```
/**
 *  Gets a command-line argument (int), and prints all the divisors of the given number.
 */
public class Divisors
{
	public static void main (String[] args)
	{
		//Gets a number
		int x = Integer.parseInt(args[0]);
		int i = 1;
		// A loop that checks the devisors of the number from 1 to the number itself
		while (i <= x)
		{
			//If the number devided by i, print i (a divisor)
			if ((x % i) == 0)
			{
				System.out.println(i);
			}
			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 str = args[0];
		String revStr = "";
		int i = str.length() - 1;
		while (i > (-1))
		{
			revStr = revStr + str.charAt(i);
			i--;
		}
		System.out.println(revStr);
		char mid = str.charAt((str.length() - 1) / 2);
		System.out.println("The middle character is " + mid);
	}
}
```

```
/**
 *  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 r = (int)((Math.random()) * 10);
		int last = r;
		while (r >= last)
		{
			last = r;
			System.out.print(r + " ");
			r = (int)((Math.random()) * 10);
		}
	}
}
```

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

```
//// Gets a command-line argument (int), and chekcs if the given number is perfect.
public class Perfect {
	public static void main (String[] args)
	{
		int x = Integer.parseInt(args[0]);
		int i = 2;
		int sum = 1;
		String str = x + " is a perfect number since " + x + " = 1";
		while (i < x)
		{
			if (x % i == 0)
			{
				sum = sum + i;
				str = str + " + " + i;

			}
			i++;
		}
		if (sum == x)
		{
			System.out.print(str);
		}
		else
		{
			System.out.println(x + " is not a perfect number");
		}
	}
}
```

```
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)
        {
                int T = Integer.parseInt(args[0]);
                int seed = Integer.parseInt(args[1]);
                int i;
                int twoKids = 0;
                int threeKids = 0;
                int fourKids = 0;
                int sumKids = 0;
                double totallSum = 0;
                boolean isBoy = false;
                boolean isGirl = false;
                int max;
                Random generator = new Random(seed);
                double kid;
                for (i = 0; i < T; i++)
                {
                        while (!isGirl || !isBoy)
```

```
                {
                        kid = generator.nextDouble();
                        if (kid >= 0.5)
                        {
                                isGirl = true;
                        }
                        else
                        {
                                isBoy = true;
                        }
                        sumKids++;
                        totallSum++;
                }
                if (sumKids == 2)
                        twoKids++;
                if (sumKids == 3)
                        threeKids++;
                if (sumKids >= 4)
                        fourKids++;
                isBoy = false;
                isGirl = false;
                sumKids = 0;
        }
        System.out.println("Average: " + (totallSum / T) + " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + twoKids);
        System.out.println("Number of families with 3 children: " + threeKids);
        System.out.println("Number of families with 4 or more children: " + fourKids);
        max = Math.max(Math.max(twoKids, threeKids), fourKids);
        if (max == twoKids)
```

```
                System.out.println("The most common number of children is 2.");
            else
            {
                if (max == threeKids)
                    System.out.println("The most common number of children is
3.");
                else
                {
                    if (max == fourKids)
                        System.out.println("The most common number of
children is 4 or more.");
                }
            }

        }
}
```