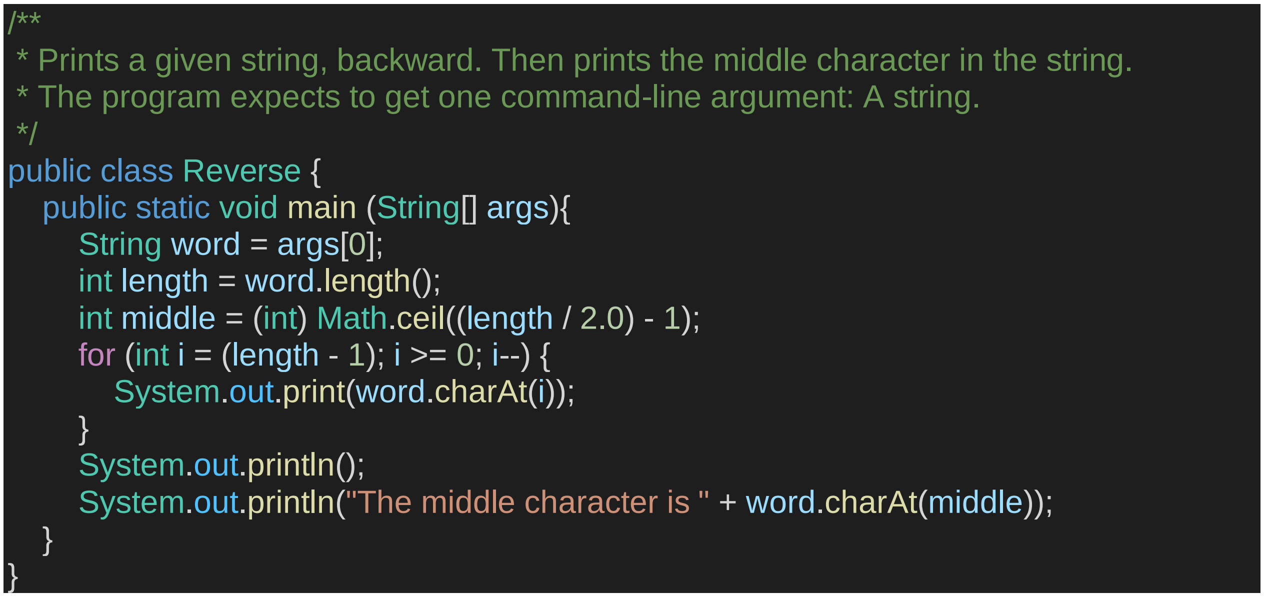
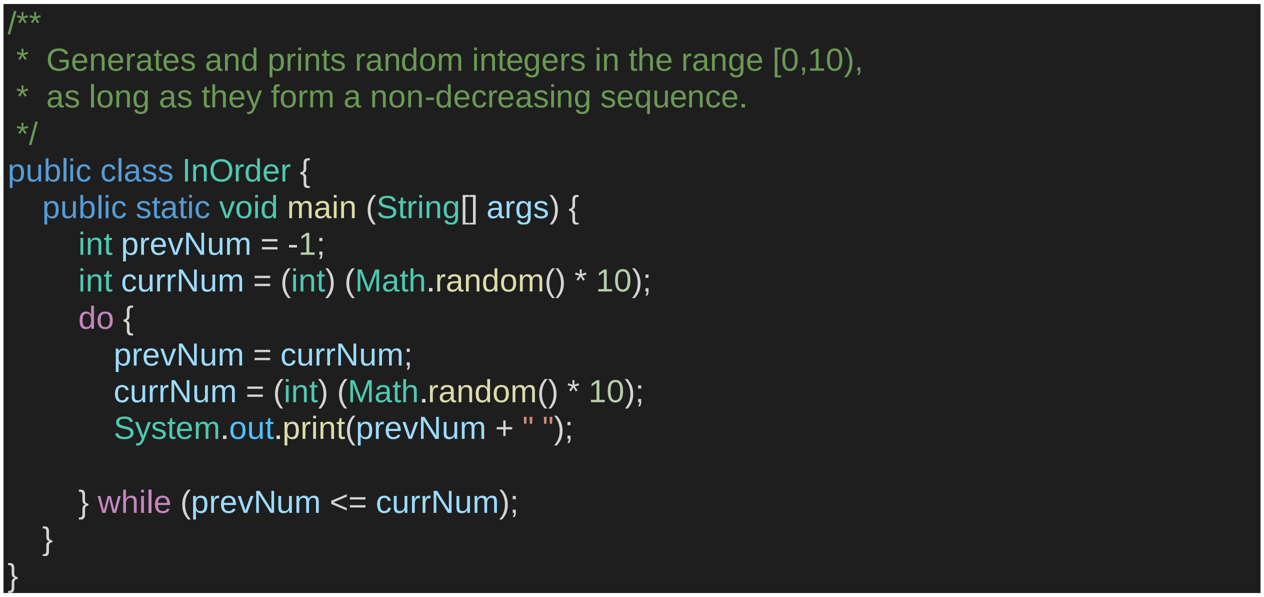
```java
/**
 *  Gets a command-line argument (int), and prints all the divisors of the given number.
 */
public class Divisors {
    public static void main (String[] args) {
        int num = Integer.parseInt(args[0]);
        for (int i = 1; i <= num; i++) {
            if ((num % i) == 0) {
                System.out.println(i);
            }
        }
    }
}
```

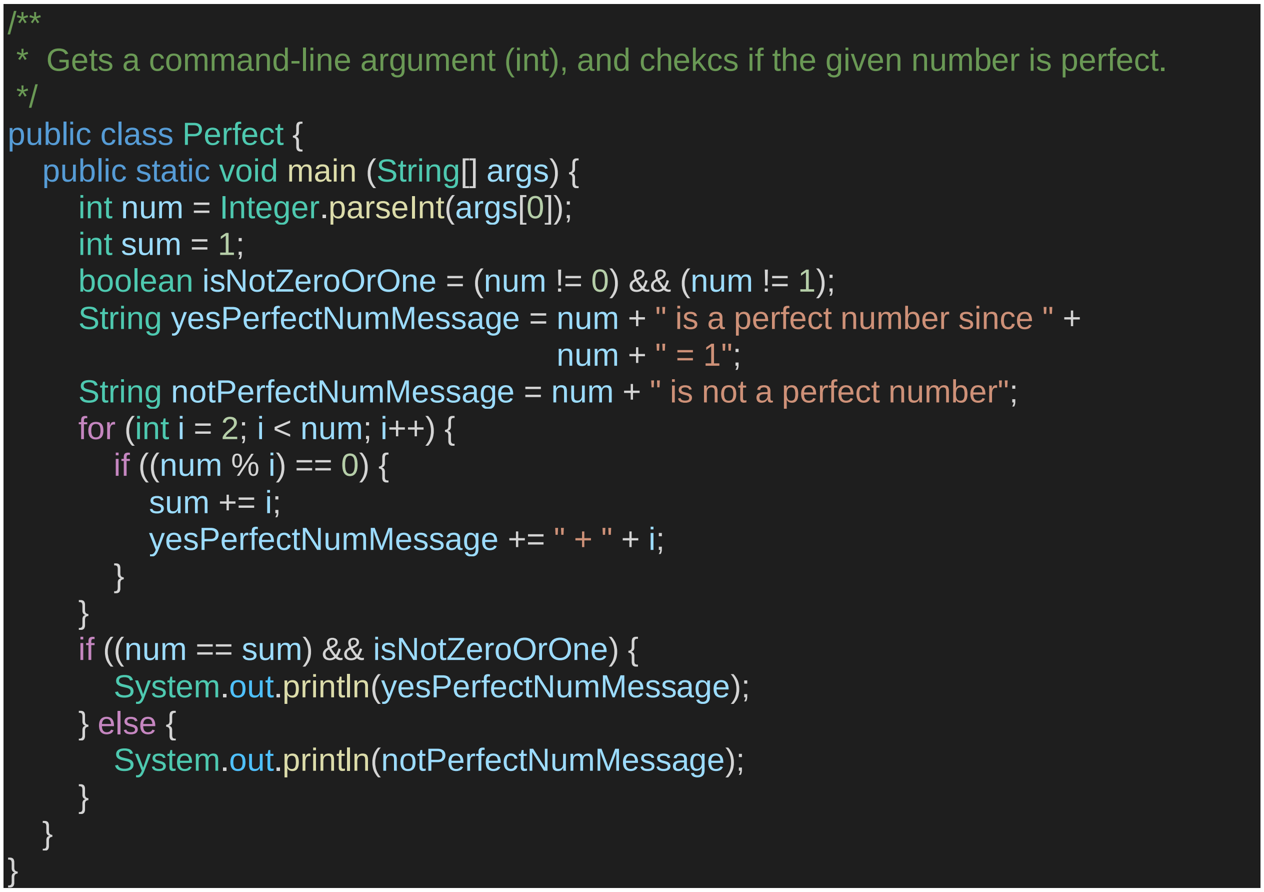


```java
/**
 * Prints a given string, backward. Then prints the middle character in the string.
 * The program expects to get one command-line argument: A string.
 */
public class Reverse {
    public static void main (String[] args){
        String word = args[0];
        int length = word.length();
        int middle = (int) Math.ceil((length / 2.0) - 1);
        for (int i = (length - 1); i >= 0; i--) {
            System.out.print(word.charAt(i));
        }
        System.out.println();
        System.out.println("The middle character is " + word.charAt(middle));
    }
}
```



```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 prevNum = -1;
        int currNum = (int) (Math.random() * 10);
        do {
            prevNum = currNum;
            currNum = (int) (Math.random() * 10);
            System.out.print(prevNum + " ");

        } while (prevNum <= currNum);
    }
}
```



```java
/**
 *  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]);
        int sum = 1;
        boolean isNotZeroOrOne = (num != 0) && (num != 1);
        String yesPerfectNumMessage = num + " is a perfect number since " +
                                      num + " = 1";
        String notPerfectNumMessage = num + " is not a perfect number";
        for (int i = 2; i < num; i++) {
            if ((num % i) == 0) {
                sum += i;
                yesPerfectNumMessage += " + " + i;
            }
        }
        if ((num == sum) && isNotZeroOrOne) {
            System.out.println(yesPerfectNumMessage);
        } else {
            System.out.println(notPerfectNumMessage);
        }
    }
}
```

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

```java
/**
 *  Simulates the formation of a family in which the parents decide
 *  to have children until they have at least one child of each gender.
 */
public class OneOfEach {
    public static void main (String[] args) {
        int numOfChildren = 0;
        double randomGender = 0;
        boolean haveBoy = false;
        boolean haveGirl = false;
        while (!(haveBoy && haveGirl)) {
            randomGender = Math.random();
            if (randomGender < 0.5) {
                haveBoy = true;
                ++numOfChildren;
                System.out.print("b ");
            } else {
                haveGirl = true;
                ++numOfChildren;
                System.out.print("g ");
            }
        }
        System.out.println();
        System.out.println("You made it... and you now have " +
                           numOfChildren + " children.");
    }
}
```

```
/**
 *  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) {
      int numOfExperiments = Integer.parseInt(args[0]);
      int twoChildren = 0;
      int threeChildren = 0;
      int fourOrMoreChildren = 0;
      double countAllChildren = 0;
      for (int i = 0; i < numOfExperiments; i++) {
         int numOfChildren = 0;
         double randomGender = 0;
         boolean haveBoy = false;
         boolean haveGirl = false;
         while (!(haveBoy && haveGirl)) {
            randomGender = Math.random();
            if (randomGender < 0.5) {
               haveBoy = true;
               ++numOfChildren;
            } else {
               haveGirl = true;
               ++numOfChildren;
            }
         }
         if (numOfChildren == 2) {
            ++twoChildren;
         } else if (numOfChildren == 3) {
            ++threeChildren;
         } else {
            ++fourOrMoreChildren;
         }
         countAllChildren += numOfChildren;
      }




      System.out.println("Average: " + (countAllChildren / numOfExperiments) +
                                  " children to get at least one of each gender.");
      System.out.println("Number of families with 2 children: " + twoChildren);
      System.out.println("Number of families with 3 children: " + threeChildren);
      System.out.println("Number of families with 4 or more children: " +
                                      fourOrMoreChildren);
        System.out.print("The most common number of children is ");
        if (Math.max(Math.max(twoChildren, threeChildren), fourOrMoreChildren) ==
                                              twoChildren) {
            System.out.println("2.");
        } else if (Math.max(Math.max(twoChildren, threeChildren), fourOrMoreChildren) ==
                                                        threeChildren) {
            System.out.println("3.");
        } else {
            System.out.println("4 or more.");
        }
    }
}
```

```java
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 numOfExperiments = Integer.parseInt(args[0]);
        int seed = Integer.parseInt(args[1]);
        Random generator = new Random(seed);
        int twoChildren = 0;
        int threeChildren = 0;
        int fourOrMoreChildren = 0;
        int max = 0;
        double countAllChildren = 0;
        for (int i = 0; i < numOfExperiments; i++) {
            int numOfChildren = 0;
            double randomGender = 0;
            boolean haveBoy = false;
            boolean haveGirl = false;
            while (!(haveBoy && haveGirl)) {
                randomGender = generator.nextDouble();
                if (randomGender < 0.5) {
                    haveBoy = true;
                    ++numOfChildren;
                } else {
                    haveGirl = true;
                    ++numOfChildren;
                }
            }

            if (numOfChildren == 2) {
                ++twoChildren;
            } else if (numOfChildren == 3) {
                ++threeChildren;
            } else {
                ++fourOrMoreChildren;
            }
            countAllChildren += numOfChildren;
        }
        System.out.println("Average: " + (countAllChildren / numOfExperiments) +
                           " children to get at least one of each gender.");
        System.out.println("Number of families with 2 children: " + twoChildren);
        System.out.println("Number of families with 3 children: " + threeChildren);
        System.out.println("Number of families with 4 or more children: " +
                                     fourOrMoreChildren);
        System.out.print("The most common number of children is ");
        max = Math.max(Math.max(twoChildren, threeChildren), fourOrMoreChildren);
        if (max == twoChildren) {
            System.out.println("2.");
        } else if (max == threeChildren) {
            System.out.println("3.");
        } else {
            System.out.println("4 or more.");
        }
    }
}
```