```java
public class Divisors {
    public static void main(String[] args) {
        //number given by the user
        int userNum = Integer.parseInt(args[0]);

        //cheks the divisors of the given number and prints them
        for(int i = 1; i <= userNum; i++) {
            if(userNum % i == 0) {
                System.out.println(i);
            }
        }
    }
}
```

```java
public class Reverse {
    public static void main(String[] args) {
        //string given by the user
        String userString = args[0];
        String result = "";

        //given string in reversed order
        for(int i = 0; i < userString.length(); i++) {
            result = result + userString.charAt(userString.length() - (1 + i));
        }

        int middleCharLocation = 0;

        //finds the middle char location dependence if the string length is even or odd
        if(userString.length() % 2 == 0) {
            middleCharLocation = (userString.length() / 2) - 1;
        }
        else{
            middleCharLocation = (userString.length() / 2);
        }

        char middleChar = userString.charAt(middleCharLocation);
        System.out.println(result);
        System.out.println("The middle character is " + middleChar);
    }
}
```

```
public class InOrder {
    public static void main(String[] args) {
        int randomNum = (int)(Math.random() * 10);
        int lastNum = 0;

        //checks if the next random number is bigger then the last random number
        do {
        System.out.println(randomNum);
        lastNum = randomNum;
        randomNum = (int)(Math.random() * 10);
        }
        while(lastNum <= randomNum);
    }
}
```

```java
public class Perfect {
    public static void main(String[] args) {
        int userNum = Integer.parseInt(args[0]);
        int sum = 0;
        String str = "";

        for(int i = 2; i < userNum; i++) {
            if(userNum % i == 0) { //cheks the divisors of the given number
                sum += i; //calculates the sum of the dividors
                String fromIntToString = Integer.toString(i);
                str = str + " + " + fromIntToString; //saves the dividors as a string
            }
        }

        //cheks if the number is perfect
        if((sum + 1) == userNum) {
            System.out.println(userNum + " is a perfect number since " + userNum + " =
            1" + str);
        }
        else {
            System.out.println(userNum + " is not a perfect number");
        }
    }
}
```

```
public class DamkaBoard {
    public static void main(String[] args) {
        int userNum = Integer.parseInt(args[0]);
        int row = 0;
        int astrisk = 0;

        //prints the pattern of the asterisks
        while(row < userNum) {
            if(row % 2 ==  0) {  //prints the row pattern of the asterisks depends if the row
           is even
                while(astrisk < userNum) {
                    System.out.print("* ");
                    astrisk++;
                }
            }
            else {  //prints the row pattern of the asterisks depends if the row is not even
                while(astrisk < userNum) {
                    System.out.print(" *");
                    astrisk++;
                }
            }
            System.out.println();
            row++;
            astrisk = 0;
        }
    }
}
```

```
public class OneOfEach {
    public static void main(String[] args) {
        String str = "";
        boolean isGirl = false;
        boolean isBoy = false;
        int howManyChildren = 0;

        //checks until there is a boy and a girl in the family
        while((isGirl && isBoy) != true) {
            double randomGender = Math.random();
            if(randomGender < 0.5) {  //the chance of getting a girl [0,0.5).
                isGirl = true;
                str += "g ";
            }
            else {  //the chance of getting a boy [0.5,1).
                isBoy = true;
                str += "b ";
            }
            howManyChildren++;
        }

        System.out.println(str);
        System.out.println("You made it... and you now have " + howManyChildren + "
          children.");
    }
}
```

```
public class OneOfEachStats1 {
    public static void main (String[] args) {
        int userNum = Integer.parseInt(args[0]);
        int experimentsNum = 0;
        int howManyChildrenOverAll = 0;
        int howManyChildrenInOneFamily = 0;
        int twoChildren = 0;
        int threeChildren = 0;
        int fourPlusChildren = 0;
        boolean isGirl = false;
        boolean isBoy = false;

        //simulates different cases of families until they have a boy and a girl
        while(experimentsNum < userNum) {
            while((isGirl && isBoy) != true) {  //checks until there is a boy and a girl in
            the family
                double randomGender = Math.random();
                if(randomGender < 0.5) {  //the chance of getting a girl [0,0.5).
                    isGirl = true;
                }
                else {  //the chance of getting a boy [0.5,1).
                    isBoy = true;
                }
                howManyChildrenOverAll++;
                howManyChildrenInOneFamily++;
            }
            if(howManyChildrenInOneFamily == 2) {
                twoChildren++;
            }
            else if(howManyChildrenInOneFamily == 3) {
                threeChildren++;
            }
            else {
                fourPlusChildren++;
            }
            experimentsNum++;
            isGirl = false;
            isBoy = false;
            howManyChildrenInOneFamily = 0;
        }

        //calculates the average number of children in a family
        double averageChildren = (double)howManyChildrenOverAll /
          (double)userNum;

        System.out.println("Average: " + averageChildren + " children to get at least one
          of each gender.");
        System.out.println("Number of families with 2 children: " + twoChildren);
        System.out.println("Number of families with 3 children: " + threeChildren);
        System.out.println("Number of families with 4 or more children: " +
fourPlusChildren);

         //checks the most common number of children in a family
        if(twoChildren >= threeChildren && twoChildren >= fourPlusChildren) {
            System.out.println("The most common number of children is 2.");
        }
        else if(threeChildren > twoChildren && threeChildren >= fourPlusChildren) {
            System.out.println("The most common number of children is 3.");
        }
        else {
            System.out.println("The most common number of children is 4 or more.");
        }
    }
}
```

```
public class OneOfEachStats {
    public static void main (String[] args) {
        int userNum = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);
        int experimentsNum = 0;
        int howManyChildrenOverAll = 0;
        int howManyChildrenInOneFamily = 0;
        int twoChildren = 0;
        int threeChildren = 0;
        int fourPlusChildren = 0;
        boolean isGirl = false;
        boolean isBoy = false;

        //Initailizes a random numbers generator with the given seed value
        Random generator = new Random(seed);

         //simulates different cases of families until they have a boy and a girl
        while(experimentsNum < userNum) {
            while((isGirl && isBoy) != true) { //checks until there is a boy and a girl in the
            family
                double randomGender = generator.nextDouble();;
                if(randomGender < 0.5) {  //the chance of getting a girl [0,0.5).
                    isGirl = true;
                }
                else {  //the chance of getting a boy [0.5,1).
                    isBoy = true;
                }
                howManyChildrenOverAll++;
                howManyChildrenInOneFamily++;
            }
            if(howManyChildrenInOneFamily == 2) {
                twoChildren++;
            }
            else if(howManyChildrenInOneFamily == 3) {
                threeChildren++;
            }
            else {
                fourPlusChildren++;
            }
            experimentsNum++;
            isGirl = false;
            isBoy = false;
            howManyChildrenInOneFamily = 0;
        }

        //calculates the average number of children in a family
        double averageChildren = (double)howManyChildrenOverAll /
           (double)userNum;
        System.out.println("Average: " + averageChildren + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + twoChildren);
        System.out.println("Number of families with 3 children: " + threeChildren);
        System.out.println("Number of families with 4 or more children: " + fourPlusChildren);

         //checks the most common number of children in a family
        if(twoChildren >= threeChildren && twoChildren >= fourPlusChildren) {
            System.out.println("The most common number of children is 2.");
        }
        else if(threeChildren > twoChildren && threeChildren >= fourPlusChildren) {
            System.out.println("The most common number of children is 3.");
        }
        else {
            System.out.println("The most common number of children is 4 or more.");
        }
    }
}
```