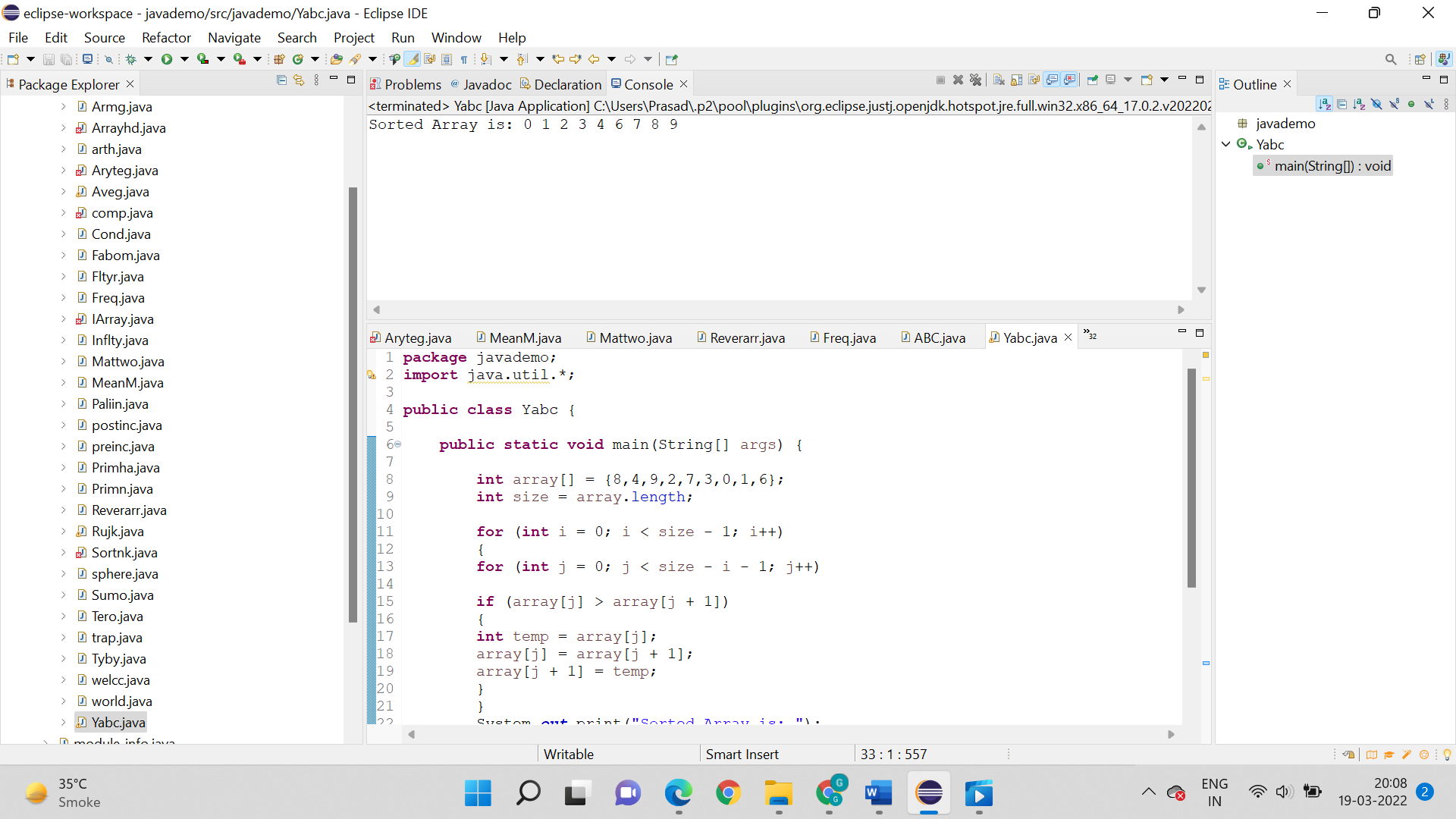
**Programs on Arrays**

1. Write a program to sort the 1-D array elements without using the inbuilt function.

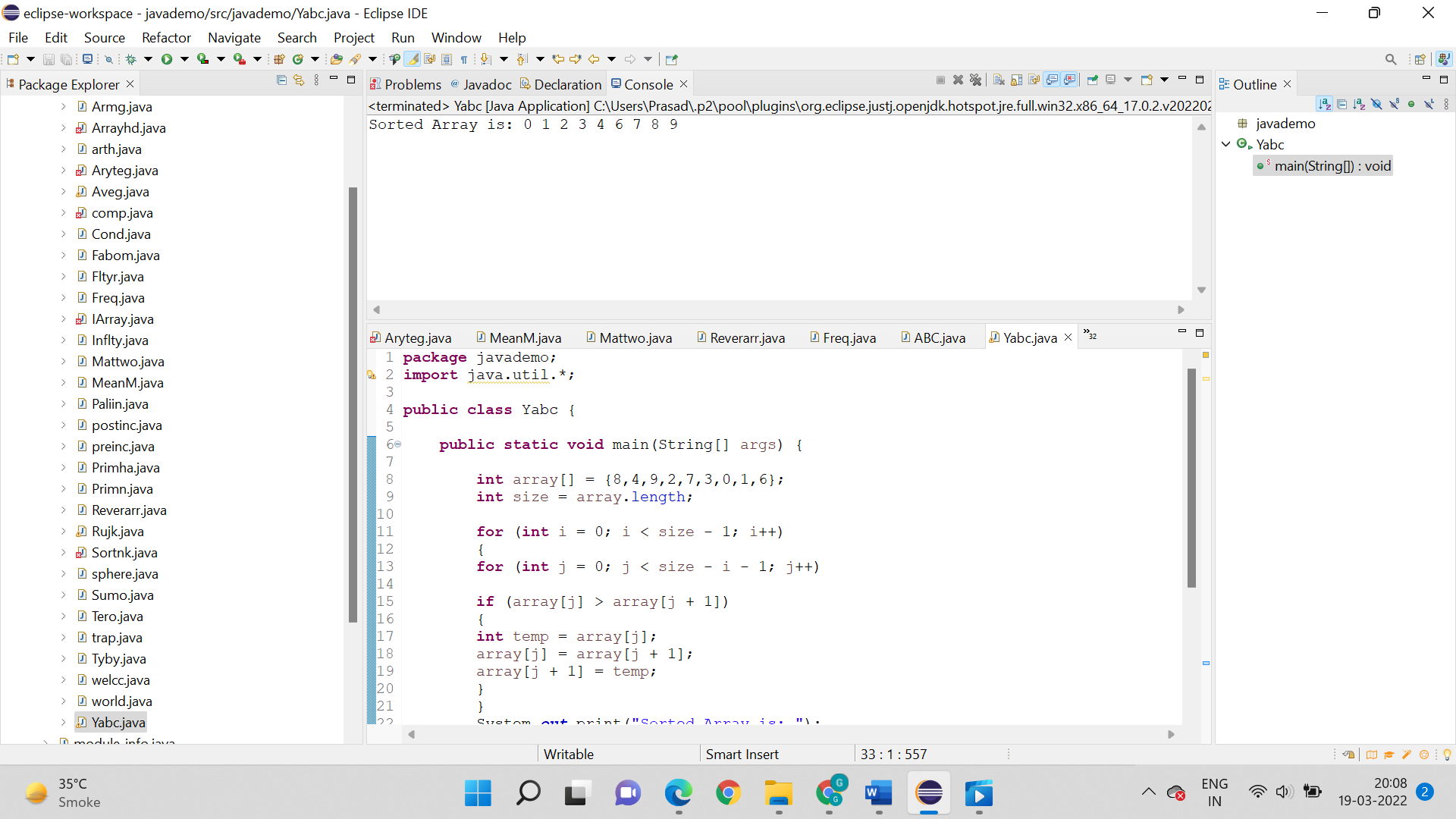
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Assign array
* Step 4: Use for and if loop to sort array
* Step 5: temp=array[j]
* Step 6: array[j+1]=temp
* Step 7 : Stop

Code:



Output:



1. Write a program to sort the 1-D array elements using the inbuilt function. (Use Array.sort())

Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: int array[] = {8,4,9,2,7,3,0,1,6};
* Step 4: Arrays.sort(array) is used to sort array
* Step 5: System.out.println is used to print sorted array
* Step 6: Stop

Code:



Output:

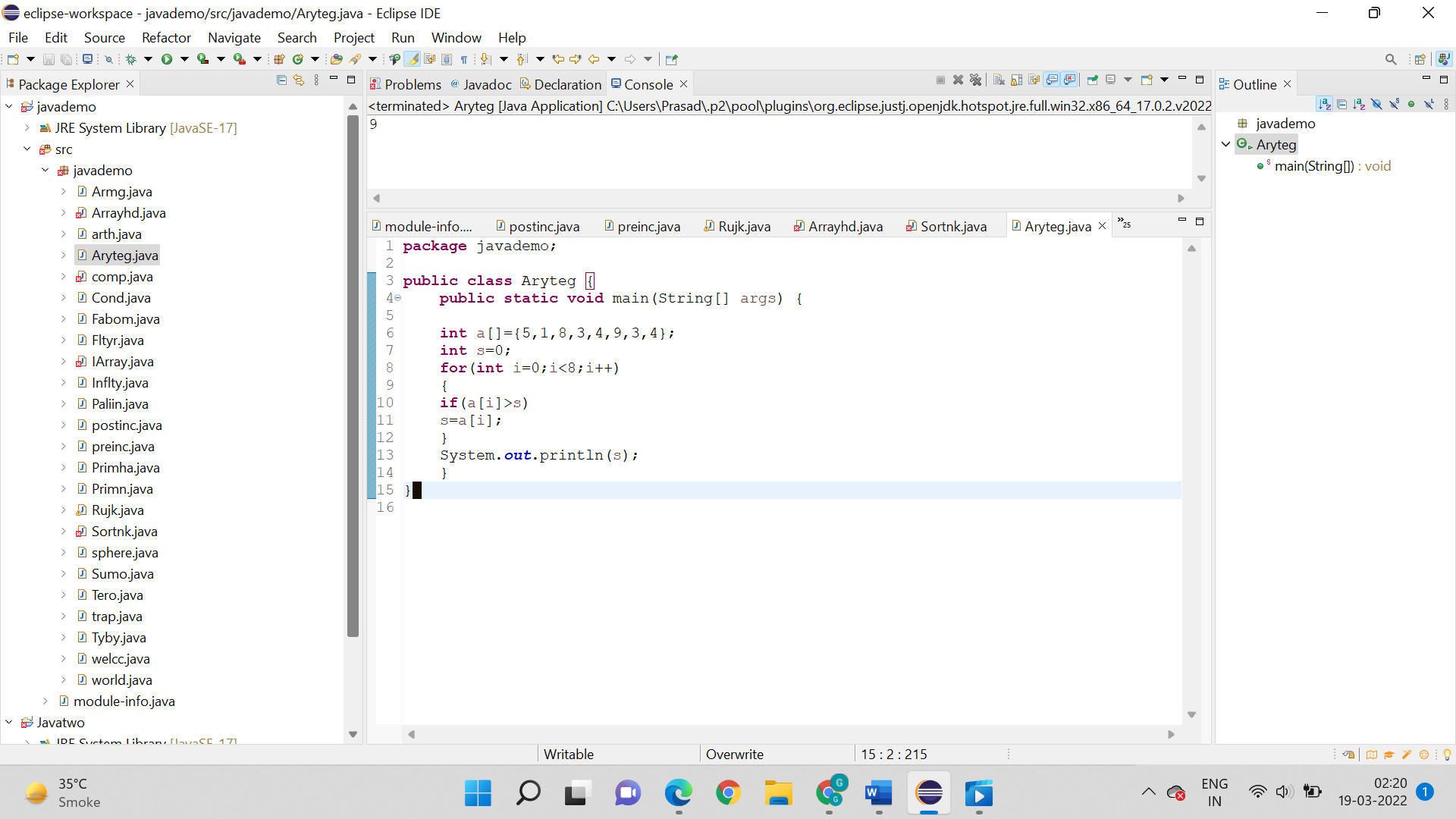


1. Write a program to find out the largest number within 1-D array elements. Display the array and the largest number.

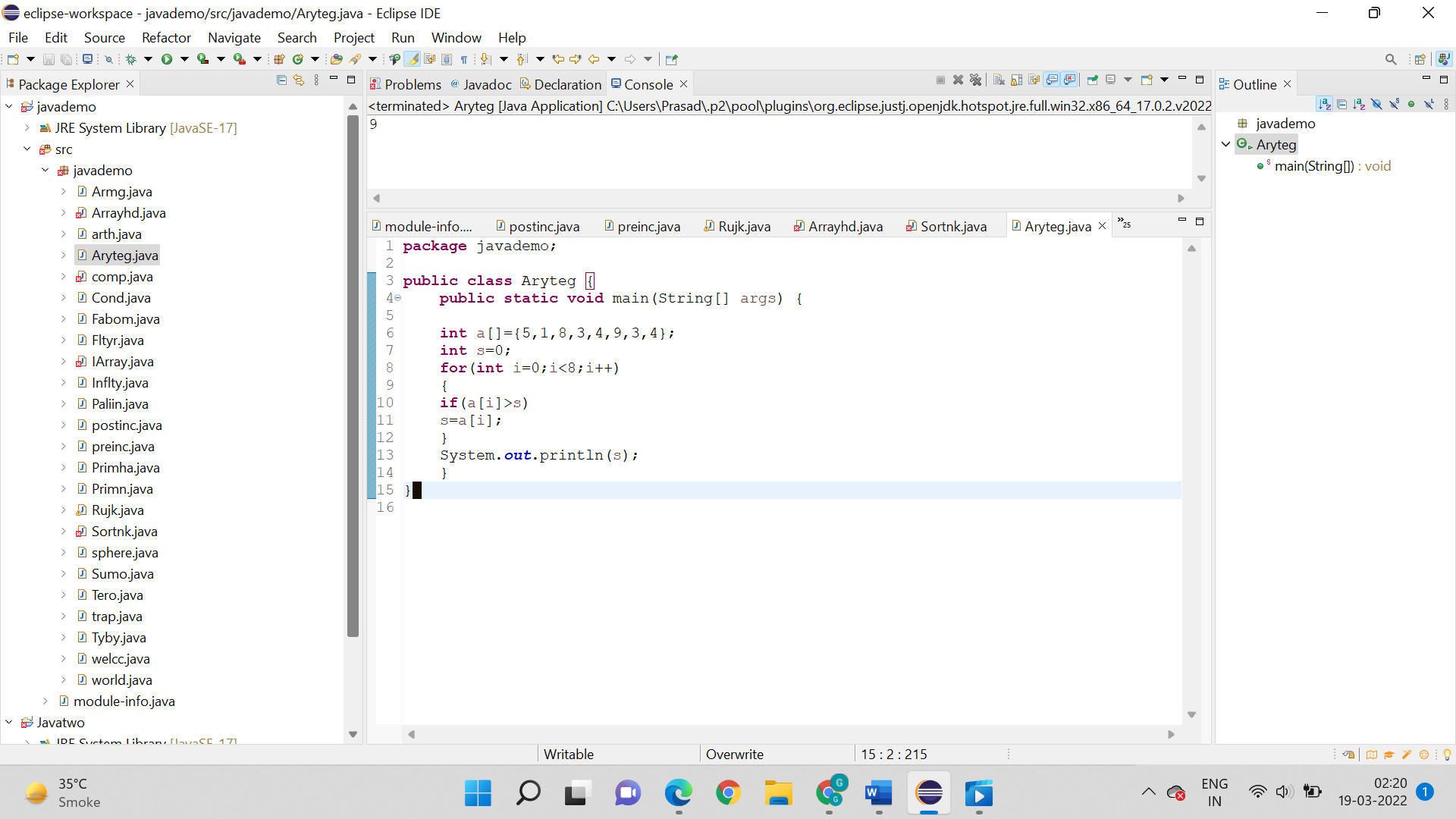
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Use for loop to find the largest no. in array
* Step 4: Use if loop for the algorithm: if(a[i]>s) s=a[i];
* Step 5: System.out.println is used to print s
* Step 6 : Stop

Code:



Output:

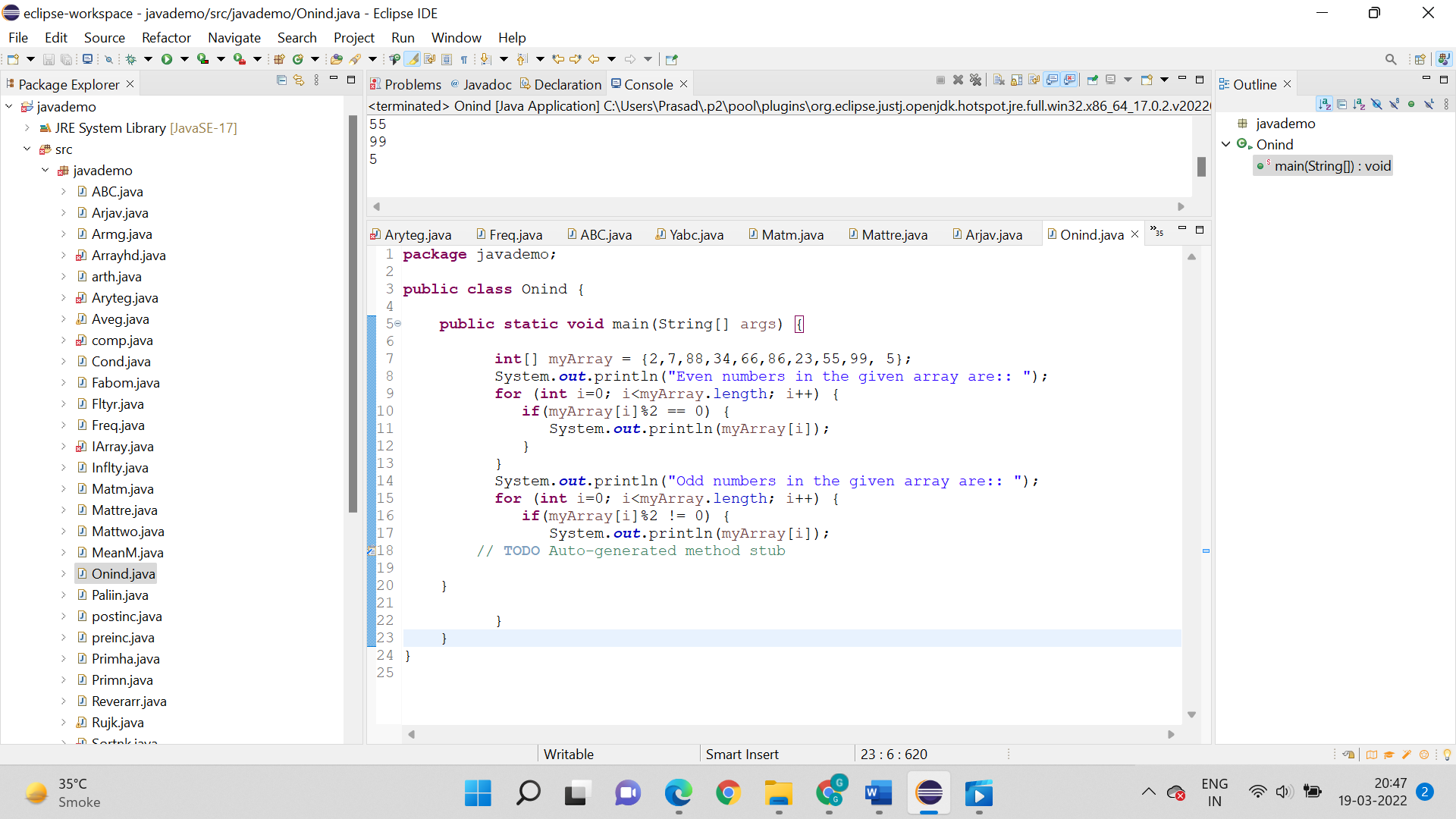


1. Create a one-dimensional array with 10 integer values and write a program to determine the element on each index is odd or even.

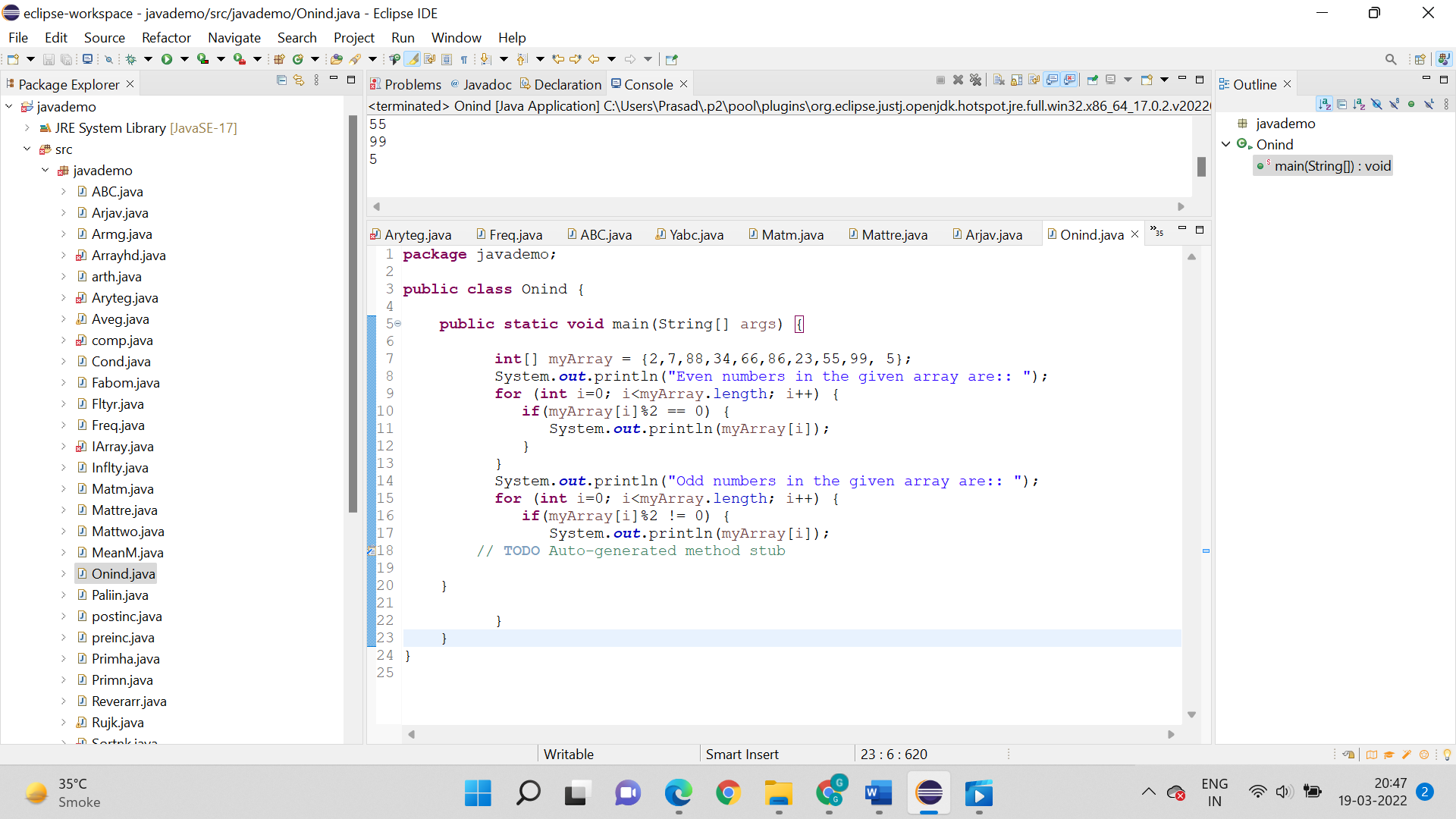
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: myarray is created and for loop is used to give even and odd no. in array.
* Step 4: for (int i=0; i<myArray.length; i++) {
  + if(myArray[i]%2 != 0) {
* Step 5: System.out.println is used to print the output
* Step 6: Stop

Code:



Output:

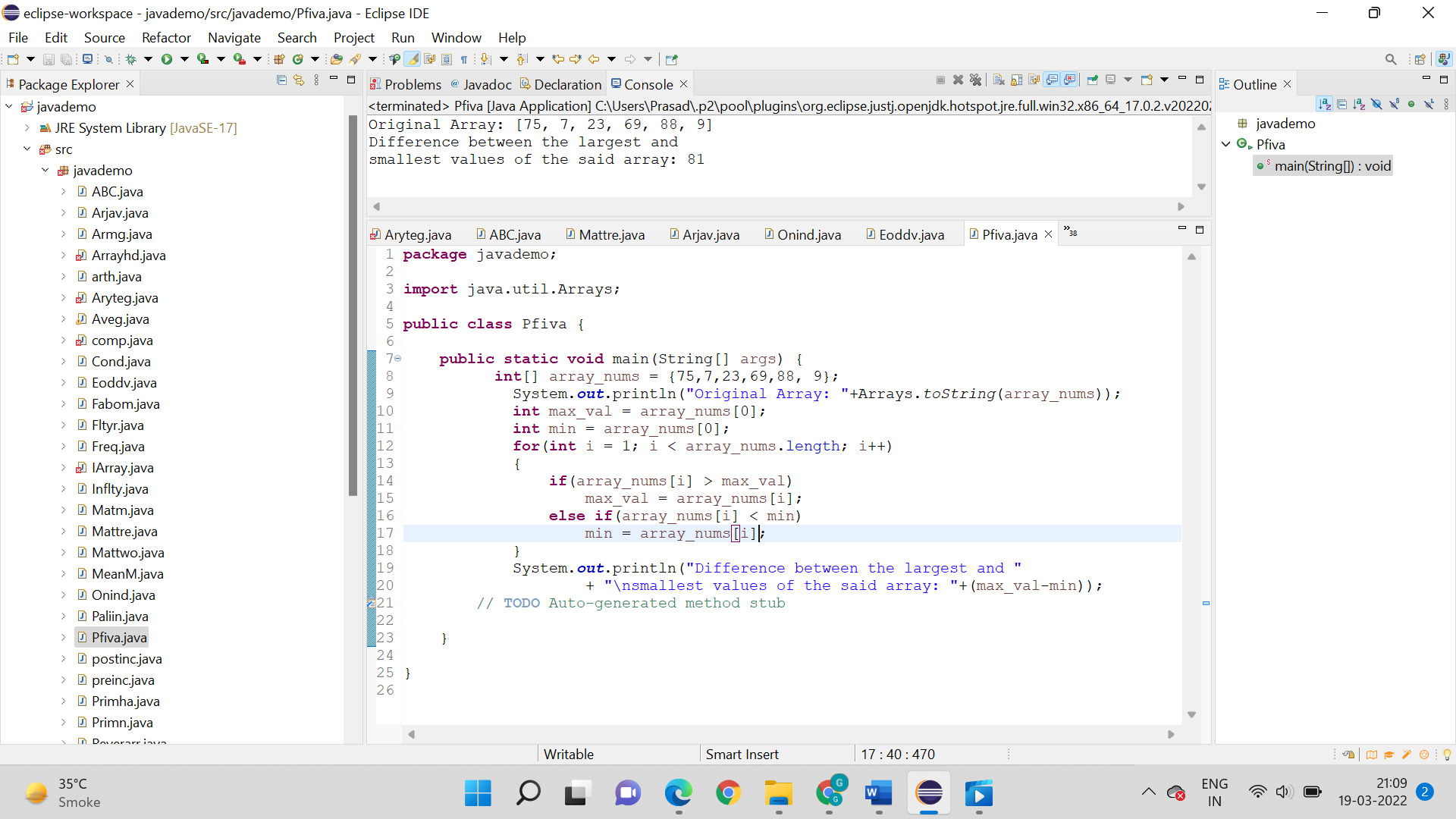


1. Write a program to determine the difference between the largest and smallest values in a 1D-array of length 10. Display the largest and the smallest value.

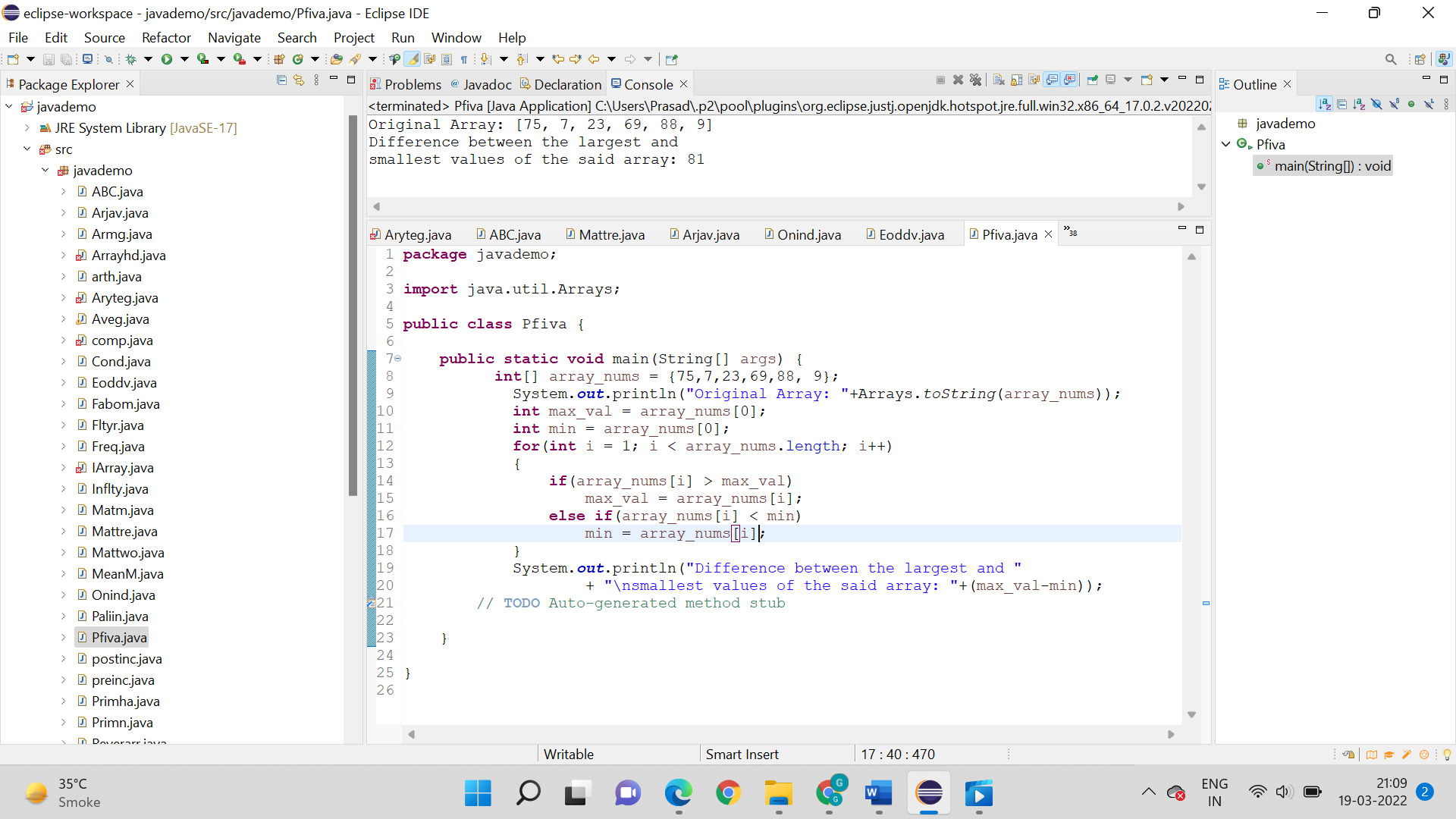
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: array\_num stores vaule of the array
* Step 4: int max\_val = array\_nums[0];
  + - int min = array\_nums[0];
* Step 5: For loop is created for the algorithm
* Step 6: System.out.println is used to print Difference between the largest and smallest values of the array.
* Step 7: Stop

Code:



Output:

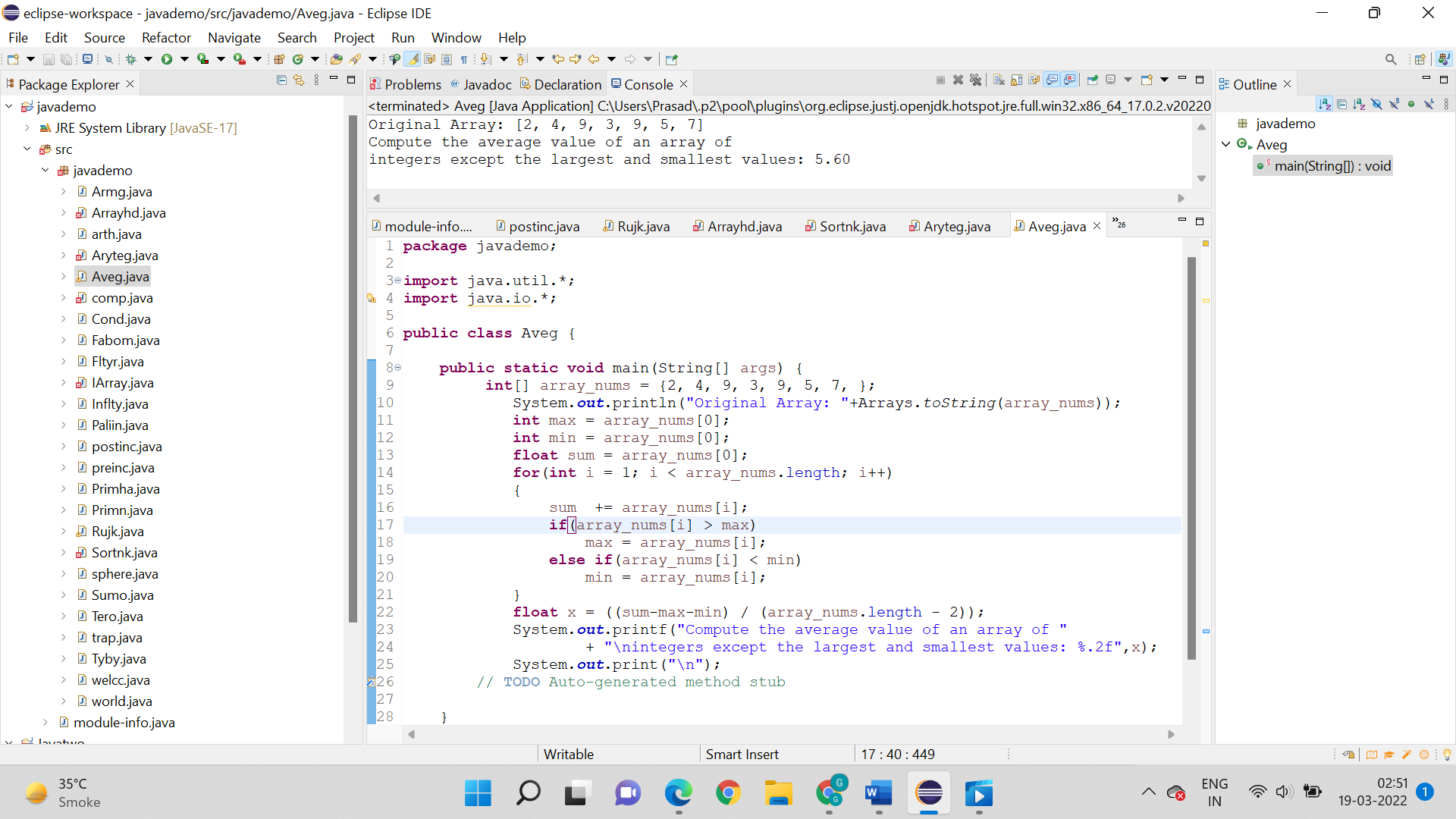


1. Write a program to calculate the average value of a 1D-array of length 7, except the largest and smallest values.

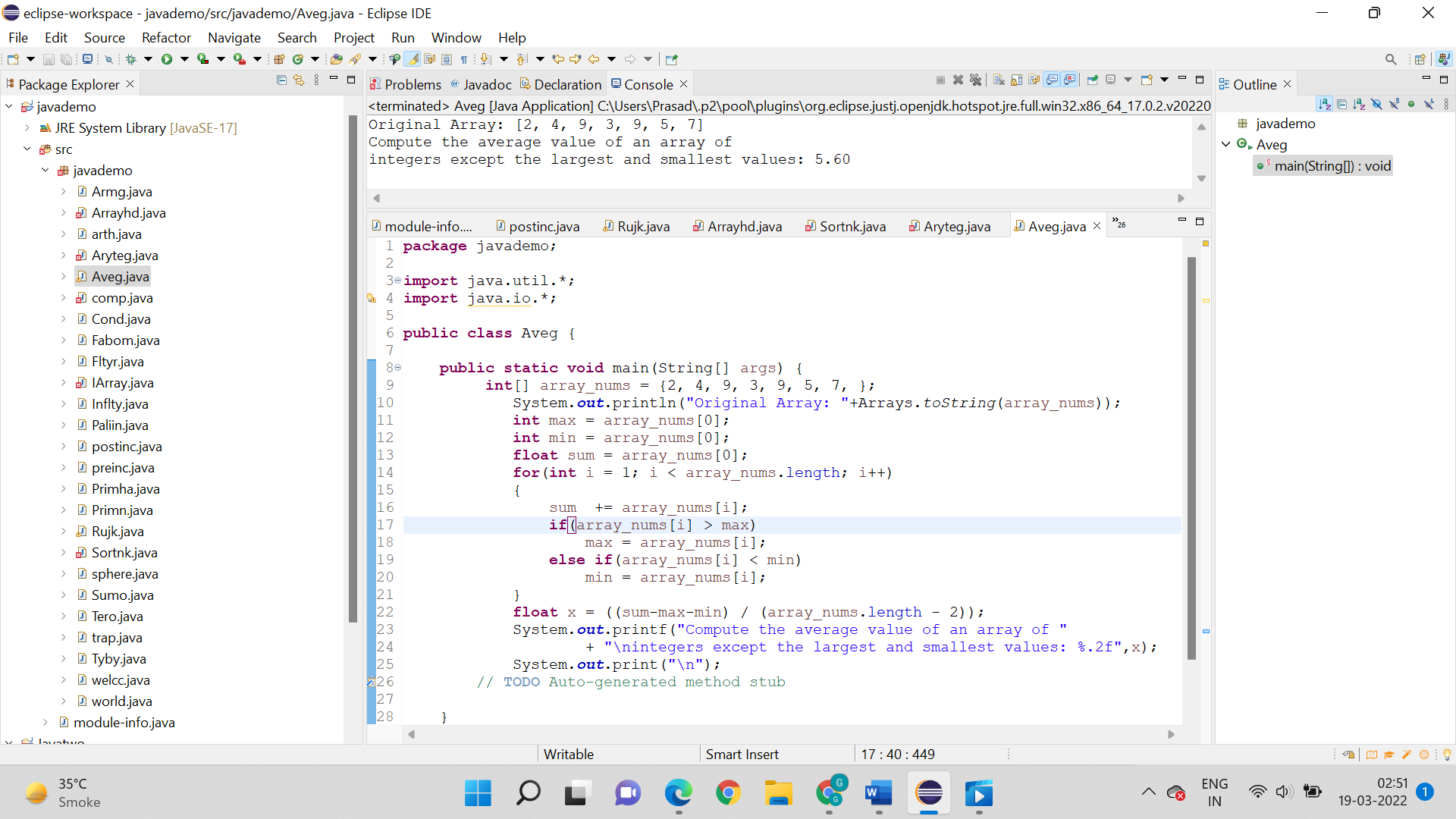
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: array\_num stored 7 values of array
* Step 4: for loop is used for the algorithm
* Step 5: if(array\_nums[i] > max)
* max = array\_nums[i];
* else if(array\_nums[i] < min)
* min = array\_nums[i];
* Step 6: System.out.println is used to print the average value of an array of integers except the largest and smallest values.
* Step 7 : Stop

Code:



Output:

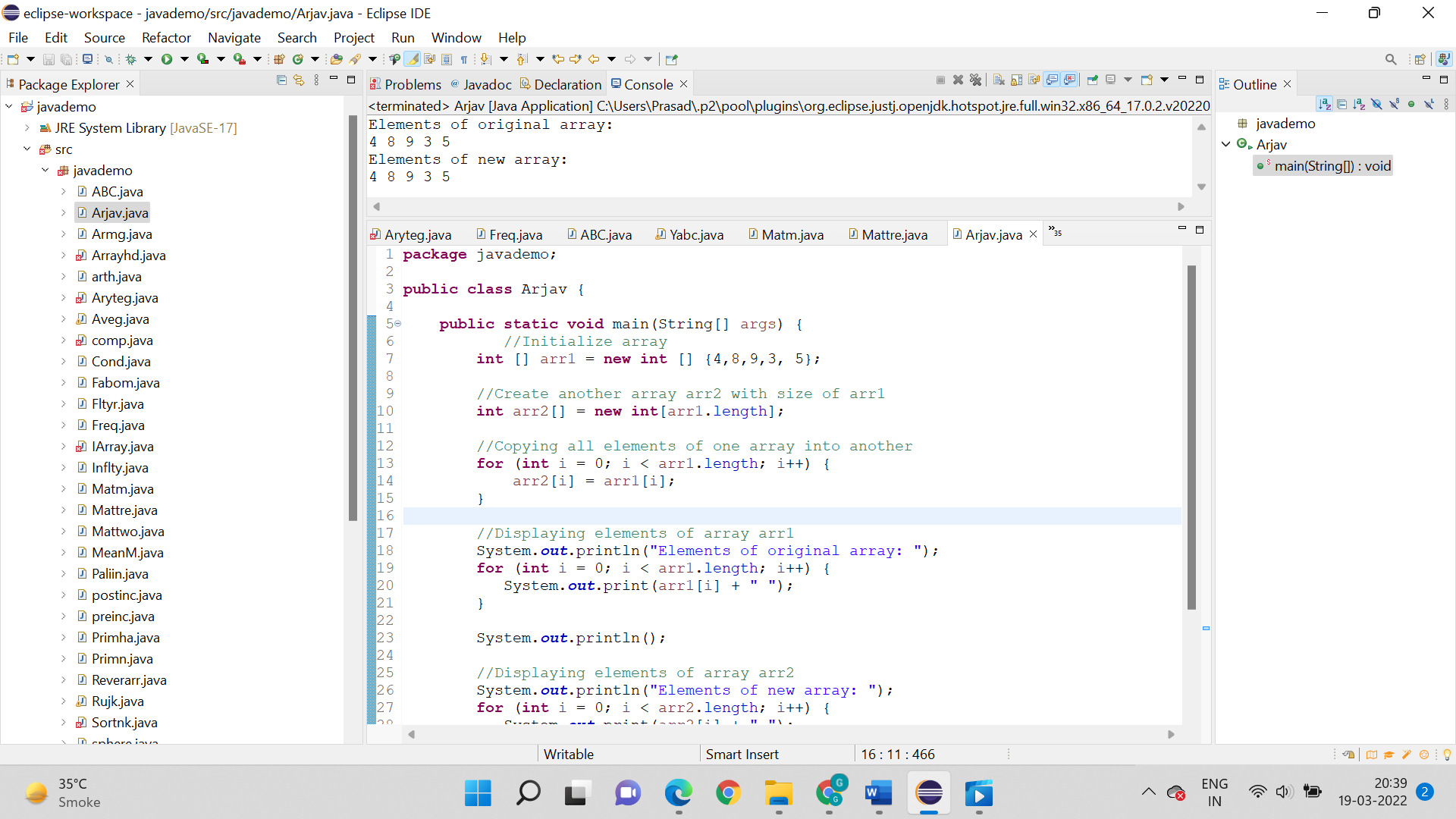


1. Write a program to copy all the elements of a 1-D array into another array.

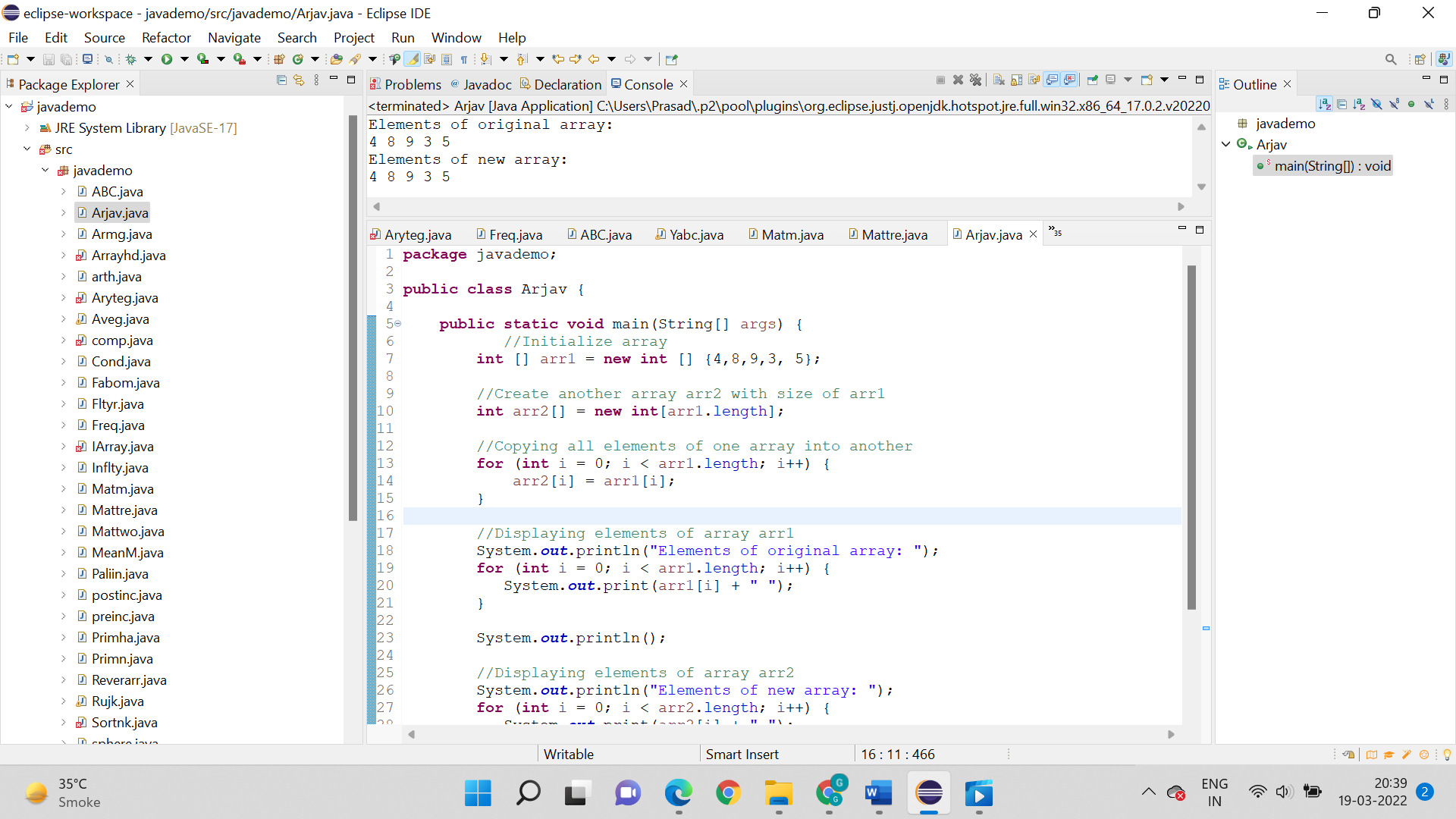
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Create another array arr2 with size of arr1
* Step 4: Copying all elements of one array into another
* Step 5: Displaying elements of array arr1
* Step 6: Displaying elements of array arr2
* Step 7: System.out.println is used to print elements of new array
* Step 8: Stop

Code:



Output:

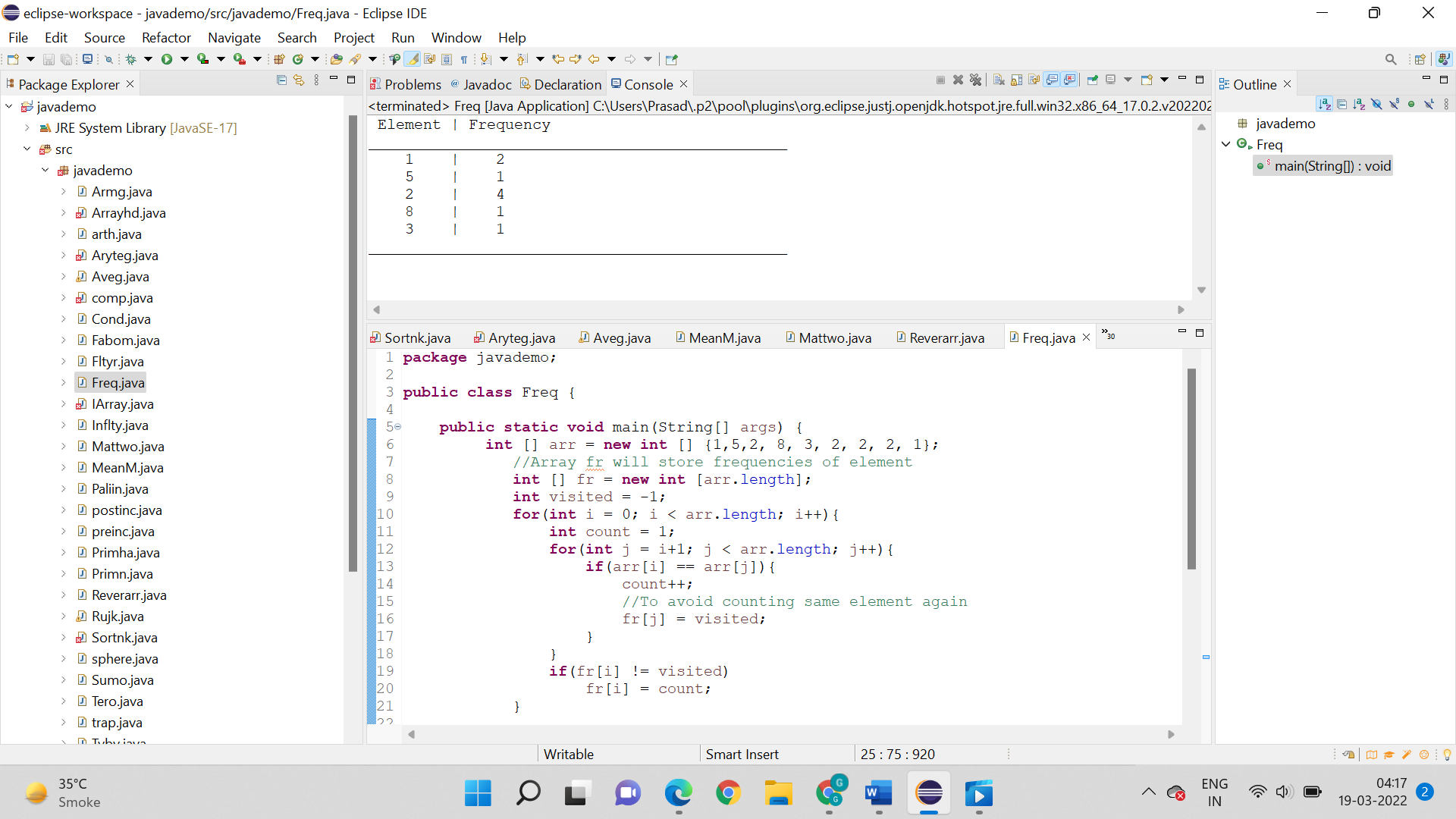


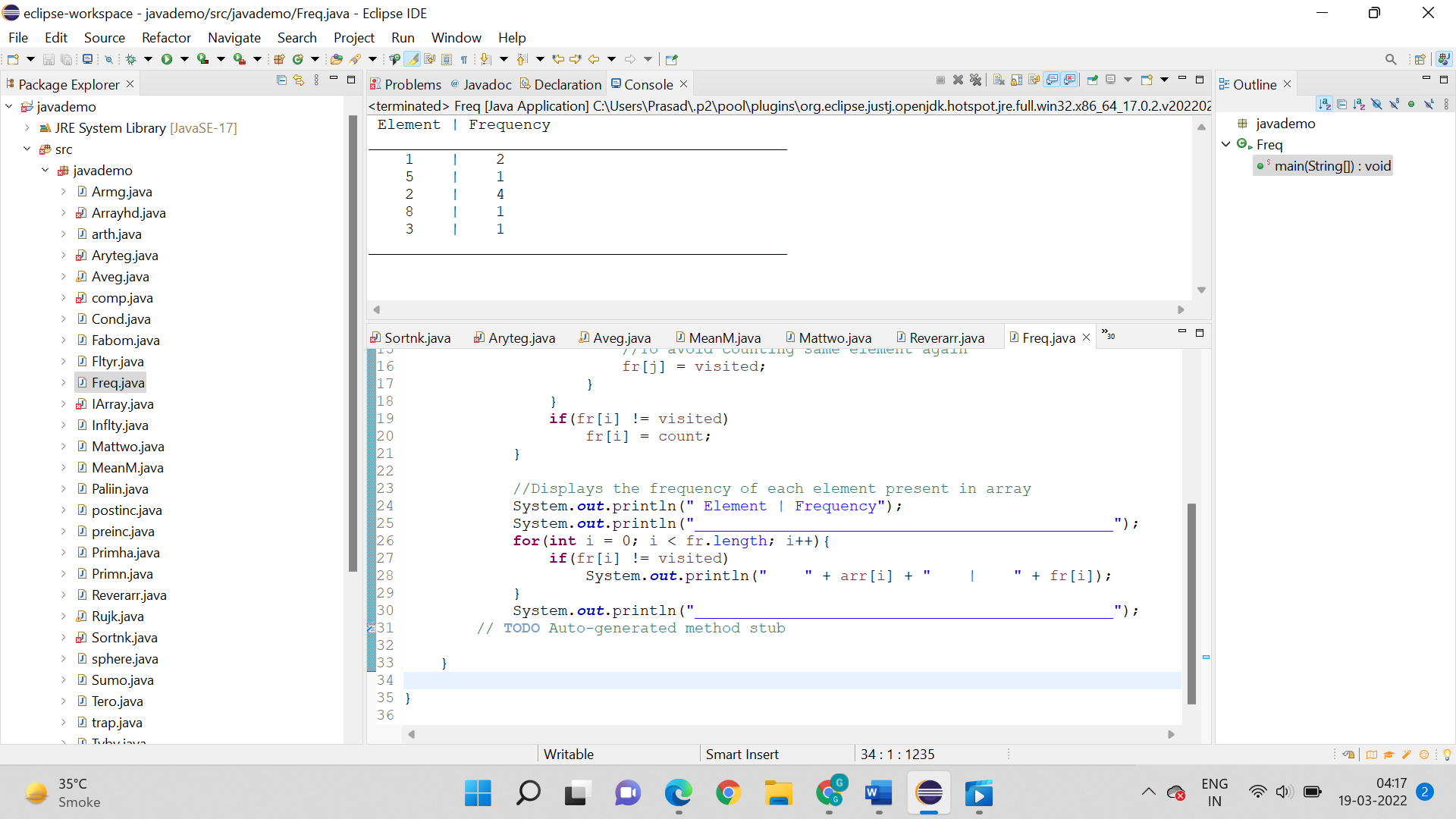
1. Write a program to print the duplicated elements in a 1-D array.

Logic:

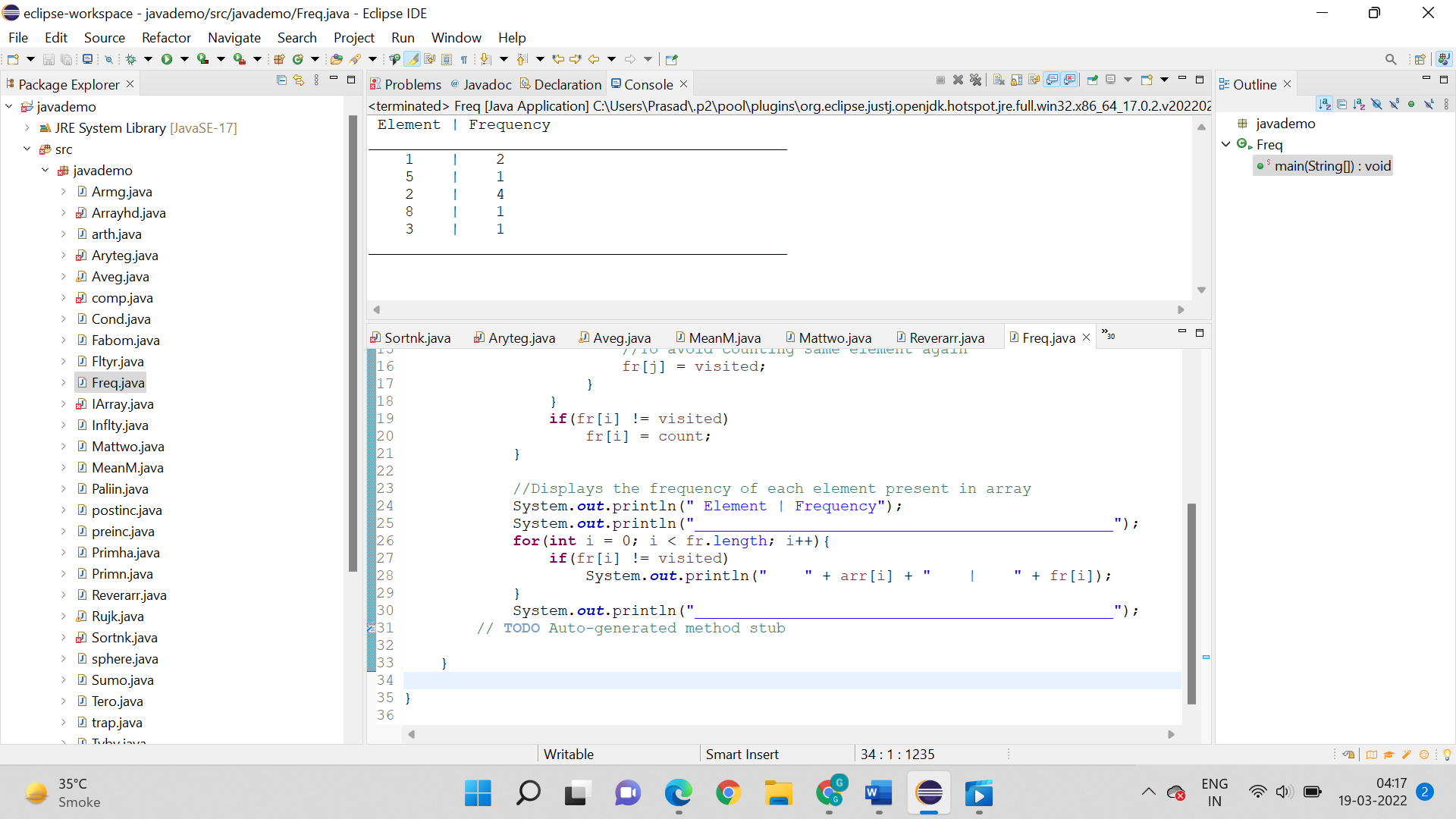
* Step 1: start
* Step 2: initialize arr[] ={1, 2, 8, 3, 2, 2, 2, 5, 1 }.
* Step 3: create fr[] of arr[] length.
* Step 4: set visited = -1.
* Step 5: repeat step 6 to step 9 for(i=0;i<arr.length;i++)
* Step 6: set count = 1
* Step 7: repeat step 8 for(j=i+1;j<arr.length;j++)
* Step 8: if(arr[i]==arr[j]) then
  + count++
  + fr[j] =visited
* Step 9: if(fr[i]!=visited) then
  + fr[i]=count
* Step 10: print "------------"
* Step 11: print "element | frequency"
* Step 12: print "-------------"
* Step 13: repeat step 14 for(i=0;i<fr.length;i++)
* Step 14: if(fr[i]!=visited) then
  + print arr[i] and fr[i]
* Step 15: print "-------------"
* Step 16: end

Code:





Output:

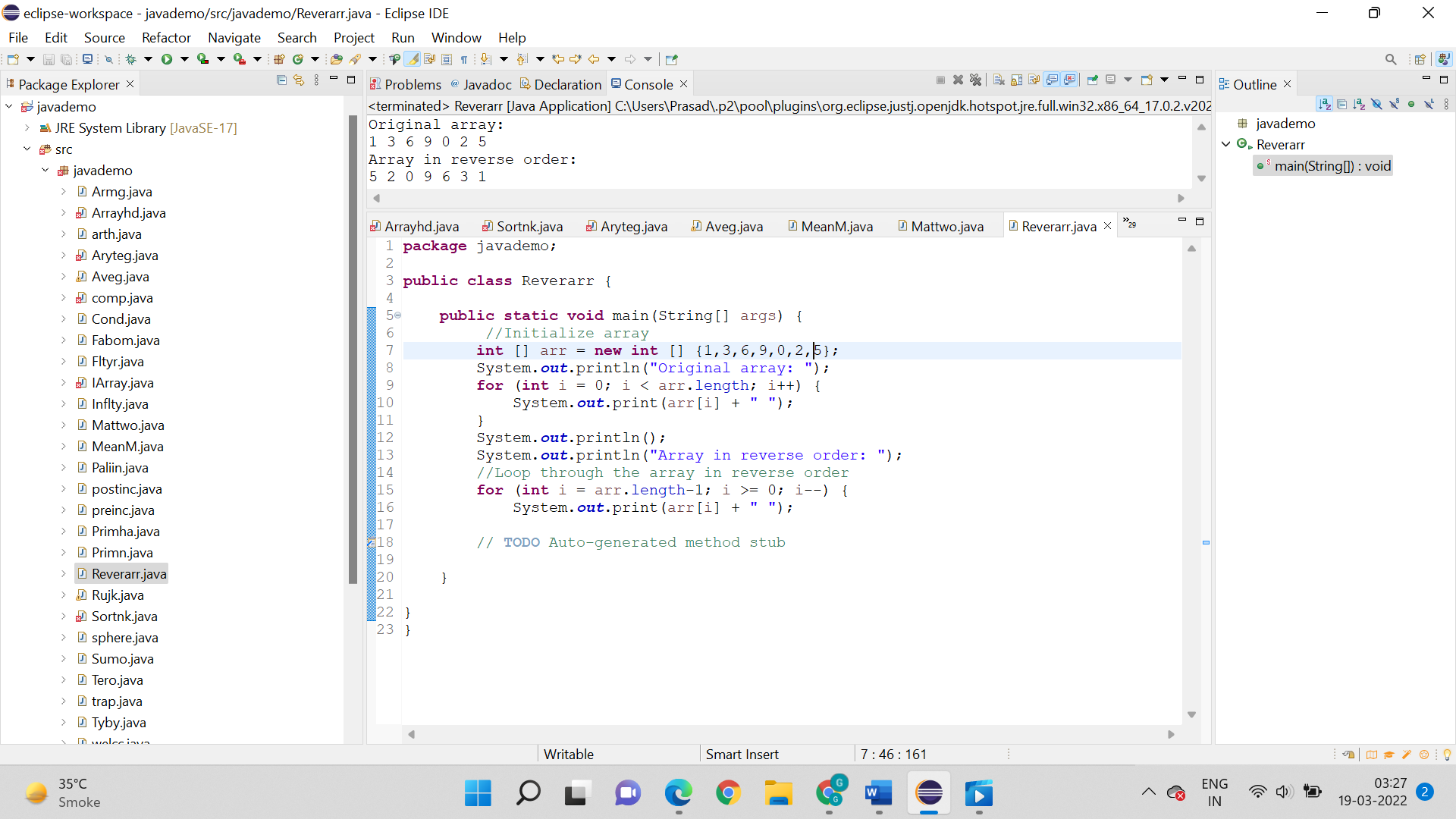


1. Write a program to determine the frequency of each element in a 1-D array.

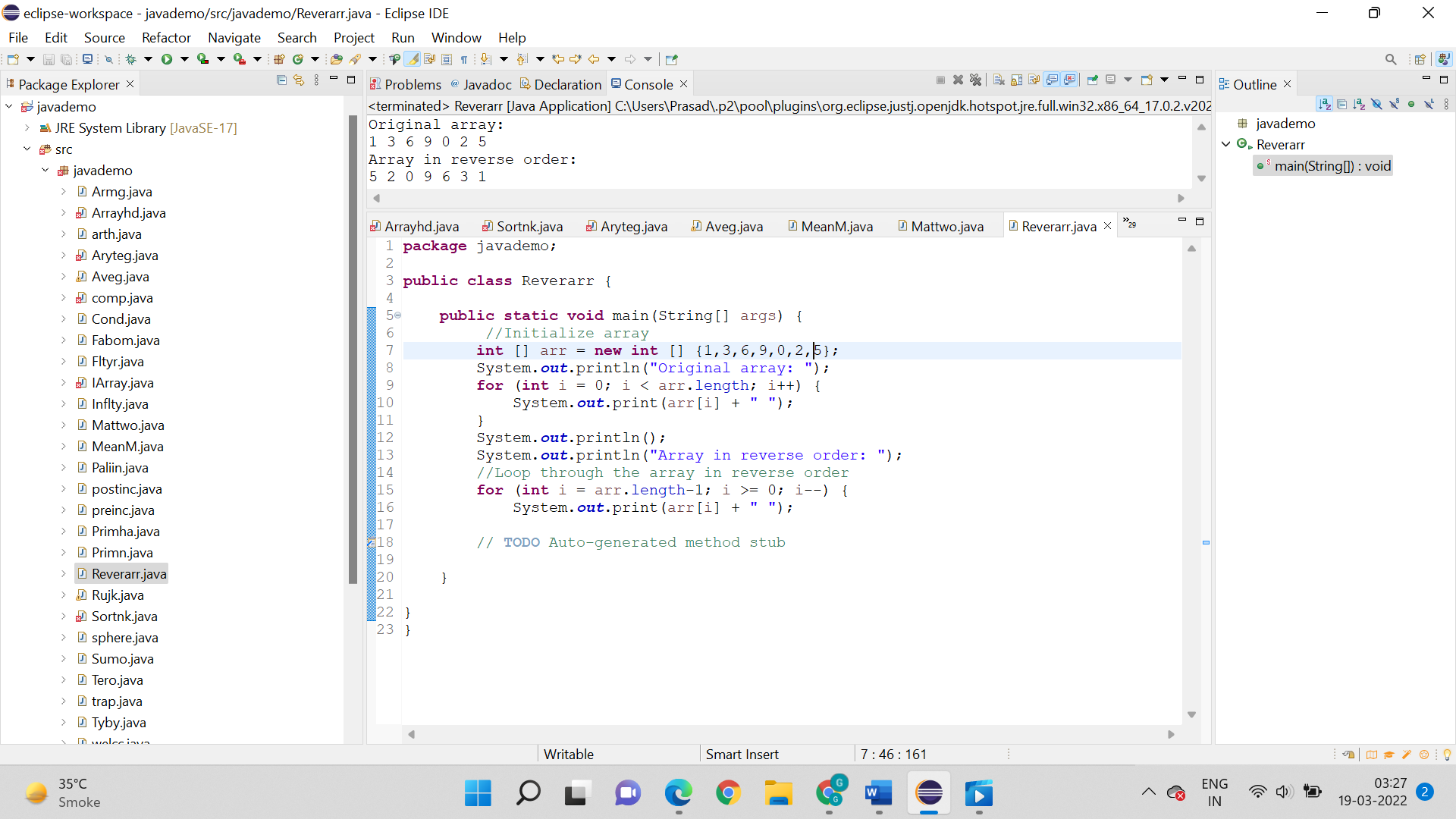
Logic:

* Step 1: start
* Step 2: initialize arr[] = {1, 2, 3, 4, 5}
* Step 3: print "original array:"
* Step 4: repeat step 5 for(i=0; i<arr.length ; i++)
* Step 5: print arr[i]
* Step 6: print "array in reverse order"
* Step 7: repeat step 8 for(i= arr.length-1; i>=0; i--)
* Step 8: print a[i]
* Step 9: end

Code:



Output:

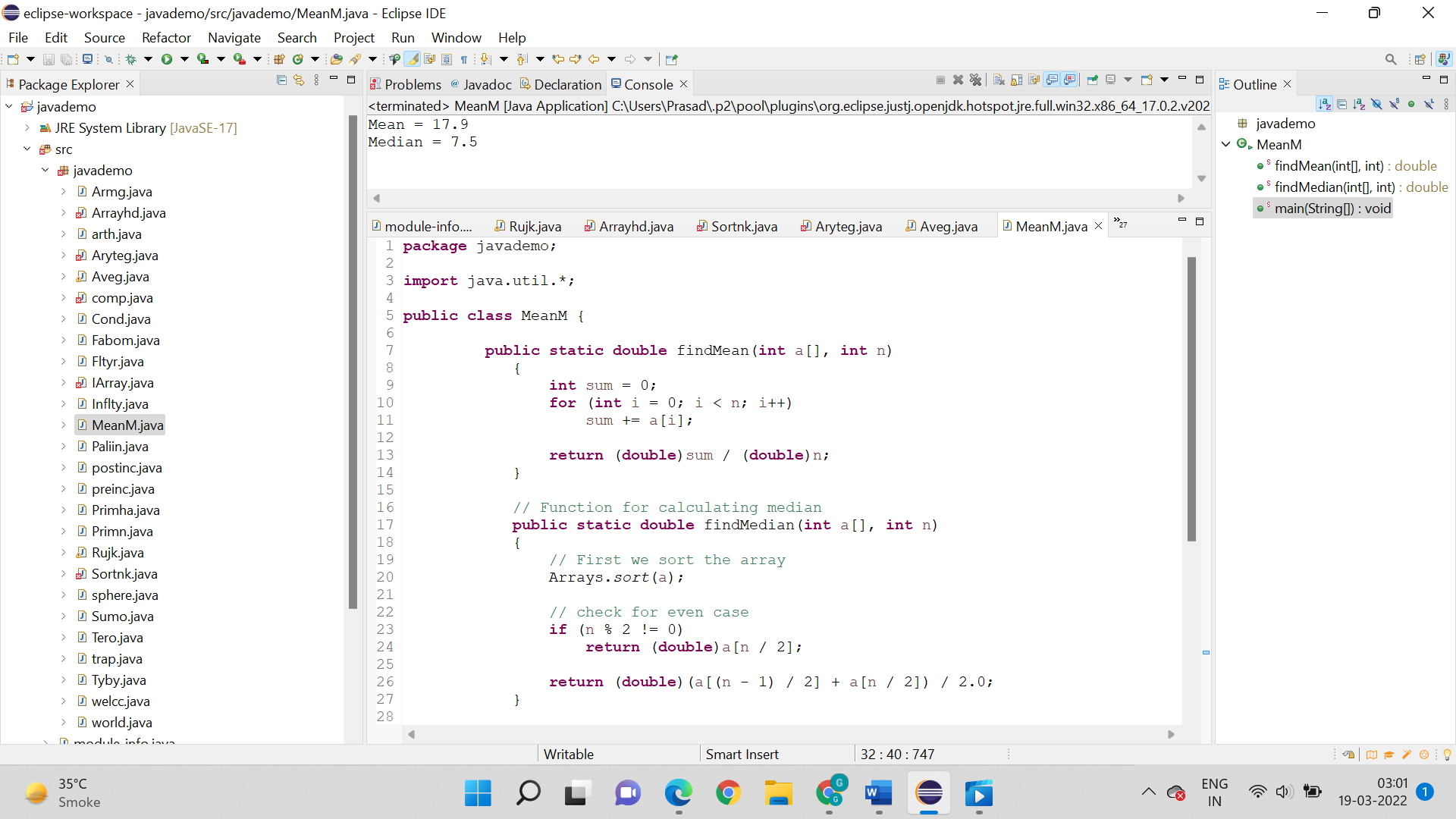


11.Write a program to calculate the mean, median, and mode of a 1-D array.

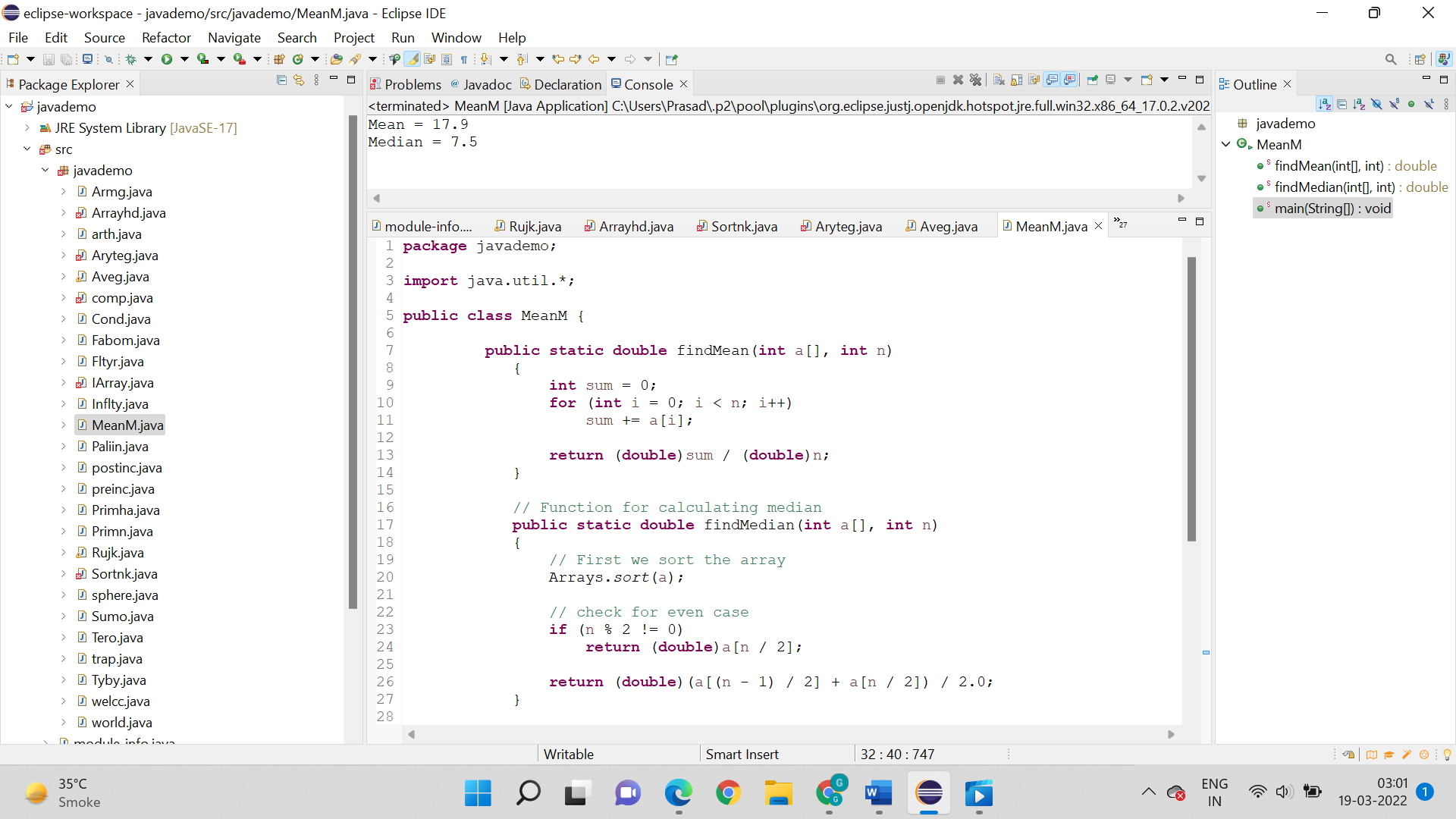
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Write function for calculating median
* Step 4: First we sort the array by using Arrays.sort(a);
* Step 5: Than check even case
* Step 6: Call Function now
* Step 7: System.out.println is used to print mean median.
* Step 8: Stop

Code:



Output:

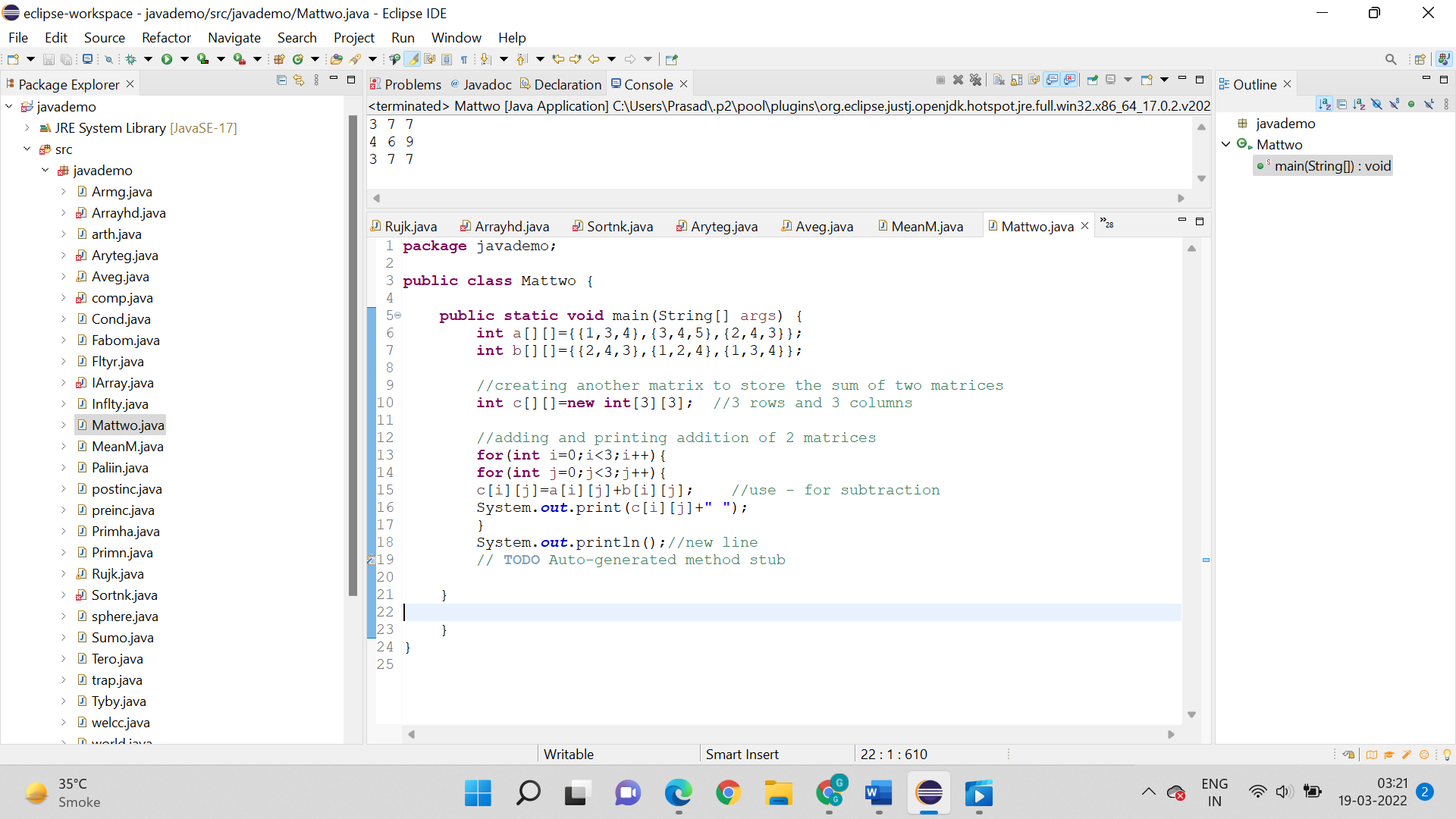


12. Write a program to perform the addition of two 2-D arrays.

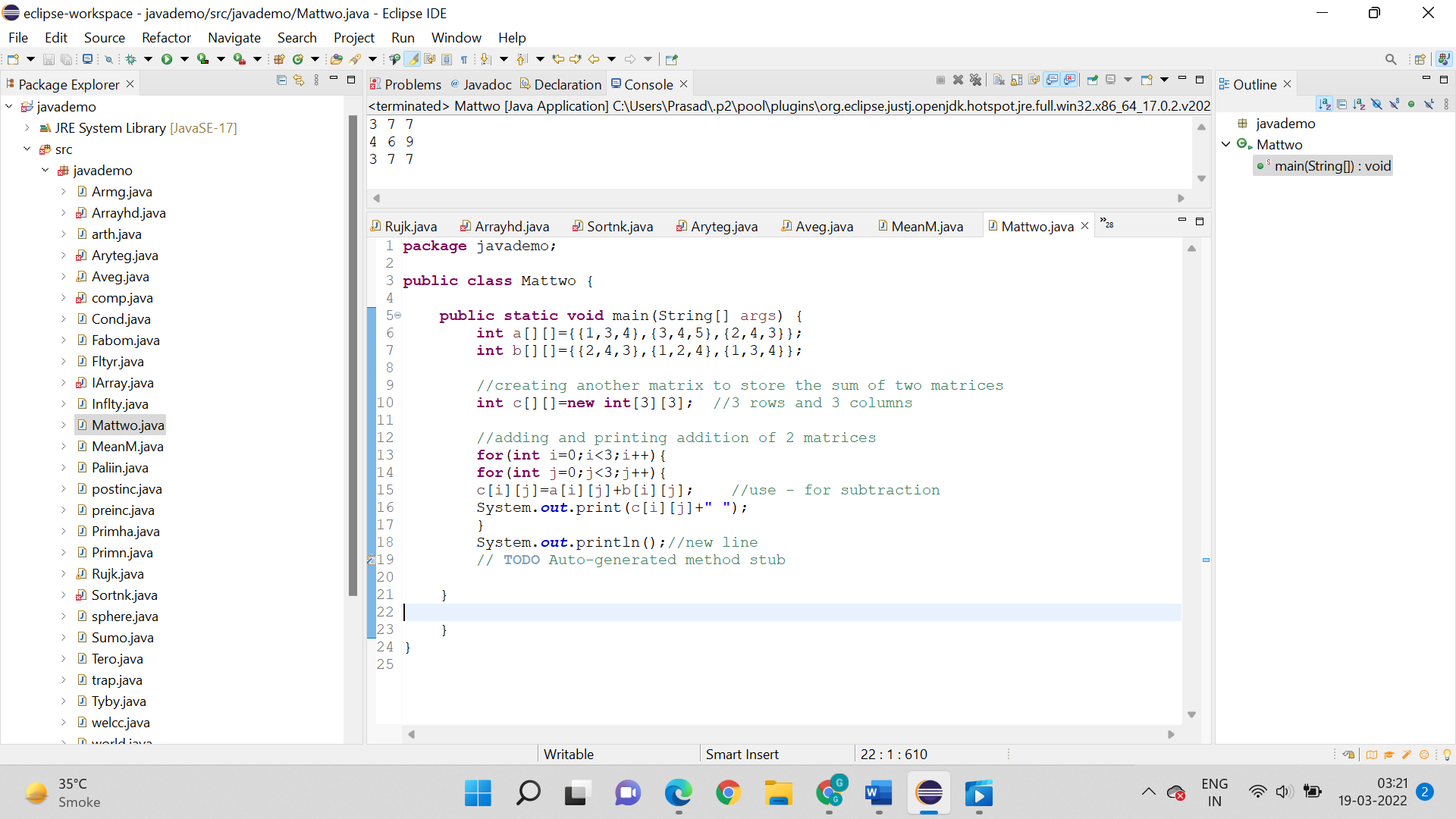
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Create array a and b
* Step 4: Creating another matrix to store the sum of two matrices
* Step 5: Adding and printing addition of 2 matrices
* Step 6: System.out.println is used to print output of addition of 2 arrays
* Step 7: Stop

Code:



Output:



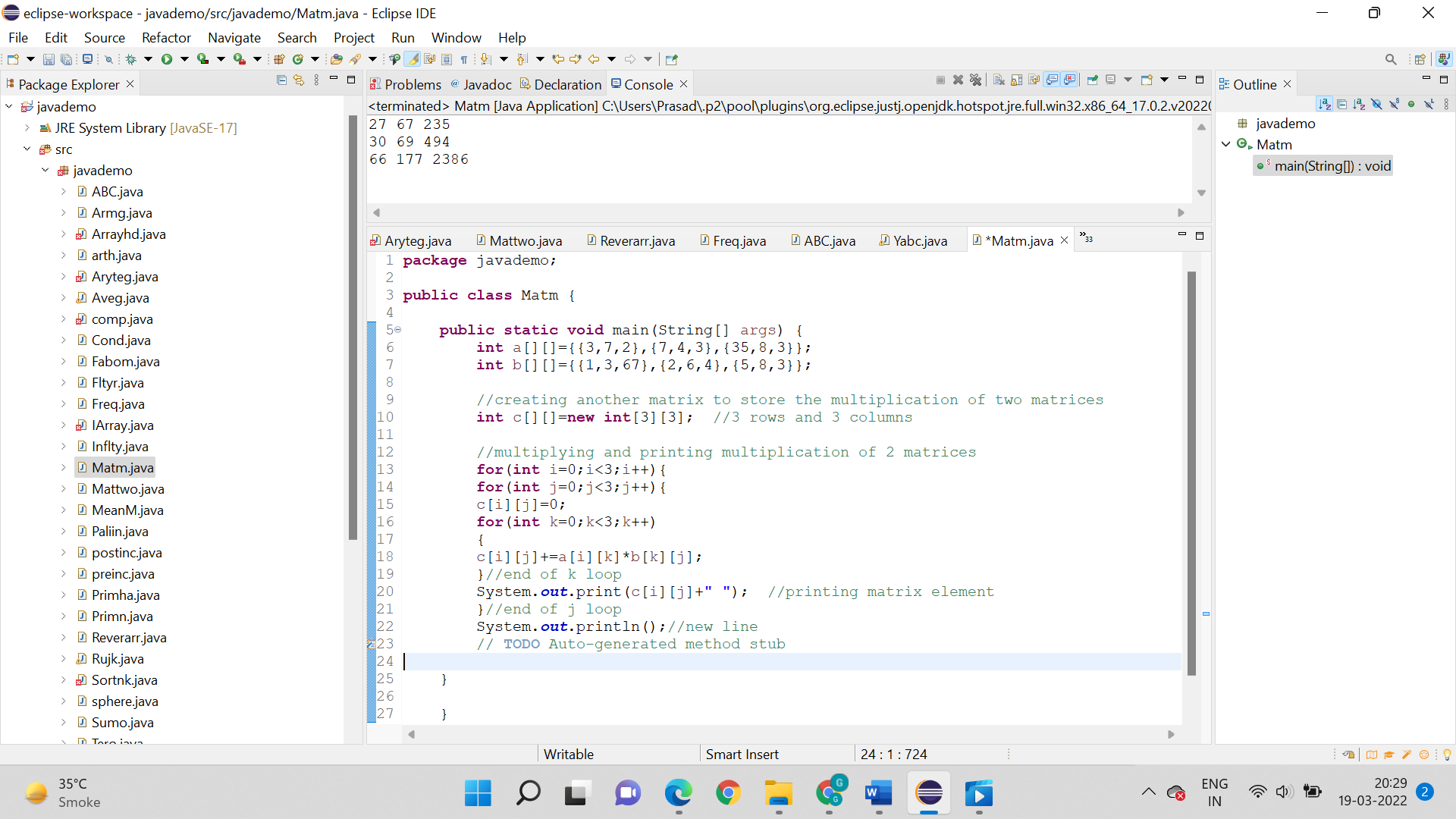
13. Write a program to perform the multiplication of two 2-D arrays.

* 1. Direct Multiplication

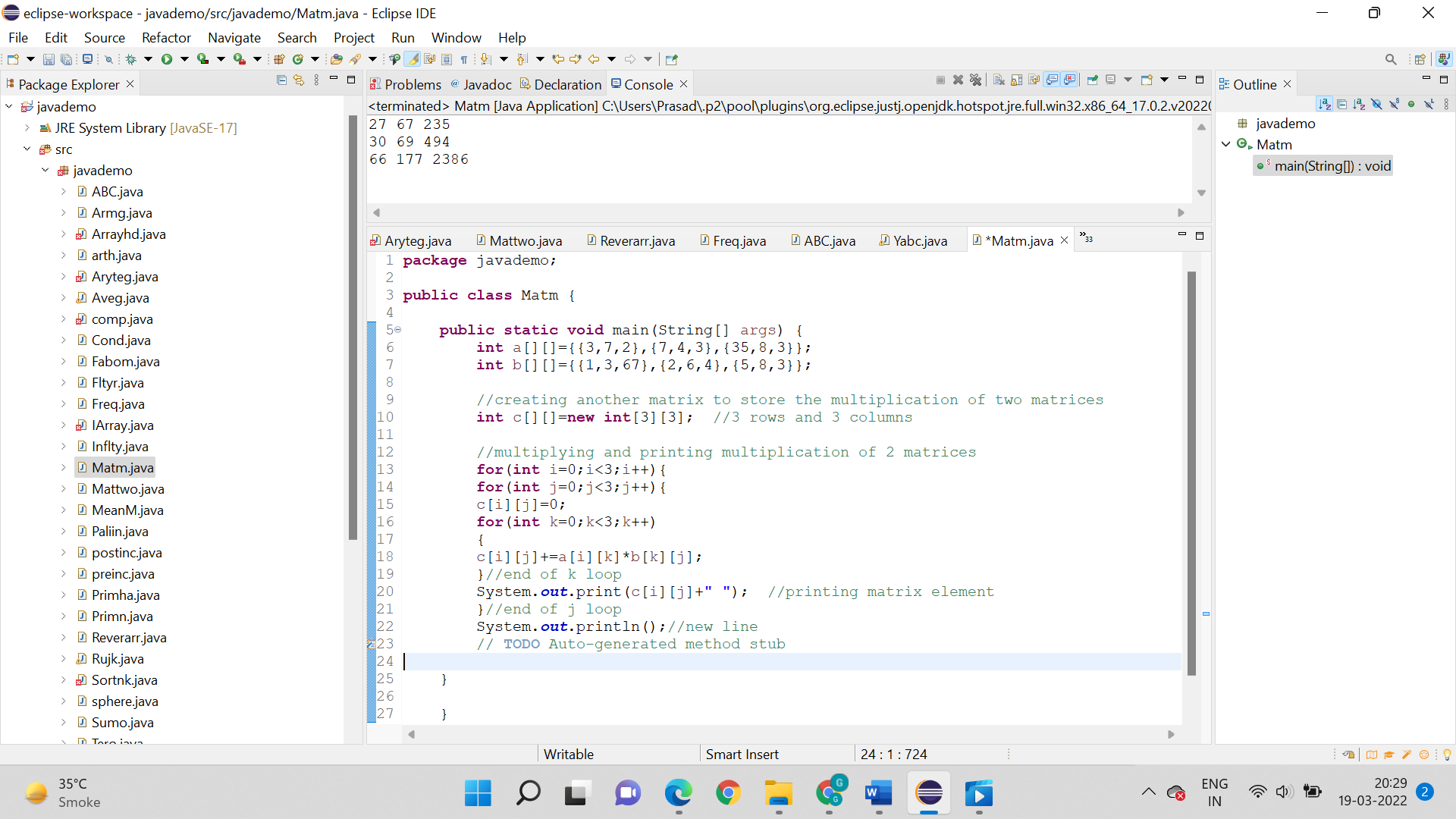
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Create array a and b
* Step 4: Creating another matrix to store the multiplication of two matrices
* Step5:Multiplying and printing multiplication of 2 matrices
* Step 6: System.out.println is used to print multiplication of matrix
* Step 7: Stop

Code:



Output:

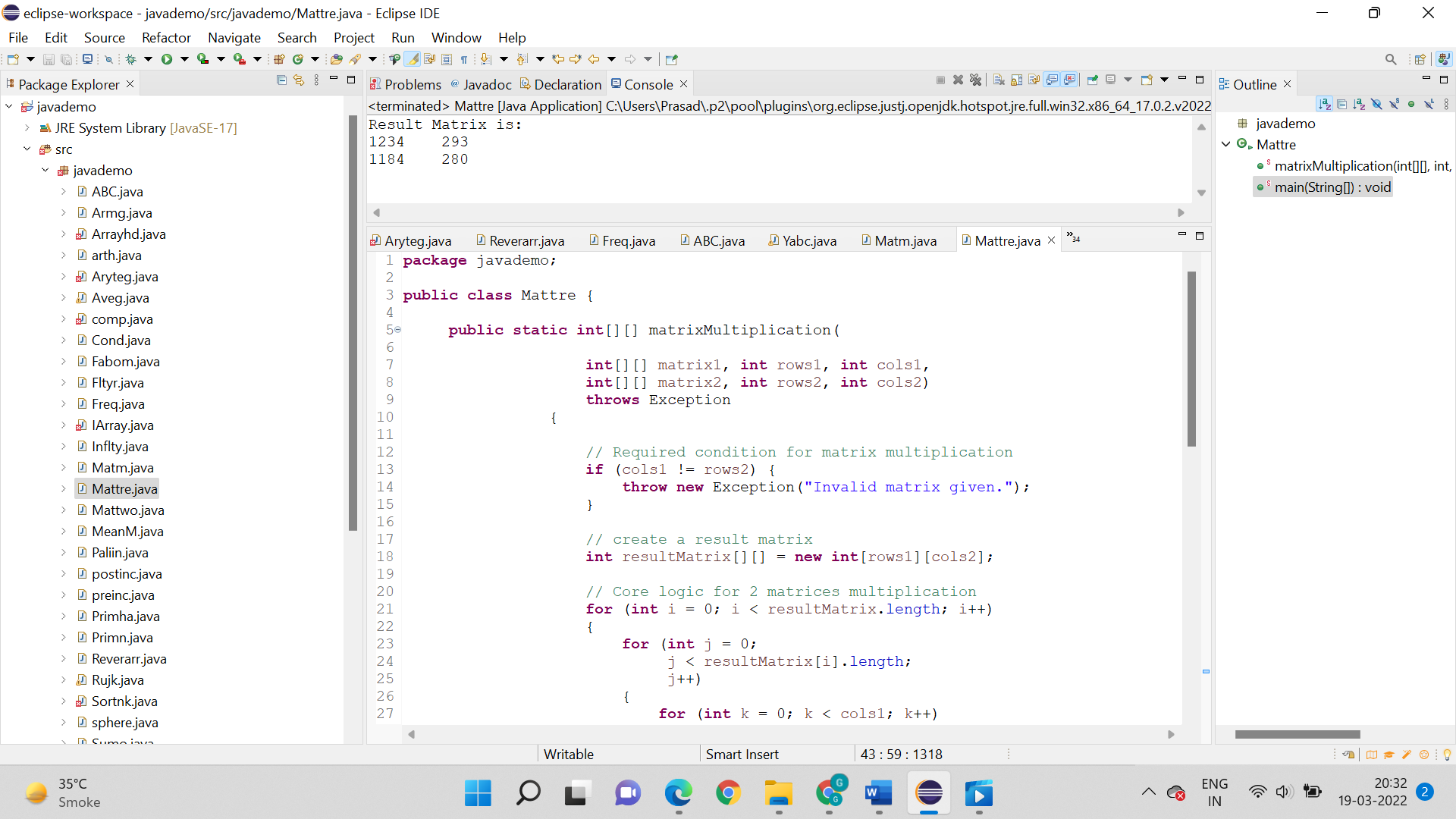


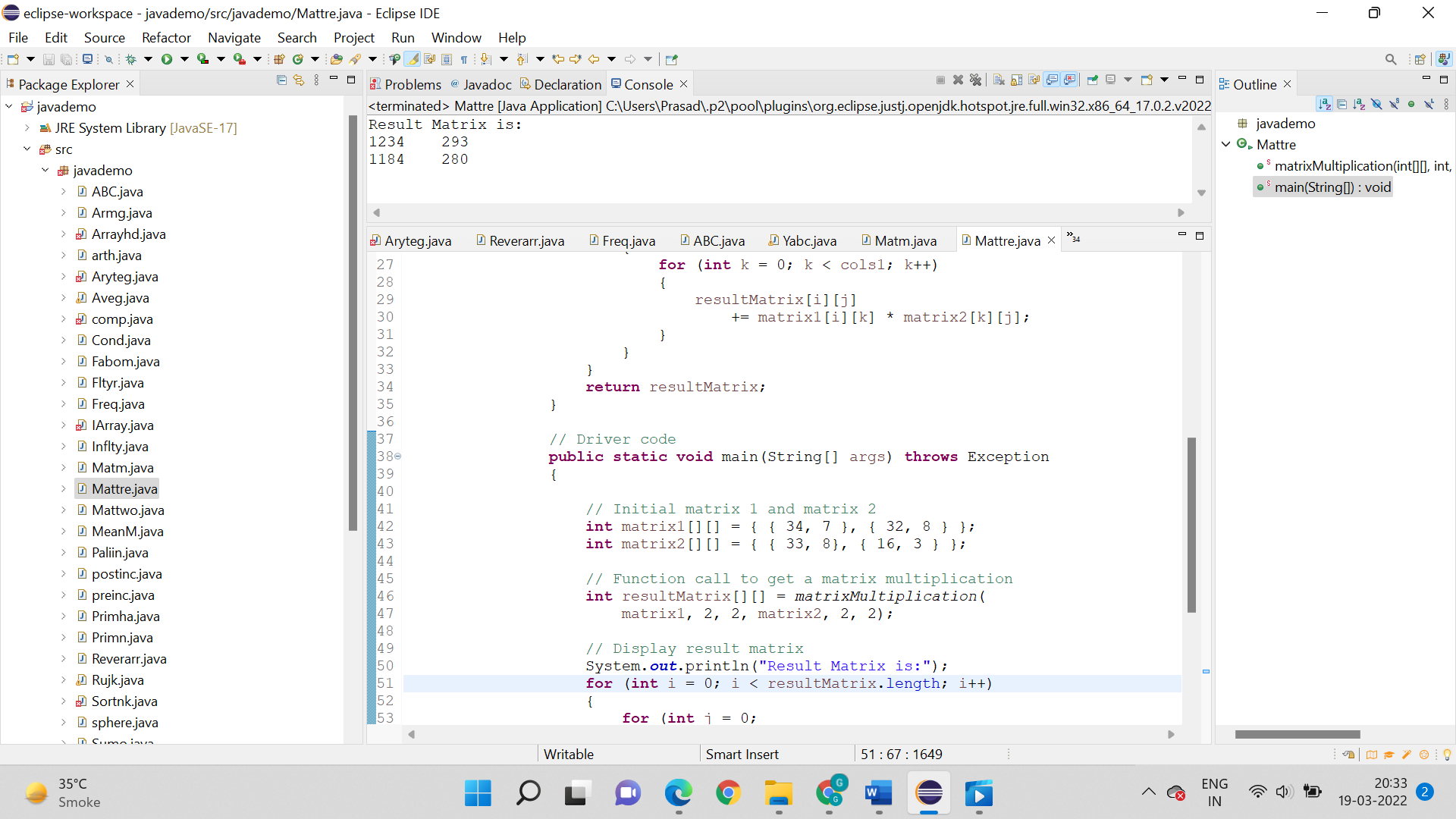
* 1. Matrix Multiplication

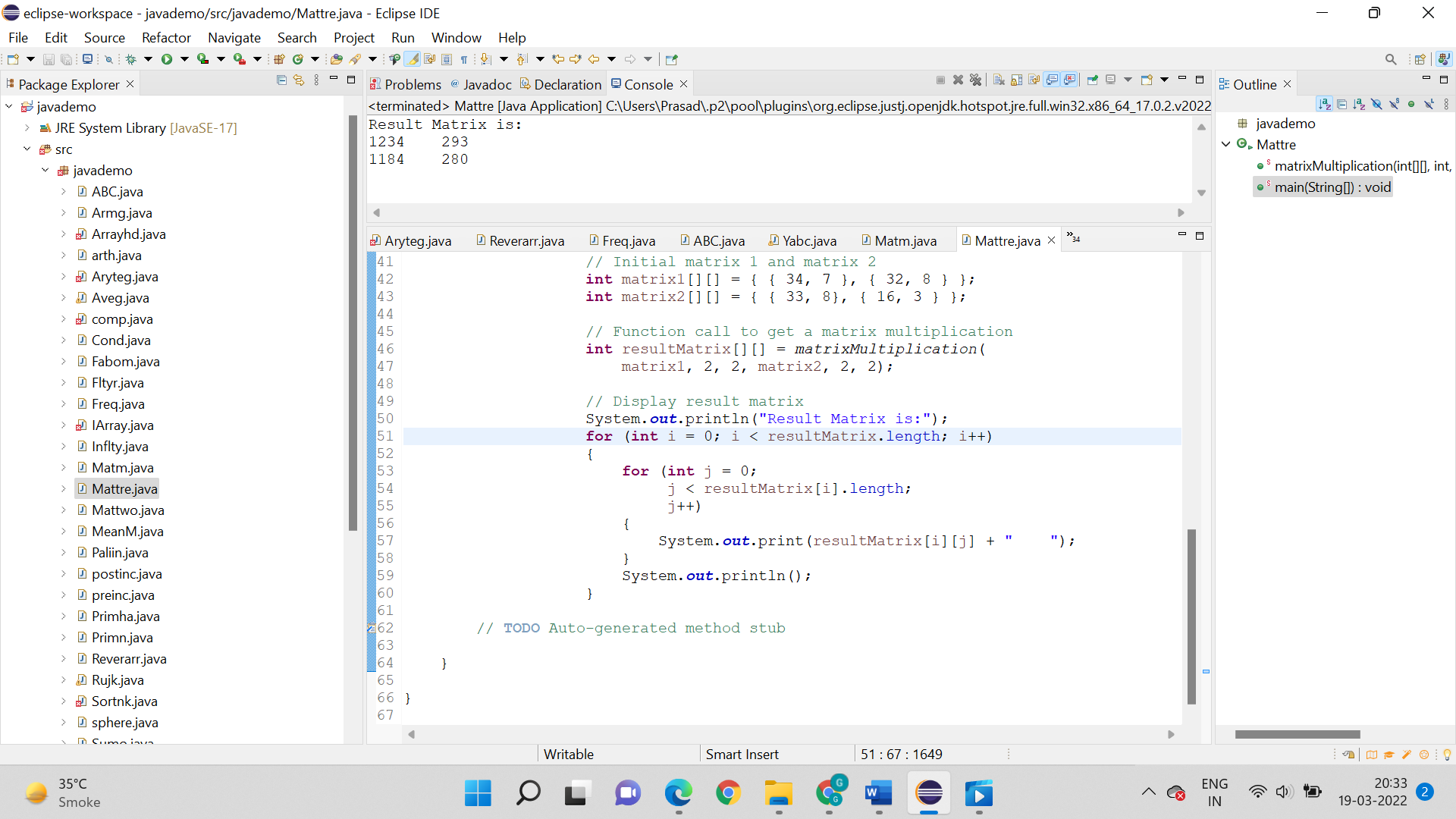
Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Required condition for matrix multiplication if (cols1 != rows2) { throw new Exception("Invalid matrix given.");
* Step 4: Create a result matrix
* Step 5: Core logic for 2 matrices multiplication for (int i = 0; i < resultMatrix.length; i++)
* Step 6: Function call to get a matrix multiplication
* Step 7: System.out.println is used to print matrix
* Step 8: Stop

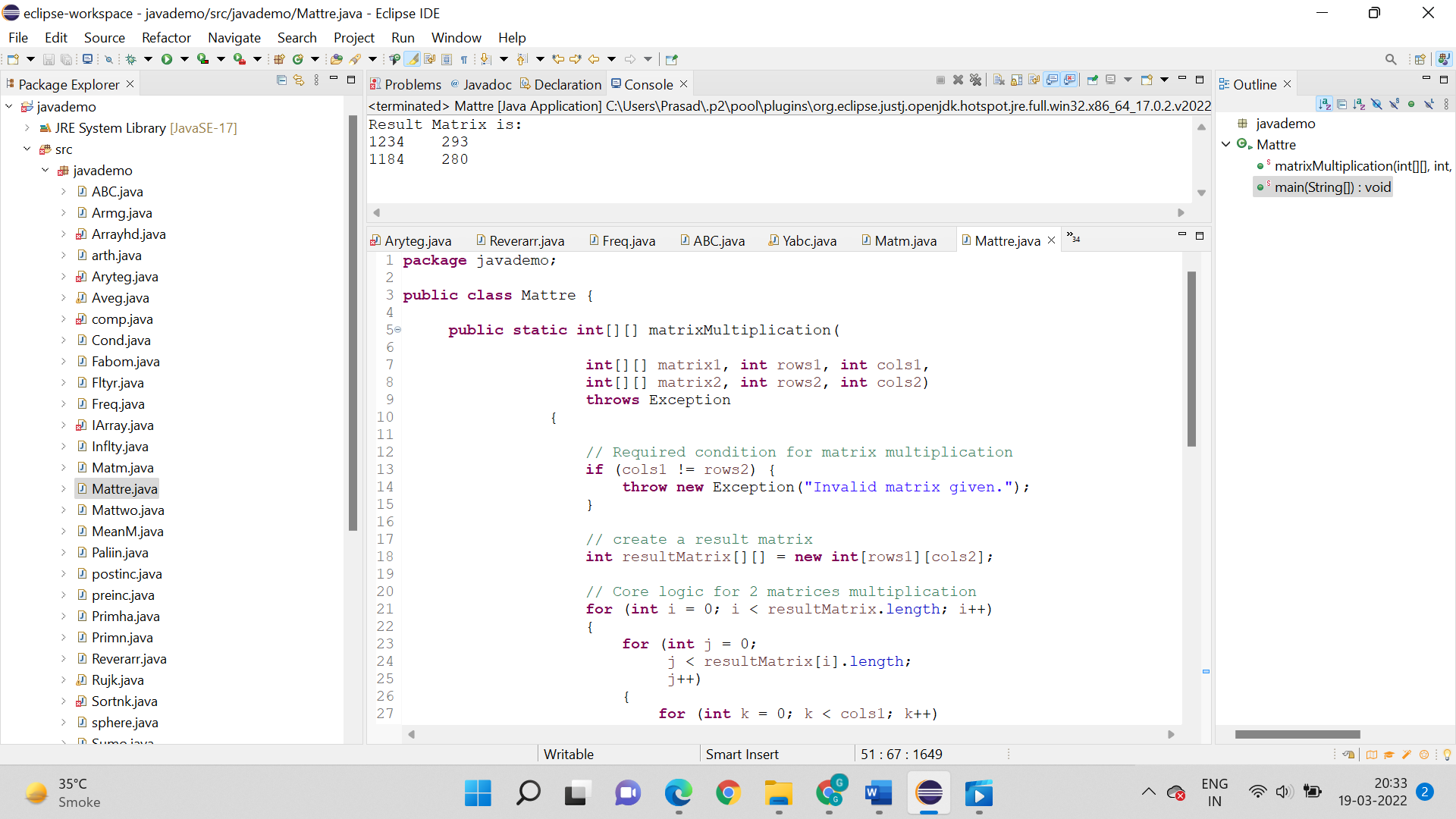
Code:







Output:

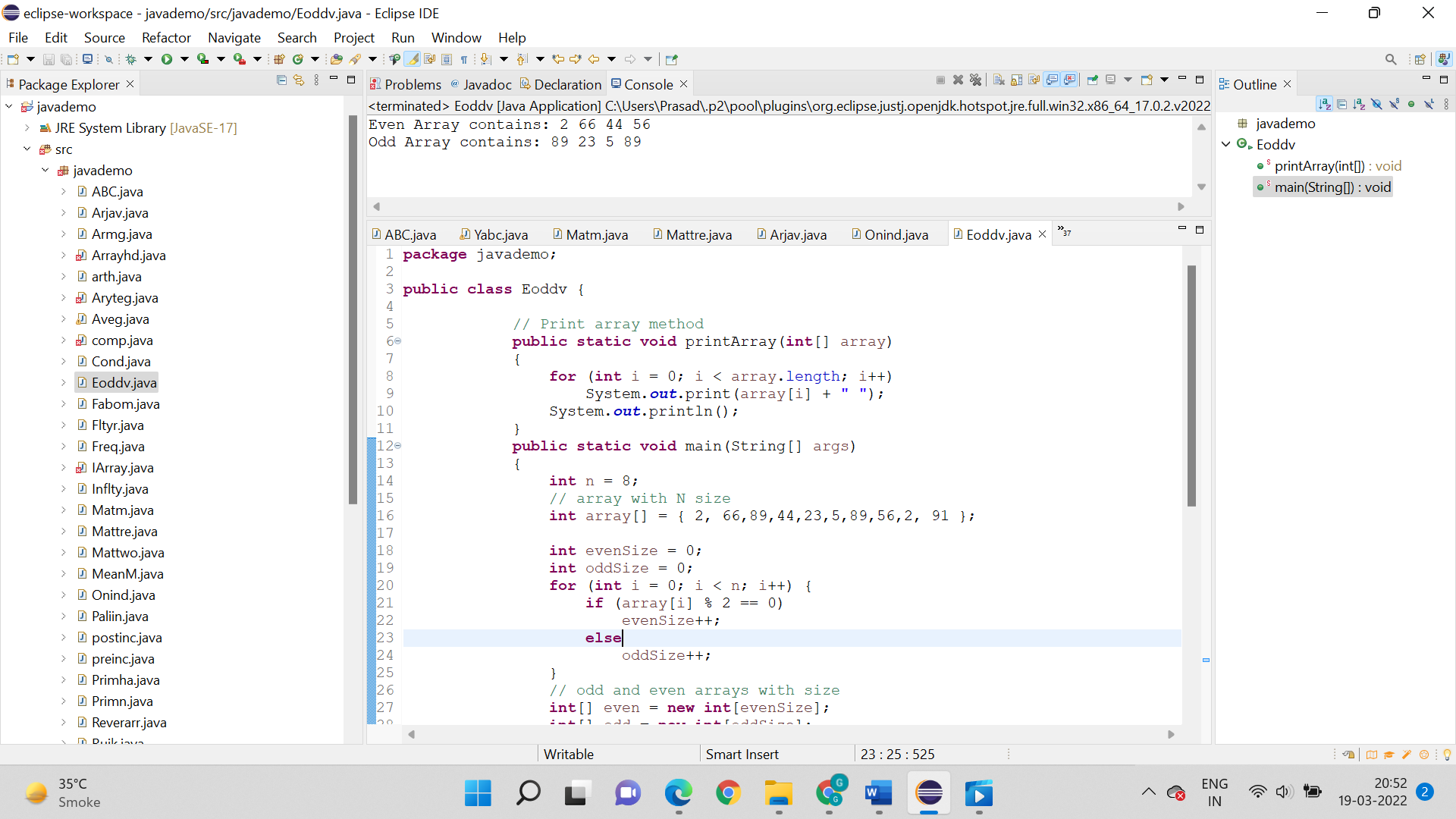


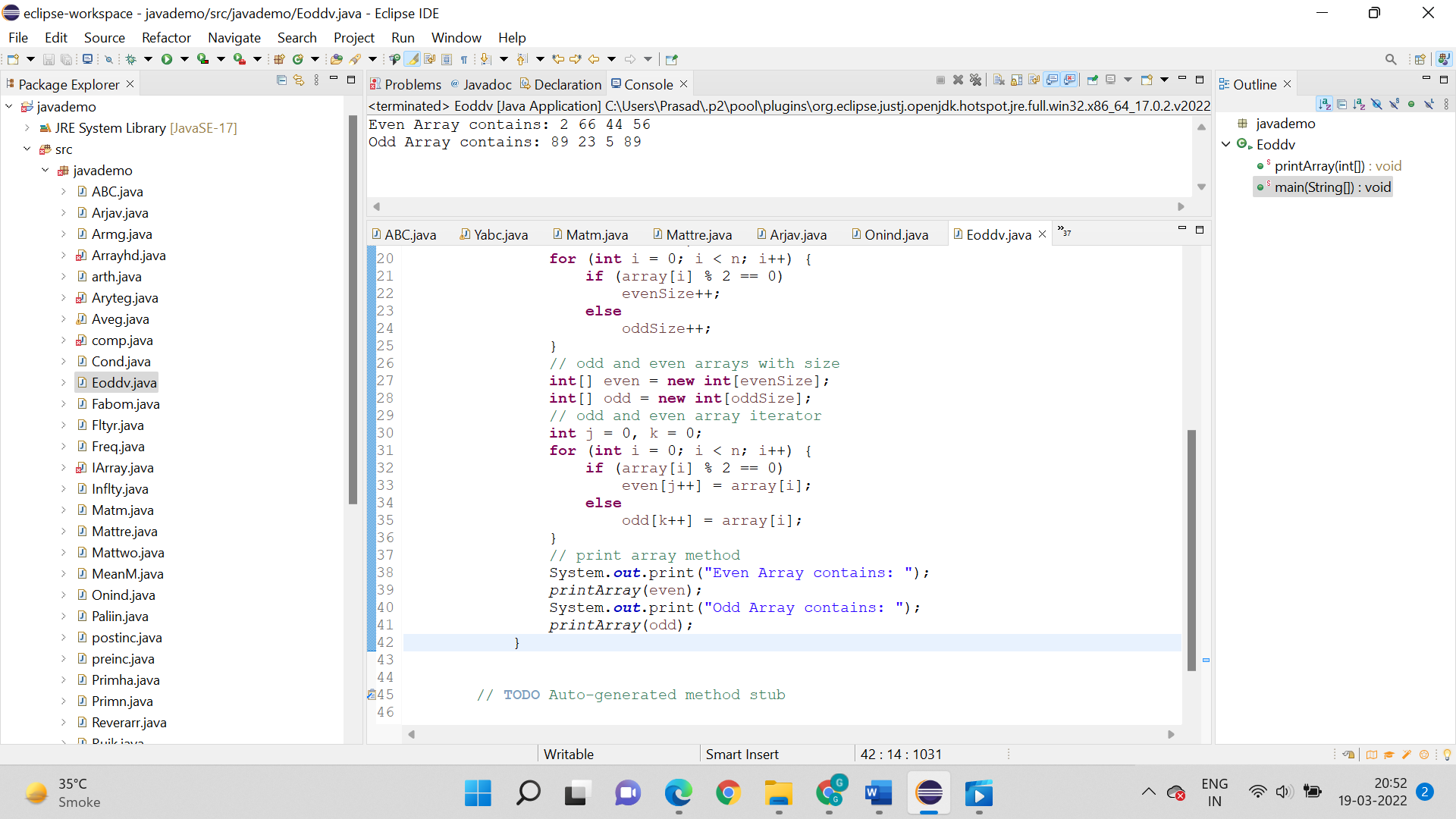
15. Write a program to determine even and odd numbers in a predefined 2-D array along with odd and even count. Print the array and odd and even numbers in an array form.

Logic:

* Step 1: Start
* Step 2: public static void main(String[] args)
* Step 3: Print array method
* Step 4: int n = 8 array with N size
* Step 5: Algorithm for odd and even array iterator and size
* Step 6: System.out.println is used to print what even and odd array is.
* Step 7: Stop

Code:





Output:

