

## Divisors

```
/**
 * Gets a command-line argument (int), and prints all the divisors of the given number.
 */
public class Divisors {
    public static void main (String[] args) {

        int a = Integer.parseInt(args[0]) ;

        for (int i=1 ; i <= a ; i++) {
            if (a%i == 0){
                System.out.println(i) ;
            }
        }
    }
}
```

### Reverse:

```
/**
 * Prints a given string, backward. Then prints the middle character in the string.
 * The program expects to get one command-line argument: A string.
 */
public class Reverse {
    public static void main (String[] args){

        String s = args[0] ;

        for (int i = s.length()-1 ; i >= 0 ; i--){

            System.out.print(s.charAt(i)) ;
        }

        int middle = (s.length()-1)/2 ;

        System.out.println() ;
        System.out.println("The middle character is " + s.charAt(middle)) ;

    }
}
```

### In Order:

```
/**
 * Generates and prints random integers in the range [0,10),
 * as long as they form a non-decreasing sequence.
 */
public class InOrder {
    public static void main (String[] args) {

        int prevNum = -1 ;

        do{
            int currentNum = (int)(Math.random()*10) ;
            if (currentNum >= prevNum){
                System.out.print(currentNum + " " );
                prevNum = currentNum ;
            }
            else {
                break ;
            }
        }while (true) ;

    }
}
```

### Damka board:

```
/**
 * Gets a command-line argument n (int), and prints an n-by-n damka board.
 */
public class DamkaBoard {
    public static void main(String[] args) {

        int n = Integer.parseInt(args[0]) ;

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

        }

    }
}
```

### Perfect:

```
/**  
 * Gets a command-line argument (int), and checks if the given number is perfect.  
 */
```

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

### One of Each Stats;

```
import java.util.Random;
```

```
/**
```

- \* Computes some statistics about families in which the parents decide
- \* to have children until they have at least one child of each gender.
- \* The program expects to get two command-line arguments: an int value
- \* that determines how many families to simulate, and an int value
- \* that serves as the seed of the random numbers generated by the program.
- \* Example usage: % java OneOfEachStats 1000 1

```
*/
```

```
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 sumOfAllChildrens = 0 ;
```

```
        int fam2Childrens = 0 ;
```

```
        int fam3Childrens = 0 ;
```

```
        int fam4Childrens = 0 ;
```

```
        for(int i =1 ; i <= t ; i++){
```

```
            boolean girl = false ;
```

```
            boolean boy = false ;
```

```
            int numOfGirls = 0 ;
```

```
            int numOfBoys = 0 ;
```

```
            int sumOfChildrens = 0 ;
```

```
            do{
```

```
                double chance = generator.nextDouble() ;
```

```

        if (chance < 0.5){
            girl = true ;
            numOfGirls++ ;
        }else {
            boy = true ;
            numOfBoys++ ;
        }

        /*System.out.println("girls:" + numOfGirls) ; */
        /*System.out.println("boys:" + numOfBoys) ; */
        sumOfChildrens = numOfGirls + numOfBoys ;
        /*System.out.println("sum:" + sumOfChildrens) ; */

    }while ((girl && boy) != true) ;

    if(sumOfChildrens == 2){
        fam2Childrens = fam2Childrens + 1 ;
    }else if(sumOfChildrens == 3){
        fam3Childrens = fam3Childrens + 1 ;
    }else if(sumOfChildrens >= 4){
        fam4Childrens = fam4Childrens + 1 ;
    }

    sumOfAllChildrens = sumOfAllChildrens + sumOfChildrens ;
    /*System.out.println("sum all:" + sumOfAllChildrens) ; */

}

double average = (double)sumOfAllChildrens/t ;

System.out.println("Average: " + average + " children to get at least one of each gender.") ;

System.out.println("Number of families with 2 children: " + fam2Childrens ) ;

System.out.println("Number of families with 3 children: " + fam3Childrens ) ;

System.out.println("Number of families with 4 or more children: " + fam4Childrens ) ;

if(fam2Childrens >= fam3Childrens && fam2Childrens >= fam4Childrens){
    System.out.println("The most common number of children is 2.") ;
}
else if(fam3Childrens >= fam2Childrens && fam3Childrens >= fam4Childrens){

```

```

        System.out.println("The most common number of children is 3.") ;
    }else if (fam4Childrens >= fam2Childrens && fam4Childrens >= fam3Childrens){
        System.out.println("The most common number of children is 4 or more.") ;
    }

}

}

```

```

//// In the previous version of this program, you used a statement like:
//// double rnd = Math.random();
//// Where "rnd" is the variable that stores the generated random value.
//// In this version of the program, replace this statement with:
//// double rnd = generator.nextDouble();
//// This statement will generate a random value in the range [0,1),
//// just like you had in the previous version, except that the
//// randomization will be based on the given seed.
//// This is the only change that you have to do in the program.

```