

Name: Essmer Sanchez

Section: CSCI 323-222[36386]

Project #: Project 0A

Project Name: Find the largest integer in the input file.

Due date: 2/2/2024 Friday before midnight.

Algorithm Steps:

step 0: inFile = open args [0] // i.e., Scanner inFile= new Scanner(new FileReader (args[0]))

outFile = open args [1] // i.e., FileWriter outFile1 = new FileWriter (args [1]);

step 1: largest = -9999

step 2: data = read an integer at a time from inFile // i.e., data = inFile.nextInt();

outFile = output data to outFile // i.e., outFile.write (data)

step 3: if data > largest

largest = data

step 4: repeat step 2 to step 3 until inFile is empty.

step 5: outFile = write "The largest integer in the input file is" largest.

Step 6: close inFile and outFile.

Source Code:

```
import java.util.Scanner;
```

```
import java.io.File;
```

```
import java.io.FileReader;
```

```
import java.io.FileWriter;
```

```
import java.io.IOException;
```

```
public class SanchezE_Project0A_Main {
```

```

public static void main(String[] args){
    if (args.length < 3){
        System.err.println("Invalid Number of Arguments, 3 required <inputFile> <outputFile>
<debugFile>");
        System.exit(1);
    }

    File inFile = new File(args[0]);
    File outFile = new File(args[1]);
    File debugFile = new File(args[2]);

    int largest = -9999;

    //try-catch automatically closes Scanner and FileWriter
    try (
        Scanner scanner = new Scanner(new FileReader(inFile));
        FileWriter writer = new FileWriter(outFile);
        FileWriter debugWriter = new FileWriter(debugFile)){

        //start debugging
        debugWriter.write("Starting File Read-In:\n");

        while (scanner.hasNextInt()){
            int data = scanner.nextInt();
            writer.write(data + "\n");

            debugWriter.write("Input number read: " + data + "\n");

            if (data > largest) {

```

```
        largest = data;
        debugWriter.write("Largest updated to: " + largest + "\n");
    }
}

writer.write("The largest integer in the input file is: " + largest);

//debugger end
debugWriter.write("Completed File Processing, Largest found: " + largest + "\n");

} catch (IOException e) {
    e.printStackTrace();
}
}
}
```

Output File:

91

322

702

8

999

12

133

14

415

213

23

724

825

226

127

538

29

91

91

91

730

361

637

213

213

388

91

322

73

213

4

95

16

702

The largest integer in the input file is: 999

Debug File:

Starting File Read-In:

Input number read: 91

Largest updated to: 91

Input number read: 322

Largest updated to: 322

Input number read: 702

Largest updated to: 702

Input number read: 8

Input number read: 999

Largest updated to: 999

Input number read: 12

Input number read: 133

Input number read: 14

Input number read: 415

Input number read: 213

Input number read: 23

Input number read: 724

Input number read: 825

Input number read: 226

Input number read: 127

Input number read: 538

Input number read: 29

Input number read: 91

Input number read: 91

Input number read: 91

Input number read: 730

Input number read: 361

Input number read: 637

Input number read: 213

Input number read: 213

Input number read: 388

Input number read: 91

Input number read: 322

Input number read: 73

Input number read: 213

Input number read: 4

Input number read: 95

Input number read: 16

Input number read: 702

Completed File Processing, Largest found: 999