

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

```

public class Reverse {
    public static void main (String[] args){
        // gets string from user and reverses it and then prints
        String mystring = args[0];
        String newString = "";
        for (int i =mystring.length()-1; i>=0; i--){
            newString = newString+mystring.charAt(i);
        }
        System.out.println(newString);
        int middleIndex = mystring.length()/2;
        char middleChar;
        if (mystring.length() % 2 == 0) {
            // For even-length strings, get the previous character of the center
            middleChar = mystring.charAt(middleIndex - 1);
        } else {
            // For odd-length strings, get the exact center character
            middleChar = mystring.charAt(middleIndex);
        }
        System.out.println("The middle character is " + middleChar);
    }
}

```

```
public class InOrder {  
    public static void main (String[] args) {  
        int randomNumber = ((int)((Math.random()*10)));  
        System.out.println(randomNumber);  
        int newRandomNumber = 0;  
        do {  
            newRandomNumber = ((int)((Math.random()*10)));  
            if (newRandomNumber >= randomNumber) {  
                System.out.println(newRandomNumber);  
                randomNumber = newRandomNumber;  
            }  
            else break;  
        } while (randomNumber >= newRandomNumber);  
    }  
}
```

```
public class DamkaBoard {  
    public static void main(String[] args) {  
        int input = Integer.parseInt(args[0]);  
        for (int i = 1; i<=input; i++){  
            for(int j=1; j<=input && i%2!=0 ; j++){  
                System.out.print("* ");  
            }  
            for (int h=1; h<=input && i%2==0; h++){  
                System.out.print(" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

```

public class Perfect {
    public static void main (String[] args) {
        int number = Integer.parseInt(args[0]);
        String initial = number + " is a perfect number since " + number + " = 1";
        int count = 1;
        for (int i = 2; i<number && number!=1; i++){
            if (number%i==0) {
                count+=i;
                initial += " + " + i;
            }
        }
        if (count==number && number!=1) {
            System.out.println(initial);
        }
        else System.out.println(number+ " is not a perfect number");
    }
}

```

```

public class OneOfEachStats {

    public static void main (String[] args) {
        int families = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);
        Random generator = new Random(seed);
        double average = 0;
        String commonKids = "";
        int countTogether = 0;
        int countTwo = 0;
        int countThree = 0;
        int countFourOrMore = 0;
        for (int j = 0; j < families; j++) {
            // Initalizes a random numbers generator with the given seed value
            boolean girl = false;
            boolean boy = false;
            int countEachFamily = 0;
            for (int i=1; !(girl&&boy); i++){
                double random = generator.nextDouble();
                if (random<0.5) {
                    boy = true;
                }
                else {
                    girl= true;
                }
                countEachFamily+=1;
            }
            if (countEachFamily==2) {
                countTwo+=1;
            } else if (countEachFamily==3) {
                countThree+=1;
            } else {countFourOrMore+=1;}
            countTogether+=countEachFamily;
        }
        average = (double)countTogether/families;
        double roundedAverage = Math.round(average * 1000.0) / 1000.0;
        Integer mostKidsCheck =
Math.max(countTwo,Math.max(countThree,countFourOrMore));

        if (mostKidsCheck.equals(countTwo)) {
            commonKids = "2";
        }
        else if (mostKidsCheck.equals(countThree)) {
            commonKids = "3";
        }
        else{ commonKids = "4 or more";}
    }
}

```

```
        System.out.println("Average: "+ roundedAverage + " children to get at least one of  
each gender.");  
        System.out.println("Number of families with 2 children: " + countTwo);  
        System.out.println("Number of families with 3 children: " + countThree);  
        System.out.println("Number of families with 4 or more children: " +  
countFourOrMore);  
        System.out.println("The most common number of children is " + commonKids +  
".");  
    }}
```