {
      public static void main (String[] args) {

                int given_num = Integer.parseInt(args[0]);
                int i = 1;

                while (i <= given_num) {
                          if (given_num % i == 0){
                                    System.out.println(i);
                                    }
                          i ++;
                }
      }
}

```

```java
public class Reverse {
      public static void main (String[] args){

                String s = args[0];
                String reversed_s = "";
                for (int i = s.length() - 1; i >= 0; i = i - 1){
                          char charecter = s.charAt(i);
                          reversed_s = reversed_s + charecter;
                }
                System.out.println(reversed_s);

                int l = (s.length() - 1)/2;
                char m = s.charAt(l);
                System.out.println("The middle character is " + m);
      }
}

```

```java
public class InOrder {
      public static void main (String[] args) {

                int lastNum = 0;
                int maxNum = 0;

                 while (lastNum >= maxNum){
                          lastNum = (int)(Math.random() * 10);
                          if (lastNum >= maxNum){
                                    maxNum = lastNum;
                                    System.out.print(lastNum + " ");
                          }
                          }
                 }

      }

```

```java
public class Perfect {
    public static void main (String[] args) {
        int n = Integer.parseInt(args[0]);
        int sum = 0;
        String perfect = n + " is a perfect number since " + n + " = ";
        String notPerfect = n + " is not a perfect number";

        for(int i = 1; i < n; i++){
            if (n % i == 0){
                perfect = perfect + i + " + ";
                sum = sum + i;
            }
        }

        if(sum == n){
                System.out.println(perfect.substring(0, perfect.length() - 2));
            }

        else{
            System.out.println(notPerfect);
        }
    }
}

```

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

```

```java
public class OneOfEach {
      public static void main (String[] args) {

                boolean girl = true;
                boolean boy = true;
                int i = 0;

                while (boy  || girl) {
                          int child = (int)(Math.random() * 2);
                          i++;
                          if (child == 0){
                                    girl = false;
                                    System.out.print("g");
                          }
                          else{
                                    boy = false;
                                    System.out.print("b");
                          }
                          System.out.print(" ");
                }
                System.out.println();
                System.out.println("You made it... and you now have " + i + " children.");
      }
}

```

```java
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);
        //Creating variables for the total boys + girls
                    double total = 0;
                    double avg = 0.0;
                    int how_many_with_2 = 0;
                    int how_many_with_3 = 0;
                    int how_many_with_4_and_more = 0;

                    //Cerating a loop that runs T times
                    for (int i = 0; i < T; i++){
                                boolean girl = true;
                                boolean boy = true;

                                //Creating a loop that runs untill we have one of each
                                int numOfChildrenInSpecificFamily = 0;
                                while (boy  || girl) {
                                            double child = generator.nextDouble();
                                            if (child < 0.5){
                                                        girl = false;
                                            }
                                            else{
                                                        boy = false;
                                            }
                                            total++;
                                            numOfChildrenInSpecificFamily++;
                                }
                                //checks which family type was created and adds one for it's num
                                switch(numOfChildrenInSpecificFamily){
                                            case 2:
                                                        how_many_with_2++;
                                                        break;
                                            case 3:
                                                        how_many_with_3++;
                                                        break;
                                            default:
                                                        how_many_with_4_and_more++;
                                                        break;
                                }

                    }
                    //calculates the avarage of children that was genorated in the for loop
                    avg = total / T;
                    System.out.println("Average: " + avg + " children to get at least one of each gender.");
                    System.out.println("Number of families with 2 children: " + how_many_with_2);
                    System.out.println("Number of families with 3 children: " + how_many_with_3);
                    System.out.println("Number of families with 4 or more children: " + how_many_with_4_and_more);
                    //checks which family type ws created the most
                    int max_family_type = 2;
                    if((how_many_with_2 >= how_many_with_3) && (how_many_with_2 >= how_many_with_4_and_more)) {
                                max_family_type = 2;
                    }
                    else {
                                if(how_many_with_3 >= how_many_with_4_and_more){
                                            max_family_type = 3;
                                }
                                else {
                                            max_family_type = 4;
                                }
                    }
                    if (max_family_type == 2 || max_family_type == 3) {
                                System.out.println("The most common number of children is " + max_family_type + ".");
                    }
                    else {
                                System.out.println("The most common number of children is 4 or more.");
                    }
                    }
        }

```