```java
/**
 *  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 num = Integer.parseInt(args[0]);
		for (int i = 1; i <= num; i++)
		{
			if (num % i == 0) {
				System.out.println(i);
			}
		}
	}
}
```

```java
/**
 * 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 word = args[0];
        String reversed_word = "";
        for (int i = word.length()-1; i >= 0; i--)
        {
            reversed_word += word.charAt(i);
        }
        System.out.println(reversed_word);
        System.out.println("The middle character is " +
        reversed_word.charAt(word.length()/2));
    }
}
```

```java
/**
 *  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 lowest = 0;
        int randomnum = 0;
        do
        {
            randomnum = (int)(Math.random() * 10);
            if (randomnum >= lowest) {
                System.out.print(randomnum + " ");
                lowest = randomnum;
            }
        }
        while (lowest < 9);
    }
}
```

```java
/**
 *  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]);
		int sum = 0;
		String score = "";
		for (int i = 1; i <= num/2; i++)  //No need to check
numbers greater than num/2
		{
			if (num % i == 0)
			{
				sum += i;
				if (sum < num) { //Checks if i is the final
divisor of num
					score += (i + " + ");
				}
				else {
					score += (i);
				}
			}
		}
		if (sum==num) {
			System.out.print(num + " is a perfect number since "
+ num
					+ " = " + score);
			if (num == 0) //fixes the string for num = 0
				System.out.print(0);
		}
		else
			System.out.print(num + " is not a perfect number");
	}
}
```

```java
/**
 *  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 size = Integer.parseInt(args[0]);
            for (int i = 0; i <= size; i++) {
                  if (i % 2 == 0) {
                        for (int j = 1; j <= size; j++) {
                              System.out.print("* ");
                        }
                  } else {
                        for (int j = 1; j <= size; j++) {
                              System.out.print(" *");
                        }
                  }
                  if (i == size-1) {
                        break;
                  }
                  else {
                        System.out.println();
                  }
            }
      }
}
```

```java
/**
 *  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 one command-line argument: an int
value
 *   that determines how many families to simulate.
 */
public class OneOfEachStats1 {
     public static void main (String[] args) {
          boolean boy_born = false, girl_born = false;
          int iterations = Integer.parseInt(args[0]);
          int children = 0;
          int total_children = 0;
          int family_of_2 = 0;
          int family_of_3 = 0;
          int family_of_4p = 0;
          for (int i = 0; i < iterations; i++) {
               while (!(boy_born && girl_born)) {
                    if (Math.random() > 0.5) {
                         boy_born = true;
                    } else {
                         girl_born = true;
                    }
                    children++;
               }
               switch (children) {
                    case 2:
                         family_of_2++;
                         break;
                    case 3:
                         family_of_3++;
                         break;
                    default:
                         family_of_4p++;
               }
               total_children += children;
               children = 0;
               boy_born = false;
               girl_born = false;
          }
          System.out.println("Average: " +
           (double)total_children/iterations +
          " children to get at least one of each gender.");
          System.out.println("Number of families with 2 children: "
          + family_of_2);
          System.out.println("Number of families with 3 children: "
          + family_of_3);
```

```
            System.out.println("Number of families with 4 children: "
            + family_of_4p);
            System.out.print("The most common number of children is
            ");
            if (family_of_2 > family_of_3) {
                  System.out.print("2.");
            }
            else {
                  if (family_of_3 > family_of_4p) {
                        System.out.print("3.");
                  }
                  else {
                        System.out.print("4.");
                  }
            }
      }
}
```