

```
public class Reverse {  
    public static void main (String[] args){  
        String s = args[0];  
        int l = s.length();  
        for (int i = l - 1; 0 <= i ; i--) { // i marks place in the string  
            System.out.print(s.charAt(i));  
        }  
        System.out.println();  
        int mid = (l - 1) / 2; // for even lengthed Strings - mid will be the first out of the 2 centers  
        System.out.println("The middle character is " + s.charAt(mid));  
    }  
}
```

```
public class Perfect {
    public static void main (String[] args) {
        int n = Integer.parseInt(args[0]);
        int sum = 1;
        //we will be skipping the number 1 in the loop (1 is not perfect and is a divisor for every number)
        String strPerfect = n + " is a perfect number since " + n + " = 1";
        for (int i = 2; i <= n/2 && sum <= n; i++) {
            //check all numbers up to i/2, with option to stop if sum is greater than n - for optimization
            if (n % i == 0) {
                sum += i;
                strPerfect = strPerfect + " + " + i;
            }
        }
        if (sum != n) {
            System.out.println(n + " is not a perfect number");
        } else {
            System.out.println(strPerfect);
        }
    }
}
```

```

public class OneOfEachStats1 {
    public static void main (String[] args) {
        int times = Integer.parseInt(args[0]);
        int totalChildren = 0, countFam2 = 0, countFam3 = 0, countFamMore = 0;
        for (int i = 1; i <= times; i++) { //checks for each family and records the numbers
            boolean bothGenders = false;
            int firstGen = (int) (Math.random()*2); // 0-boy 1-girl
            int kids = 1;
            while (!bothGenders) {
                int gen = (int) (Math.random()*2); // 0-boy 1-girl
                kids++;
                bothGenders = (firstGen != gen);
            }
            switch (kids) {
                case 2 :
                    countFam2++;
                    break;
                case 3 :
                    countFam3++;
                    break;
                default :
                    countFamMore++;
                    break;
            }
            totalChildren += kids;
        }
        double average = (double) totalChildren / times;
        System.out.println("Average: " + average + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + countFam2);
        System.out.println("Number of families with 3 children: " + countFam3);
        System.out.println("Number of families with 4 or more children: " + countFamMore);
        String s = "";
        int mostCommon = Math.max(Math.max(countFam2, countFam3), countFamMore);
        if (mostCommon == countFam2) {
            s = "2.";
        } else if (mostCommon == countFam3) {
            s = "3.";
        } else {
            s = "4 or more.";
        }
        System.out.println("The most common number of children is " + s);
    }
}

```

```

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);
        int times = Integer.parseInt(args[0]);
        int totalChildren = 0, countFam2 = 0, countFam3 = 0, countFamMore = 0;
        for (int i = 1; i <= times; i++) { //checks for each family and records the numbers
            boolean bothGenders = false;
            int firstGen = (int) (generator.nextDouble()*2); // 0-boy 1-girl
            int kids = 1;
            while (!bothGenders) {
                int gen = (int) (generator.nextDouble()*2); // 0-boy 1-girl
                kids++;
                bothGenders = (firstGen != gen);
            }
            switch (kids) {
                case 2 :
                    countFam2++;
                    break;
                case 3 :
                    countFam3++;
                    break;
                default :
                    countFamMore++;
                    break;
            }
            totalChildren += kids;
        }
        double average = (double) totalChildren / times;
        System.out.println("Average: " + average + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + countFam2);
        System.out.println("Number of families with 3 children: " + countFam3);
        System.out.println("Number of families with 4 or more children: " + countFamMore);
        String s = "";
        int mostCommon = Math.max(Math.max(countFam2, countFam3), countFamMore);
        if (mostCommon == countFam2) {
            s = "2.";
        } else if (mostCommon == countFam3) {
            s = "3.";
        } else {
            s = "4 or more.";
        }
        System.out.println("The most common number of children is " + s);
    }
}

```

```
public class OneOfEach {
    public static void main (String[] args) {
        boolean bothGenders = false;
        int firstGen = (int) (Math.random()*2); // 0-boy 1-girl
        int count = 1;
        if (firstGen == 0) {
            System.out.print("b");
        } else {
            System.out.print("g");
        }
        while (!bothGenders) {
            int gen = (int) (Math.random()*2); // 0-boy 1-girl
            count++;
            bothGenders = (firstGen != gen);
            if (gen == 0) {
                System.out.print(" b");
            } else {
                System.out.print(" g");
            }
        }
        System.out.println();
        System.out.println("You made it... and you now have " + count + " children.");
    }
}
```

```
public class InOrder {
    public static void main (String[] args) {
        int n = (int) (Math.random() * 10);
        int old = 0; // so as that 'n' will always be greater than or equal to 'old' at the start.
        while (old <= n) {
            // Generates and prints random number in [0,10).
            System.out.print(n + " ");
            old = n;
            n = (int) (Math.random() * 10);
        }

        /*
        //this is code that i wrote for myself to test the Do-While Loop.
        int n = (int) (Math.random() * 10);
        int old = 0; // so as that 'n' will always be greater than or equal to 'old' at the start.
        do {
            // Generates and prints random number in [0,10).
            System.out.print(n + " ");
            old = n;
            n = (int) (Math.random() * 10);
        } while (old <= n);

        */
    }
}
```

```
public class Divisors {  
    public static void main (String[] args) {  
        int x = Integer.parseInt(args[0]);  
        for (int i = 1; i <= x/2; i++) { //check all numbers up to i/2 for efficiency  
            if (x % i == 0) {  
                System.out.println(i);  
            }  
        }  
        System.out.println(x);  
    }  
}
```

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