

```
/**
 * 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);
            }
        }
    }
}
```

```

/**
 * 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];
        int n = s.length();
        String news = "";
        String savec = "";
        int count = 0;
        for(int i = n-1; i >= 0; i--)
        {
            count++;
            news += s.charAt(i);
        }
        System.out.println(news);
        System.out.println("The middle character is "+
news.charAt(count/2));
    }
}

```

```
/**
 * 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 num1 = 0;
        int num2 = 0;
        int stop = 0;
        while (stop == 0)
        {
            num1 = (int)((Math.random() * 10));
            if (num1 >= num2)
                System.out.println(num1);
            else stop = 1;
            num2 = num1;
        }
    }
}
```

```

/**
 * Gets a command-line argument (int), and chekcs 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)
            {
                s += " + "+i;
                sum += i;
            }
        }
        if (sum != num)
        {
            System.out.println(num+ " is not a perfect
number");
        }
        else System.out.println(s);
    }
}

```

```

/**
 * 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 num = Integer.parseInt(args[0]);
        for (int i = 0; i < num; i++)
        {
            if (i % 2 == 0)
            {
                for (int j = 0; j < num; j++)
                {
                    System.out.print("* ");
                }
            }
            else
            {
                for (int j = 0; j < num; j++)
                {
                    System.out.print(" *");
                }
            }
            System.out.println();
        }
    }
}

```

```
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
theprogram.
 * Exampleusage%javaOneOfEachStats10001*
publicclassOneOfEachStats {
    public static void main (String[] args) {
```

```
        int t = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);
        Random generator = new Random(seed);
        int countk=0;
        double avarage=0;
        boolean isgirl= true;
        boolean isboy= true;
        int twok=0;
        int threek=0;
        int morek=0;
        for (int i=0; i<t; i++)
        {
```

```
            countk=0; // reset to count number of kids in a new
family
            while(isboy==true || isgirl ==true)
            {
                double num = generator.nextDouble(); // if there
are not at least one boy and girl it gives a new random
                if (num>=0.5) // checks if its boy or girl
                {
                    isgirl=false;
                }
                else
                {
                    isboy=false;
                }
            }
            countk++; // total num of children in this
family;
        }
        isboy=true;
        isgirl=true;
        avarage+=countk; // total num of children
        if (countk==2)
```

```

        {
            twok++; // adding to the total of two
kids in a family
        }
        else if (countk==3)
        {
            threek++; // adding to the total of 3
kids in a family
        }
        else if (countk>3)
        {
            morek++; // adding to the total of 4 or
more kids in a family
        }
    }
    double total= avarage/t;
    System.out.println("Average: "+total+" children to get at
least one of each gender.");
    System.out.println("Number of families with 2 children:
"+ twok);
    System.out.println("Number of families with 3 children:
"+ threek);
    System.out.println("Number of families with 4 or more
children: "+ morek);
    if (twok>=threek && twok>= morek)
    {
        System.out.println("The most common number of children
is 2.");
    }
    else if (threek>=morek&& threek>=morek)
    {
        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.");
}
}

```