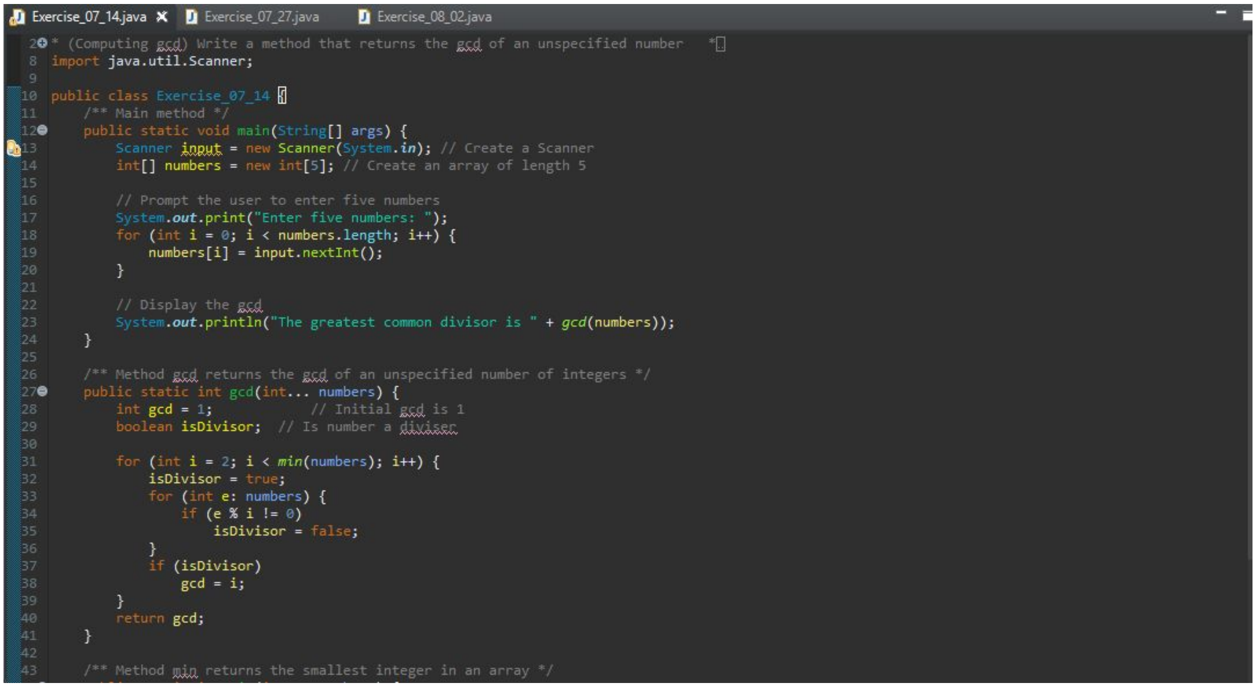
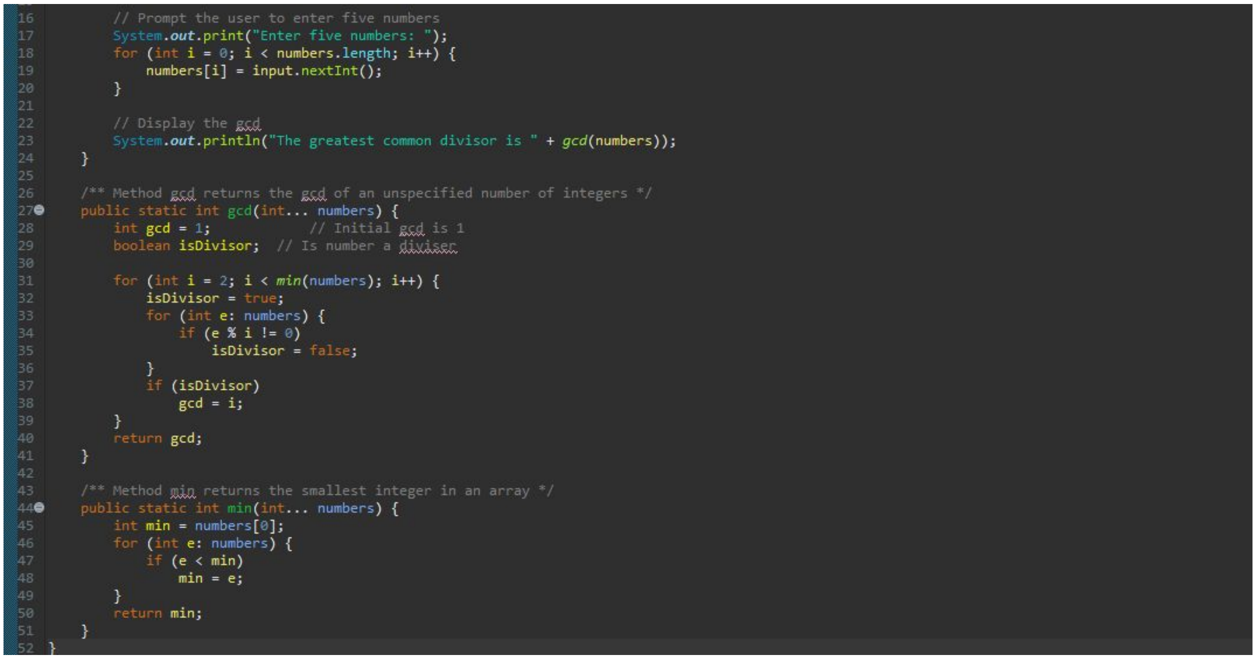
Brandon London 2261

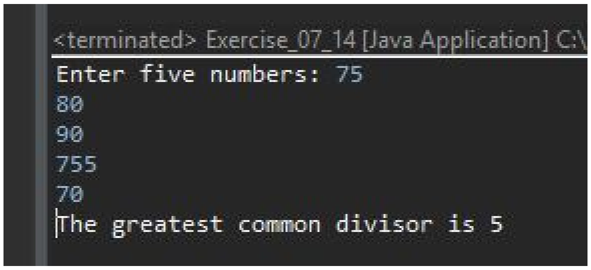


```
* (Computing gcd) Write a method that returns the gcd of an unspecified number
import java.util.Scanner;

public class Exercise_07_14 {
    /** Main method */
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in); // Create a Scanner
        int[] numbers = new int[5]; // Create an array of length 5

        // Prompt the user to enter five numbers
        System.out.print("Enter five numbers: ");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = input.nextInt();
        }

        // Display the gcd
        System.out.println("The greatest common divisor is " + gcd(numbers));
    }

    /** Method gcd returns the gcd of an unspecified number of integers */
    public static int gcd(int... numbers) {
        int gcd = 1;            // Initial gcd is 1
        boolean isDivisor;  // Is number a diviser

        for (int i = 2; i < min(numbers); i++) {
            isDivisor = true;
            for (int e: numbers) {
                if (e % i != 0)
                    isDivisor = false;
            }
            if (isDivisor)
                gcd = i;
        }
        return gcd;
    }

    /** Method min returns the smallest integer in an array */
```

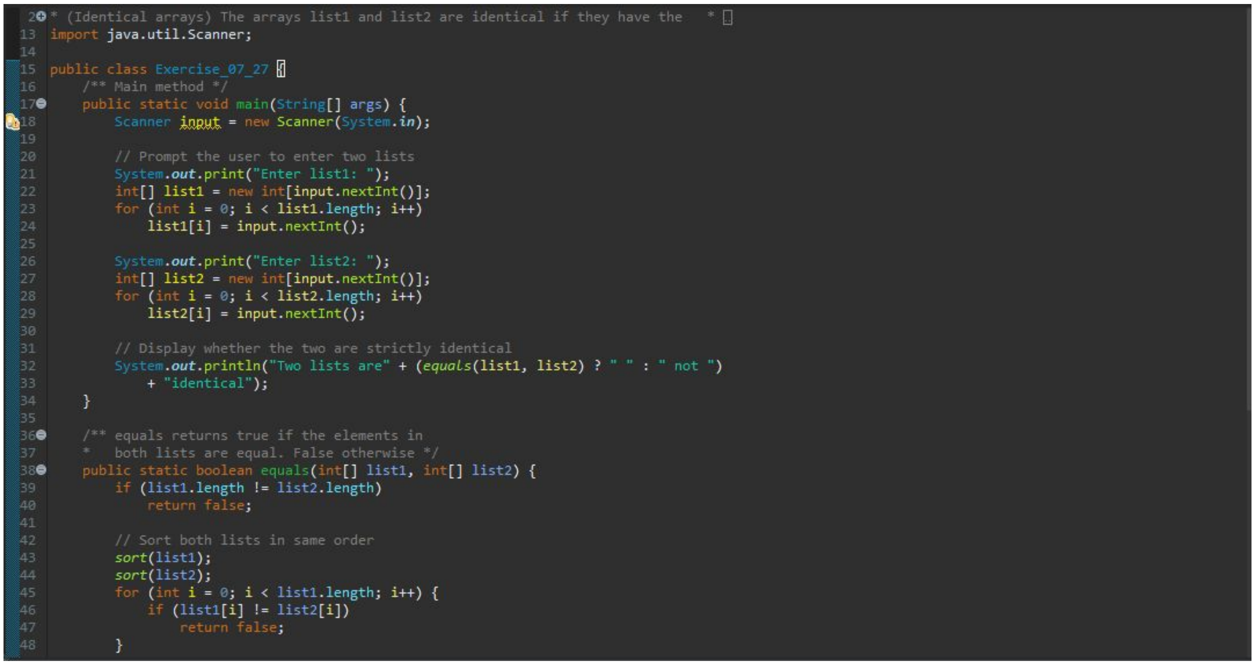


```
        // Prompt the user to enter five numbers
        System.out.print("Enter five numbers: ");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = input.nextInt();
        }

        // Display the gcd
        System.out.println("The greatest common divisor is " + gcd(numbers));
    }

    /** Method gcd returns the gcd of an unspecified number of integers */
    public static int gcd(int... numbers) {
        int gcd = 1;            // Initial gcd is 1
        boolean isDivisor;  // Is number a diviser

        for (int i = 2; i < min(numbers); i++) {
            isDivisor = true;
            for (int e: numbers) {
                if (e % i != 0)
                    isDivisor = false;
            }
            if (isDivisor)
                gcd = i;
        }
        return gcd;
    }

    /** Method min returns the smallest integer in an array */
```

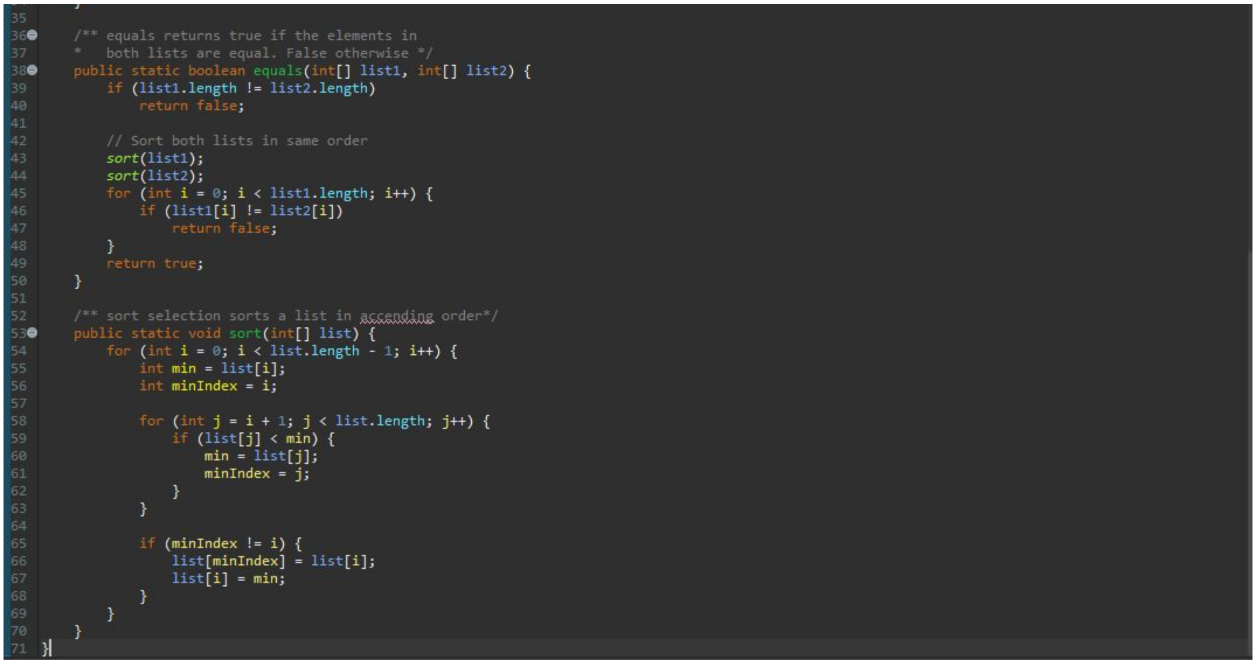
```
    public static int min(int... numbers) {
        int min = numbers[0];
        for (int e: numbers) {
            if (e < min)
                min = e;
        }
        return min;
    }
}
```



```
<terminated> Exercise_07_14 [Java Application] C:\
Enter five numbers: 75
80
90
755
70
The greatest common divisor is 5
```

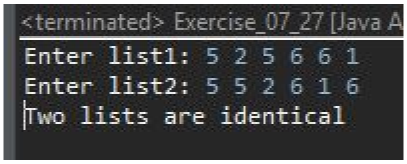


```
* (Identical arrays) The arrays list1 and list2 are identical if they have the   *
import java.util.Scanner;

public class Exercise_07_27 {
    /** Main method */
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Prompt the user to enter two lists
        System.out.print("Enter list1: ");
        int[] list1 = new int[input.nextInt()];
        for (int i = 0; i < list1.length; i++)
            list1[i] = input.nextInt();

        System.out.print("Enter list2: ");
        int[] list2 = new int[input.nextInt()];
        for (int i = 0; i < list2.length; i++)
            list2[i] = input.nextInt();

        // Display whether the two are strictly identical
        System.out.println("Two lists are" + (equals(list1, list2) ? " " : " not ")
            + "identical");
    }

    /** equals returns true if the elements in
    *   both lists are equal. False otherwise */
    public static boolean equals(int[] list1, int[] list2) {
        if (list1.length != list2.length)
            return false;

        // Sort both lists in same order
        sort(list1);
        sort(list2);
        for (int i = 0; i < list1.length; i++) {
            if (list1[i] != list2[i])
                return false;
        }
```

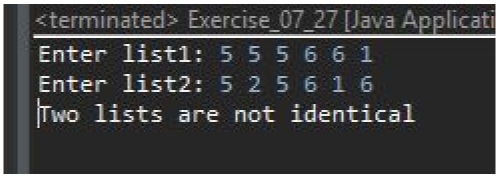


```

    /** equals returns true if the elements in
    *   both lists are equal. False otherwise */
    public static boolean equals(int[] list1, int[] list2) {
        if (list1.length != list2.length)
            return false;

        // Sort both lists in same order
        sort(list1);
        sort(list2);
        for (int i = 0; i < list1.length; i++) {
            if (list1[i] != list2[i])
                return false;
        }
```

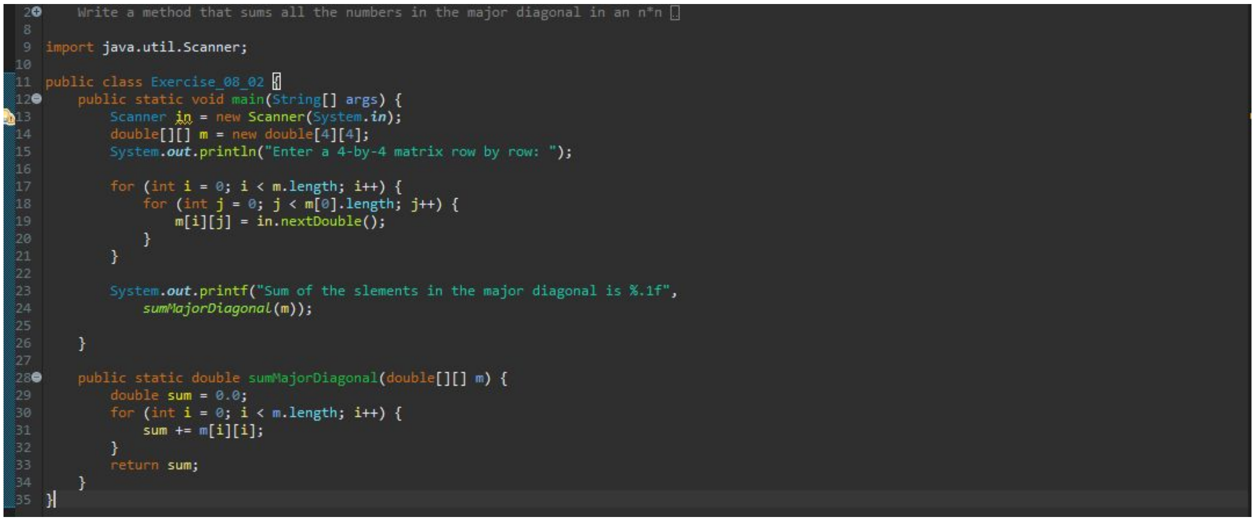
```
        return true;
    }

    /** sort selection sorts a list in accending order*/
    public static void sort(int[] list) {
        for (int i = 0; i < list.length - 1; i++) {
            int min = list[i];
            int minIndex = i;

            for (int j = i + 1; j < list.length; j++) {
                if (list[j] < min) {
                    min = list[j];
                    minIndex = j;
                }
            }

            if (minIndex != i) {
                list[minIndex] = list[i];
                list[i] = min;
            }
        }
    }
}
```



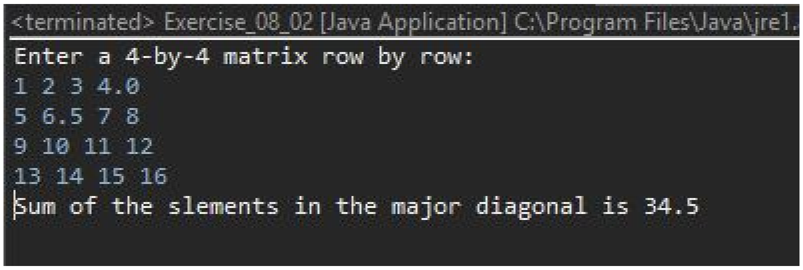
```
<terminated> Exercise_07_27 [Java A
Enter list1: 5 2 5 6 6 1
Enter list2: 5 5 2 6 1 6
Two lists are identical
```



```
<terminated> Exercise_07_27 [Java Applicati
Enter list1: 5 5 5 6 6 1
Enter list2: 5 2 5 6 1 6
Two lists are not identical
```



```java
    Write a method that sums all the numbers in the major diagonal in an n*n 

import java.util.Scanner;

public class Exercise_08_02 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        double[][] m = new double[4][4];
        System.out.println("Enter a 4-by-4 matrix row by row: ");

        for (int i = 0; i < m.length; i++) {
            for (int j = 0; j < m[0].length; j++) {
                m[i][j] = in.nextDouble();
            }
        }

        System.out.printf("Sum of the slements in the major diagonal is %.1f",
            sumMajorDiagonal(m));

    }

    public static double sumMajorDiagonal(double[][] m) {
        double sum = 0.0;
        for (int i = 0; i < m.length; i++) {
            sum += m[i][i];
        }
        return sum;
    }
}
```



<terminated> Exercise_08_02 [Java Application] C:\Program Files\Java\jre1.

```
Enter a 4-by-4 matrix row by row:
1 2 3 4.0
5 6.5 7 8
9 10 11 12
13 14 15 16
Sum of the slements in the major diagonal is 34.5
```