# Level 1 Practice Programs

1. Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.

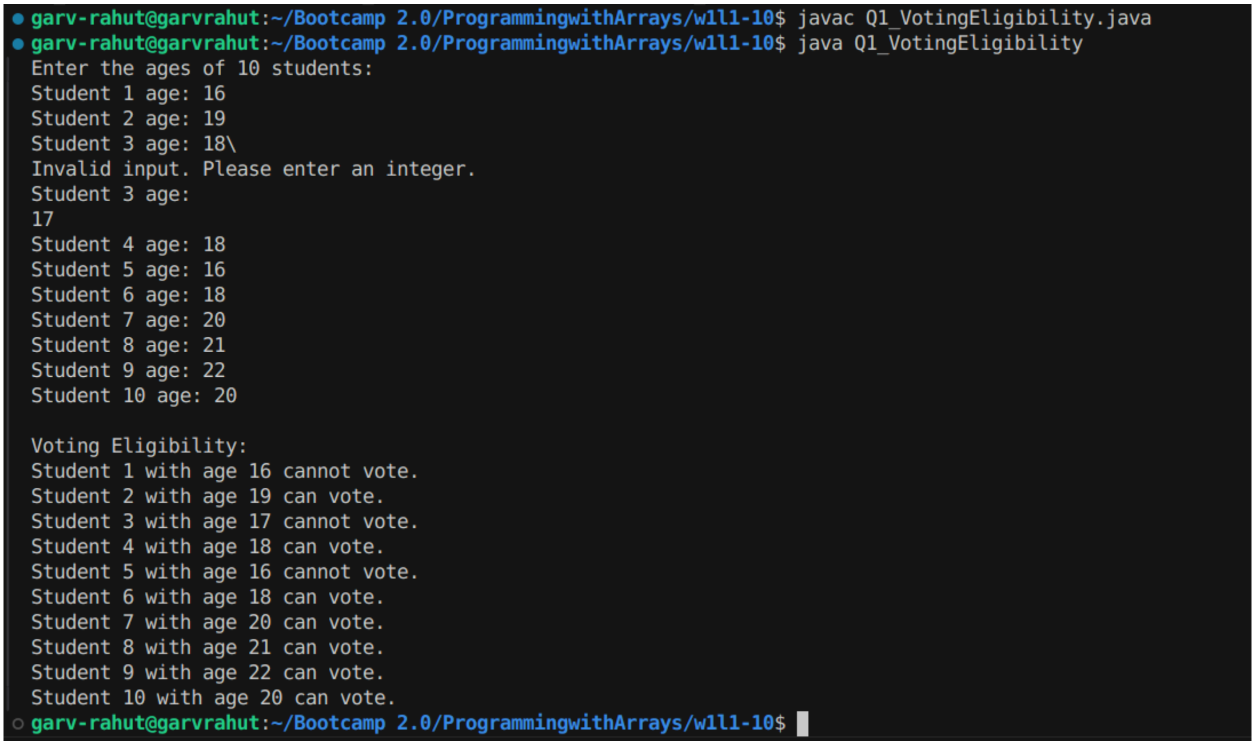
   **Hint =>**

   a. Define an array of 10 integer elements and take user input for the student's age.

   b. Loop through the array using the length property and for the element of the array check If the age is a negative number print an invalid age and if 18 or above, print The student with the age ___ can vote. Otherwise, print The student with the age ___ cannot vote.

```
import java.util.Scanner;


public class Q1_VotingEligibility {


   public static void main(String[] args) {

       Scanner input = new Scanner(System.in);


       int[] studentAges = new int[10]; // Array to store ages of 10 
students


       System.out.println("Enter the ages of 10 students:");


       // Input ages for 10 students

       for (int i = 0; i < studentAges.length; i++) {

           System.out.print("Student " + (i + 1) + " age: ");

           if (!input.hasNextInt()) {

               System.out.println("Invalid input. Please enter an 
integer.");

               input.next(); // Clear the invalid input

               i--; // Decrement i to re-enter the age for the same 
student
```

```
                continue; // Skip to the next iteration
            }
            studentAges[i] = input.nextInt();
        }

        System.out.println("\nVoting Eligibility:");

        // Check voting eligibility for each student
        for (int i = 0; i < studentAges.length; i++) {
            int age = studentAges[i];

            if (age < 0) {
                System.out.println("Student " + (i + 1) + ": Invalid age.");
            } else if (age >= 18) {
                System.out.println("Student " + (i + 1) + " with age " + age + " can vote.");
            } else {
                System.out.println("Student " + (i + 1) + " with age " + age + " cannot vote.");
            }
        }

        input.close();
    }
}
```



```
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q1_VotingEligibility.java
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q1_VotingEligibility
  Enter the ages of 10 students:
  Student 1 age: 16
  Student 2 age: 19
  Student 3 age: 18\
  Invalid input. Please enter an integer.
  Student 3 age:
  17
  Student 4 age: 18
  Student 5 age: 16
  Student 6 age: 18
  Student 7 age: 20
  Student 8 age: 21
  Student 9 age: 22
  Student 10 age: 20

  Voting Eligibility:
  Student 1 with age 16 cannot vote.
  Student 2 with age 19 can vote.
  Student 3 with age 17 cannot vote.
  Student 4 with age 18 can vote.
  Student 5 with age 16 cannot vote.
  Student 6 with age 18 can vote.
  Student 7 with age 20 can vote.
  Student 8 with age 21 can vote.
  Student 9 with age 22 can vote.
  Student 10 with age 20 can vote.
○ garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

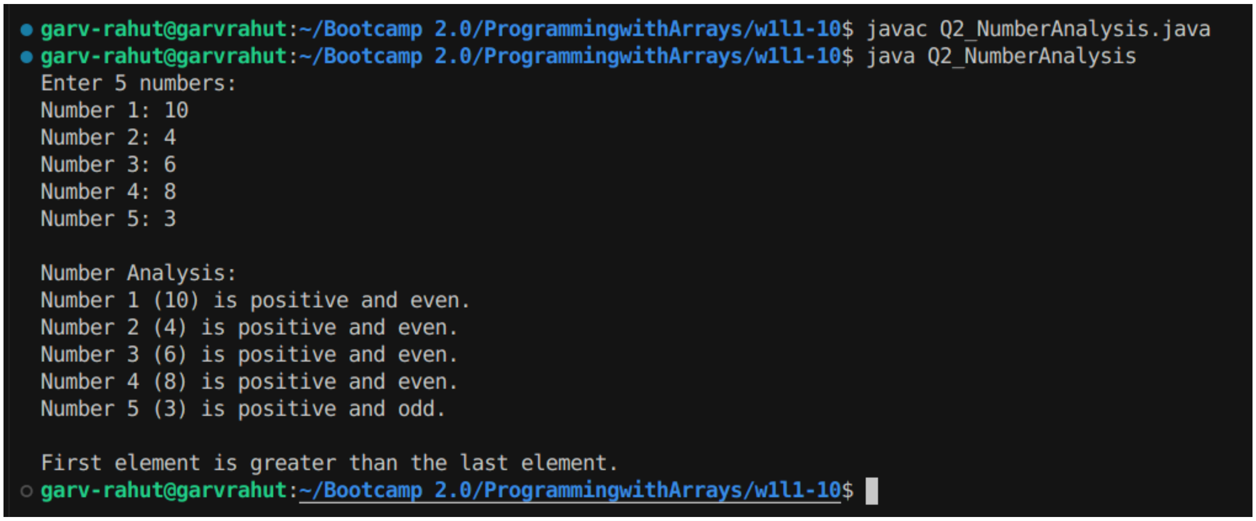
2. Write a program to take user input for 5 numbers and check whether a number is positive, negative, or zero. Further for positive numbers check if the number is even or odd. Finally compare the first and last elements of the array and display if they equal, greater or less

   **Hint =>**

   a. Define an integer array of 5 elements and get user input to store in the array.
   b. Loop through the array using the length If the number is positive, check for even or odd numbers and print accordingly
   c. If the number is negative, print negative. Else if the number is zero, print zero.
   d. Finally compare the first and last element of the array and display if they equal, greater or less

```
import java.util.Scanner;


public class Q2_NumberAnalysis {


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);


        int[] numbers = new int[5]; // Array to store 5 numbers
        System.out.println("Enter 5 numbers:");

        // Input 5 numbers
        for (int i = 0; i < numbers.length; i++) {
            System.out.print("Number " + (i + 1) + ": ");
            if (!input.hasNextInt()) {
                System.out.println("Invalid input. Please enter an
integer.");
                input.next(); // Clear invalid input
                i--; // Re-enter input for the same index
                continue;
            }
            numbers[i] = input.nextInt();
        }

        System.out.println("\nNumber Analysis:");

        // Analyze each number
        for (int i = 0; i < numbers.length; i++) {
            int num = numbers[i];

            if (num > 0) {
                if (num % 2 == 0) {
                    System.out.println("Number " + (i + 1) + " (" + num +
") is positive and even.");
                } else {
                    System.out.println("Number " + (i + 1) + " (" + num +
") is positive and odd.");
```

```
                }
            } else if (num < 0) {
                System.out.println("Number " + (i + 1) + " (" + num + ") is negative.");
            } else {
                System.out.println("Number " + (i + 1) + " (" + num + ") is zero.");
            }
        }

        // Compare first and last elements
        if (numbers[0] == numbers[numbers.length - 1]) {
            System.out.println("\nFirst and last elements are equal.");
        } else if (numbers[0] > numbers[numbers.length - 1]) {
            System.out.println("\nFirst element is greater than the last element.");
        } else {
            System.out.println("\nFirst element is less than the last element.");
        }

        input.close();
    }
}
```



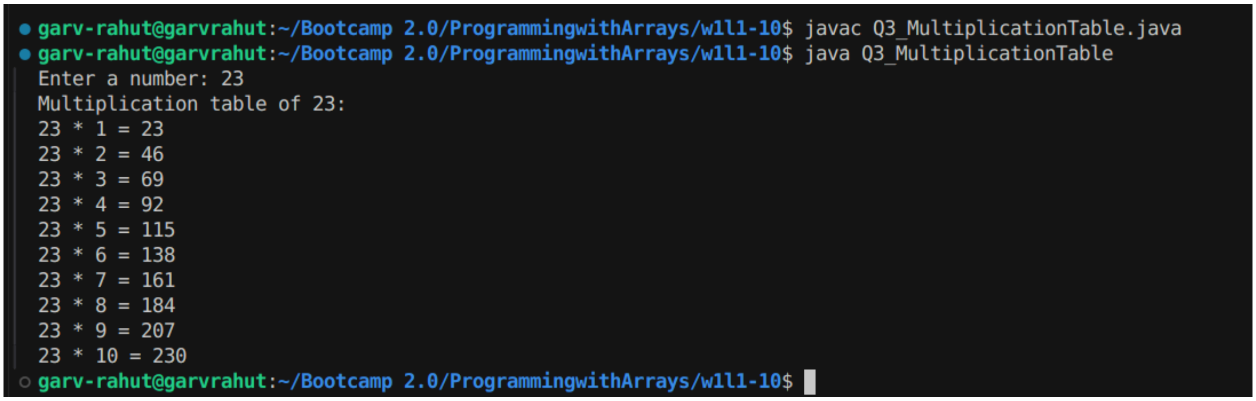
```
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q2_NumberAnalysis.java
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q2_NumberAnalysis
  Enter 5 numbers:
  Number 1: 10
  Number 2: 4
  Number 3: 6
  Number 4: 8
  Number 5: 3

  Number Analysis:
  Number 1 (10) is positive and even.
  Number 2 (4) is positive and even.
  Number 3 (6) is positive and even.
  Number 4 (8) is positive and even.
  Number 5 (3) is positive and odd.

  First element is greater than the last element.
○ garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

3. Create a program to print a multiplication table of a number.

   **Hint =>**

   a. Get an integer input and store it in the number variable. Also, define a integer array to store the results of multiplication from 1 to 10
   b. Run a loop from 1 to 10 and store the results in the multiplication table array
   c. Finally, display the result from the array in the format number * i = ___

```java
import java.util.Scanner;


public class Q3_MultiplicationTable {


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);


        System.out.print("Enter a number: ");


        if (!input.hasNextInt()) {

            System.out.println("Invalid input. Please enter an integer.");

            input.close();

            return;
        }

        int number = input.nextInt();

        if (number <= 0) {
            System.out.println("Please enter a positive integer.");
            input.close();
            return;
        }

        int[] multiplicationTable = new int[10]; // Array to store 
multiplication results

        // Calculate multiplication table
        for (int i = 0; i < 10; i++) {
            multiplicationTable[i] = number * (i + 1);
        }

        // Display multiplication table
        System.out.println("Multiplication table of " + number + ":");
        for (int i = 0; i < 10; i++) {
            System.out.println(number + " * " + (i + 1) + " = " + 
multiplicationTable[i]);
        }

        input.close();
    }
}
```



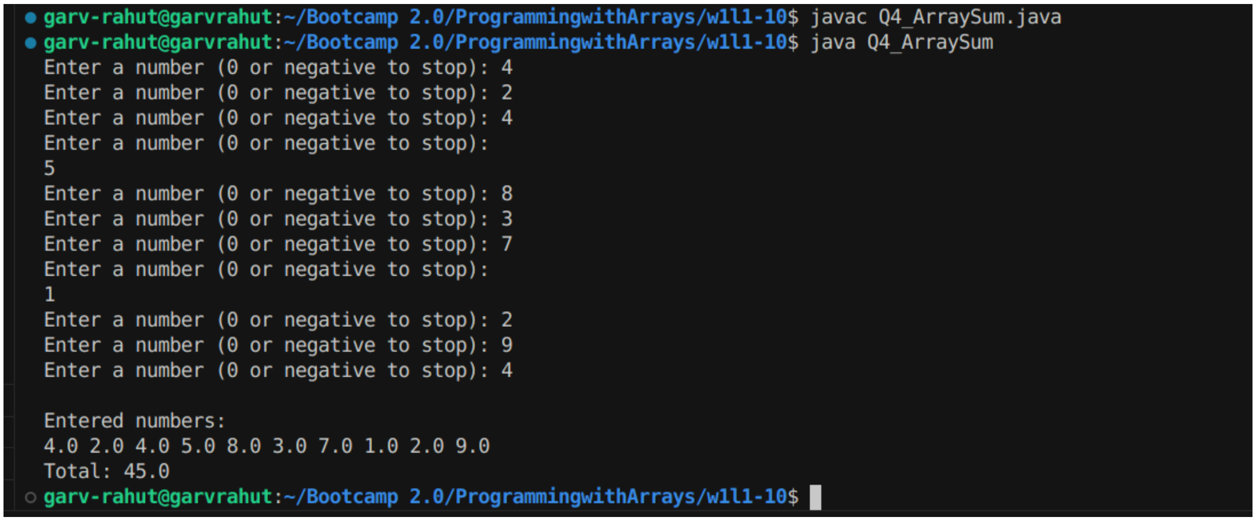
```
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q3_MultiplicationTable.java
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q3_MultiplicationTable
  Enter a number: 23
  Multiplication table of 23:
  23 * 1 = 23
  23 * 2 = 46
  23 * 3 = 69
  23 * 4 = 92
  23 * 5 = 115
  23 * 6 = 138
  23 * 7 = 161
  23 * 8 = 184
  23 * 9 = 207
  23 * 10 = 230
○ garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$
```

4. Write a program to store multiple values in an array up to a maximum of 10 or until the user enters a 0 or a negative number. Show all the numbers as well as the sum of all numbers

   **Hint =>**

   a. Create a variable to store an array of 10 elements of type double as well as a variable to store the total of type double initializes to 0.0. Also, the index variable is initialized to 0 for the array
   b. Use infinite while loop as in ***while (true)***
   c. Take the user entry and check if the user entered 0 or a negative number to break the loop
   d. Also, ***break*** from the loop if the index has a value of 10 as the array size is limited to 10.
   e. If the user entered a number other than 0 or a negative number inside the while loop then assign the number to the array element and increment the index value
   f. Take another ***for*** loop to get the values of each element and add it to the total
   g. Finally display the total value

```
import java.util.Scanner;


public class Q4_ArraySum {


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);


        double[] numbers = new double[10];

        double total = 0.0;

        int index = 0;
        while (true) {
            System.out.print("Enter a number (0 or negative to stop): ");

            if (!input.hasNextDouble()) {
                System.out.println("Invalid input. Please enter a
number.");
                input.next(); // Clear invalid input
                continue;
            }

            double number = input.nextDouble();

            if (number <= 0 || index == 10) {
                break;
            }

            numbers[index] = number;
            index++;
        }

        System.out.println("\nEntered numbers:");
        for (int i = 0; i < index; i++) {
            System.out.print(numbers[i] + " ");
            total += numbers[i];
        }

        System.out.println("\nTotal: " + total);
        input.close();

    }

}
```



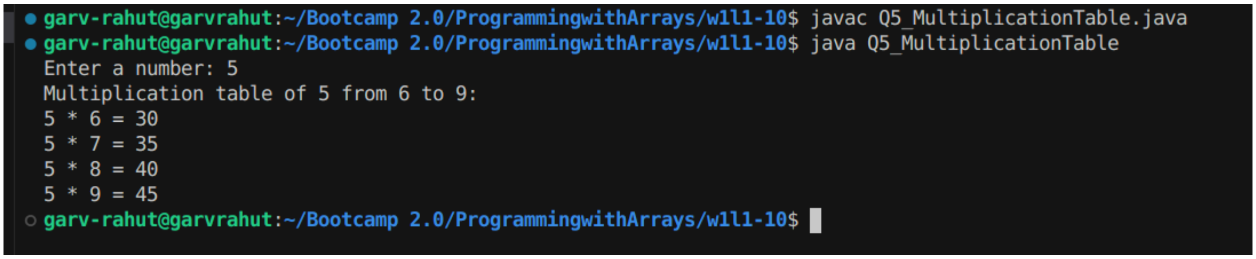
```
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q4_ArraySum.java
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q4_ArraySum
  Enter a number (0 or negative to stop): 4
  Enter a number (0 or negative to stop): 2
  Enter a number (0 or negative to stop): 4
  Enter a number (0 or negative to stop):
  5
  Enter a number (0 or negative to stop): 8
  Enter a number (0 or negative to stop): 3
  Enter a number (0 or negative to stop): 7
  Enter a number (0 or negative to stop):
  1
  Enter a number (0 or negative to stop): 2
  Enter a number (0 or negative to stop): 9
  Enter a number (0 or negative to stop): 4

  Entered numbers:
  4.0 2.0 4.0 5.0 8.0 3.0 7.0 1.0 2.0 9.0
  Total: 45.0
○ garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

5. Create a program to find the multiplication table of a number entered by the user from 6 to 9 and display the result

   **Hint =>**

   a. Take integer input and store it in the variable number as well as define an integer array to store the multiplication result in the variable multiplicationResult
   b. Using a for loop, find the multiplication table of numbers from 6 to 9 and save the result in the array
   c. Finally, display the result from the array in the format number * i = ___

```
import java.util.Scanner;


public class Q5_MultiplicationTable

{


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");

        if (!input.hasNextInt()) {
            System.out.println("Invalid input. Please enter an integer.");
            input.close();
            return;
        }

        int number = input.nextInt();

        if (number <= 0) {
            System.out.println("Please enter a positive integer.");
            input.close();
            return;
        }

        int[] multiplicationResult = new int[4]; // Array to store results
(6 to 9)

        // Calculate multiplication table from 6 to 9
        for (int i = 6; i <= 9; i++) {
            multiplicationResult[i - 6] = number * i; // Store results in
the array
        }

        // Display multiplication table
        System.out.println("Multiplication table of " + number + " from 6
to 9:");
        for (int i = 6; i <= 9; i++) {
            System.out.println(number + " * " + i + " = " +
multiplicationResult[i - 6]);
        }

        input.close();
    }
}
```



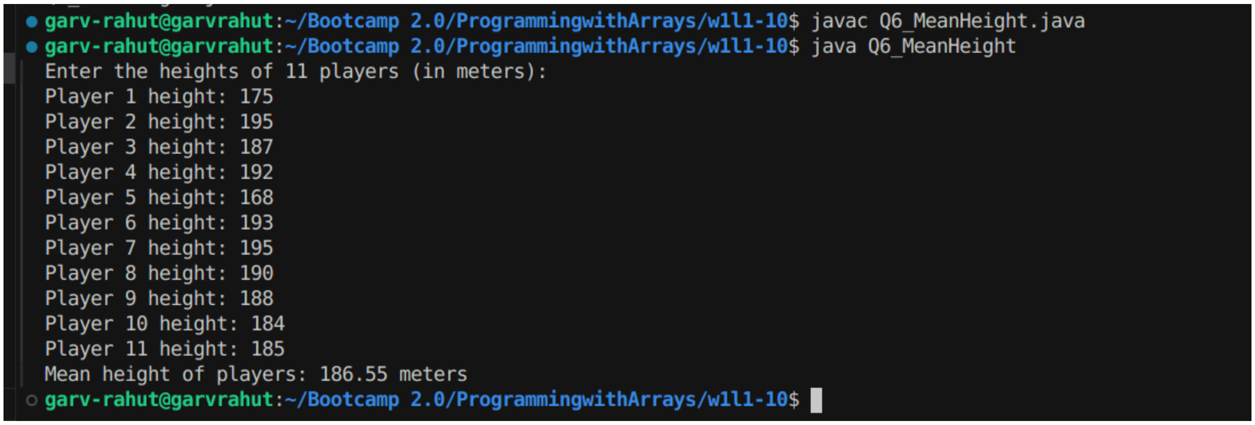
```
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q5_MultiplicationTable.java
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q5_MultiplicationTable
Enter a number: 5
Multiplication table of 5 from 6 to 9:
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$
```

6. Create a program to find the mean height of players present in a football team.

   **Hint =>**

   a. The formula to calculate the mean is: mean = sum of all elements / number of elements
   b. Create a double array named heights of size 11 and get input values from the user.
   c. Find the sum of all the elements present in the array.
   d. Divide the sum by 11 to find the mean height and print the mean height of the football team

```
import java.util.Scanner;

public class Q6_MeanHeight {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        double[] heights = new double[11];
        double sum = 0;
        System.out.println("Enter the heights of 11 players (in meters):");

        for (int i = 0; i < heights.length; i++) {
            System.out.print("Player " + (i + 1) + " height: ");
            if (!input.hasNextDouble()) {
                System.out.println("Invalid input. Please enter a valid
height (in meters).");
                input.next(); // Clear invalid input
                i--; // Allow re-entry for this index.
                continue;
            }
            heights[i] = input.nextDouble();
            sum += heights[i];
        }

        double meanHeight = sum / heights.length;

        System.out.printf("Mean height of players: %.2f meters\n",
meanHeight);

        input.close();
    }
}
```



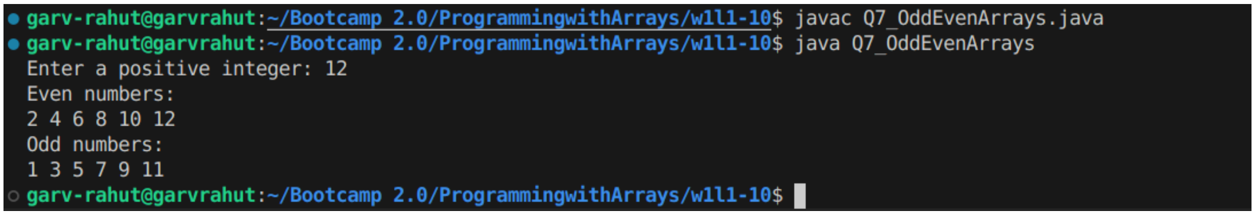
```
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q6_MeanHeight.java
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q6_MeanHeight
Enter the heights of 11 players (in meters):
Player 1 height: 175
Player 2 height: 195
Player 3 height: 187
Player 4 height: 192
Player 5 height: 168
Player 6 height: 193
Player 7 height: 195
Player 8 height: 190
Player 9 height: 188
Player 10 height: 184
Player 11 height: 185
Mean height of players: 186.55 meters
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

7. Create a program to save odd and even numbers into odd and even arrays between 1 to the number entered by the user. Finally, print the odd and even numbers array

**Hint =>**

a. Get an integer input from the user, assign it to a variable ***number,*** and check for Natural Number. If not a natural number then print an error and exit the program
b. Create an integer array for even and odd numbers with `size = number / 2 + 1`
c. Create index variables for odd and even numbers and initialize them to zero
d. Using a for loop, iterate from 1 to the number, and in each iteration of the loop, save the odd or even number into the corresponding array
e. Finally, print the odd and even numbers array using the odd and even index

```
import java.util.Scanner;


public class Q7_OddEvenArrays {


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);


        System.out.print("Enter a positive integer: ");


        if (!input.hasNextInt()) {

            System.out.println("Invalid input. Please enter an integer.");

            input.close();

            return;

        }
        int number = input.nextInt();

        if (number <= 0) {
            System.out.println("Please enter a positive integer.");
            input.close();
            return;
        }

        int[] evenNumbers = new int[number / 2 + 1]; // Maximum possible even numbers
        int[] oddNumbers = new int[number / 2 + 1]; // Maximum possible odd numbers
        int evenIndex = 0;
        int oddIndex = 0;

        for (int i = 1; i <= number; i++) {
            if (i % 2 == 0) {
                evenNumbers[evenIndex] = i;
                evenIndex++;
            } else {
                oddNumbers[oddIndex] = i;
                oddIndex++;
            }
        }

        System.out.println("Even numbers:");
        for (int i = 0; i < evenIndex; i++) {
            System.out.print(evenNumbers[i] + " ");
        }
        System.out.println();

        System.out.println("Odd numbers:");
        for (int i = 0; i < oddIndex; i++) {
            System.out.print(oddNumbers[i] + " ");
        }
        System.out.println();

        input.close();
    }
}
```



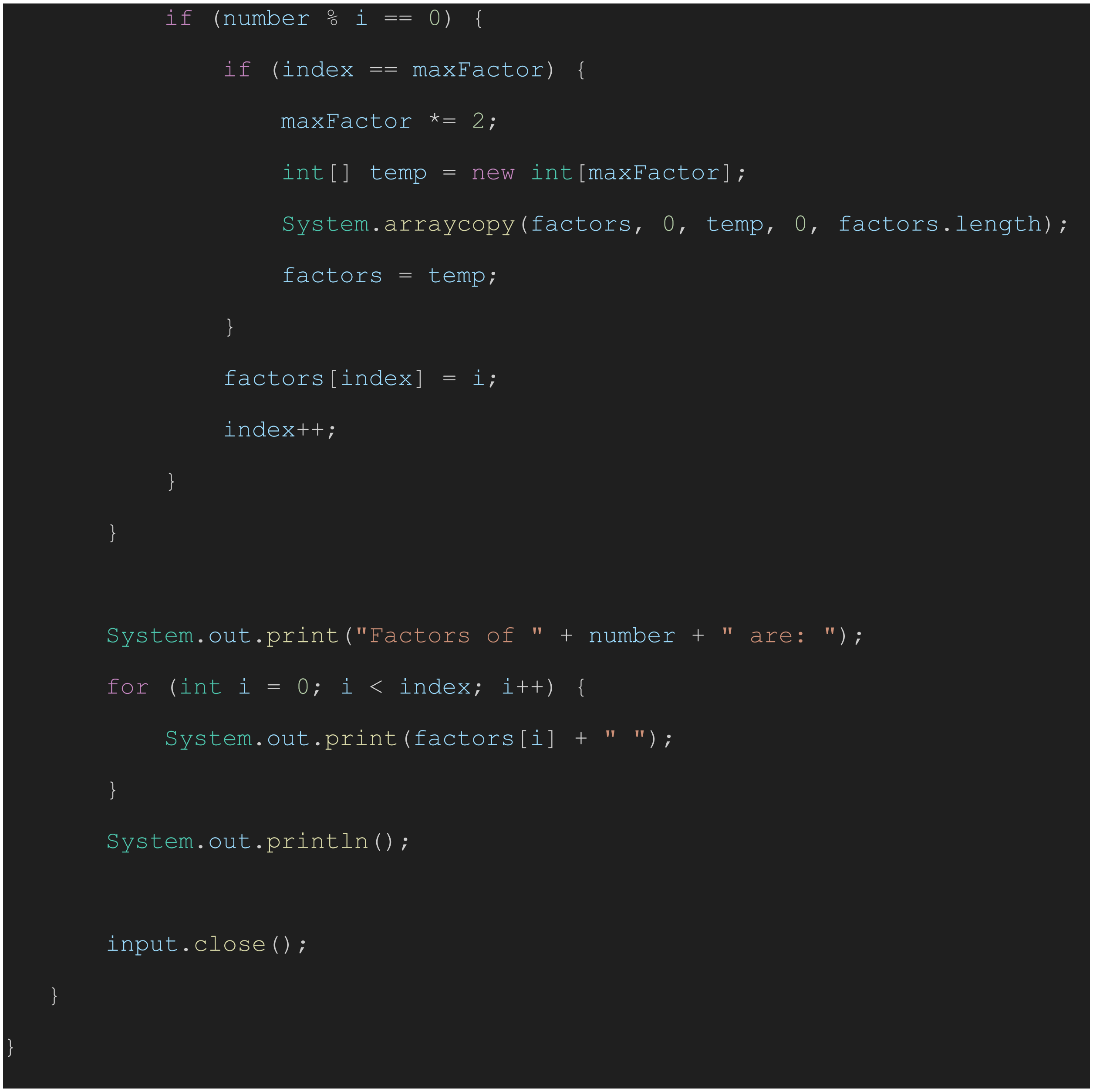
```
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q7_OddEvenArrays.java
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q7_OddEvenArrays
Enter a positive integer: 12
Even numbers:
2 4 6 8 10 12
Odd numbers:
1 3 5 7 9 11
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$
```

8. Create a program to find the factors of a number taken as user input, store the factors in an array, and display the factors

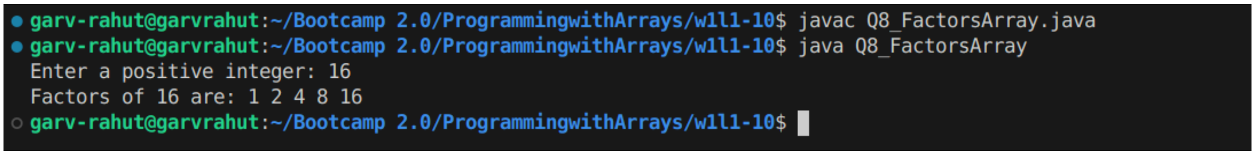
   **Hint =>**

   a. Take the input for a number
   b. Find the factors of the number and save them in an array. For this create integer variable maxFactor and initialize to 10, factors array of size maxFactor and index variable to reflect the index of the array.
   c. To find factors loop through the numbers from 1 to the number, find the factors, and add them to the array element by incrementing the index. If the index is equal to maxIndex, then need factors array to store more elements
   d. To store more elements, reset the maxIndex to twice its size, use the temp array to store the elements from the factors array, and eventually assign the factors array to the temp array
   e. Finally, Display the factors of the number

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

public class Q8_FactorsArray {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a positive integer: ");

        if (!input.hasNextInt()) {
            System.out.println("Invalid input. Please enter an integer.");
            input.close();
            return;
        }

        int number = input.nextInt();

        if (number <= 0) {
            System.out.println("Please enter a positive integer.");
            input.close();
            return;
        }

        int maxFactor = 10;
        int[] factors = new int[maxFactor];
        int index = 0;

        for (int i = 1; i <= number; i++) {
```



```
            if (number % i == 0) {
                if (index == maxFactor) {
                    maxFactor *= 2;
                    int[] temp = new int[maxFactor];
                    System.arraycopy(factors, 0, temp, 0, factors.length);
                    factors = temp;
                }
                factors[index] = i;
                index++;
            }
        }

        System.out.print("Factors of " + number + " are: ");
        for (int i = 0; i < index; i++) {
            System.out.print(factors[i] + " ");
        }
        System.out.println();

        input.close();
    }
}
```



```
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q8_FactorsArray.java
● garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q8_FactorsArray
  Enter a positive integer: 16
  Factors of 16 are: 1 2 4 8 16
○ garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

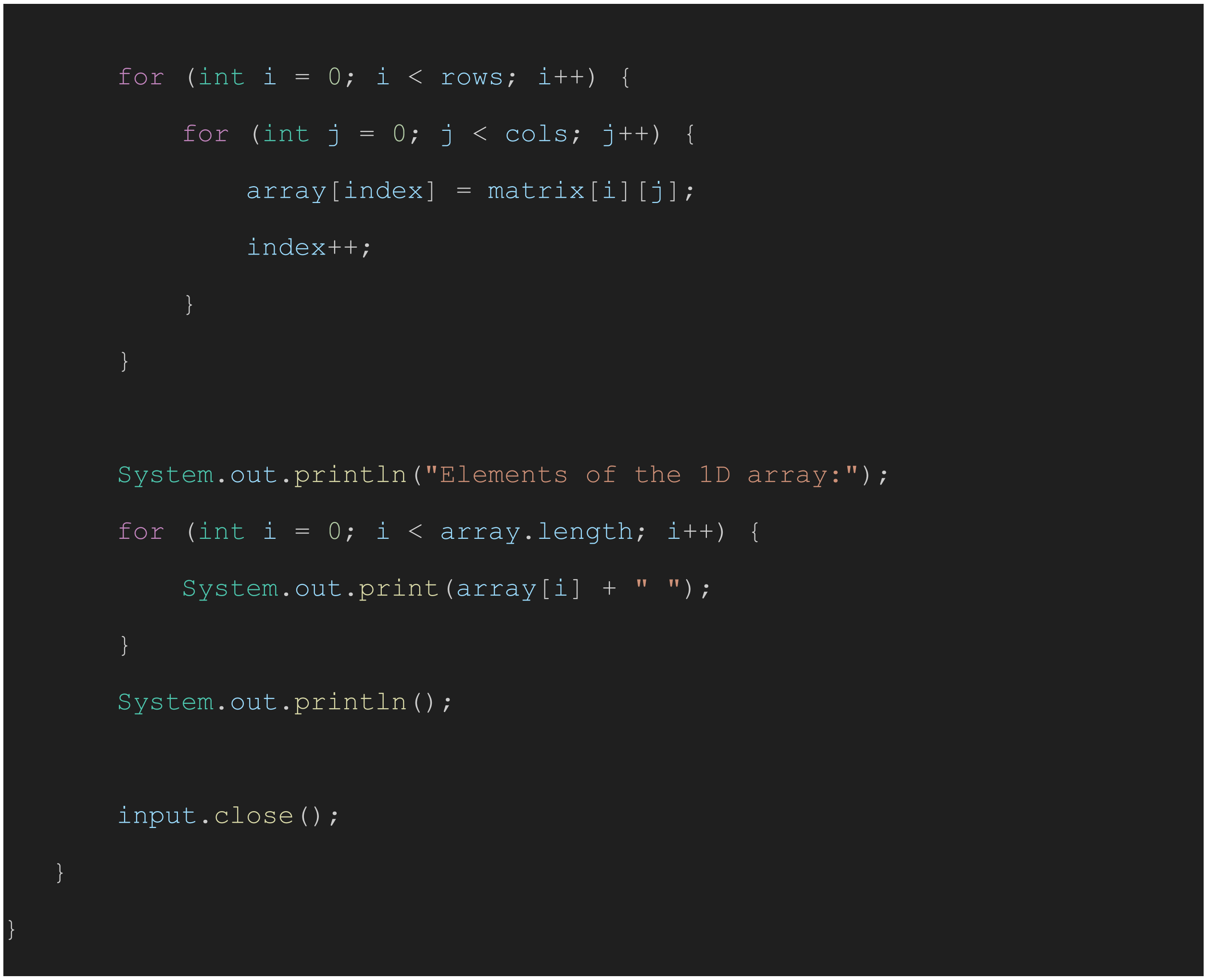
9. Working with Multi-Dimensional Arrays. Write a Java program to create a 2D Array and Copy the 2D Array into a single dimension array

   **Hint =>**

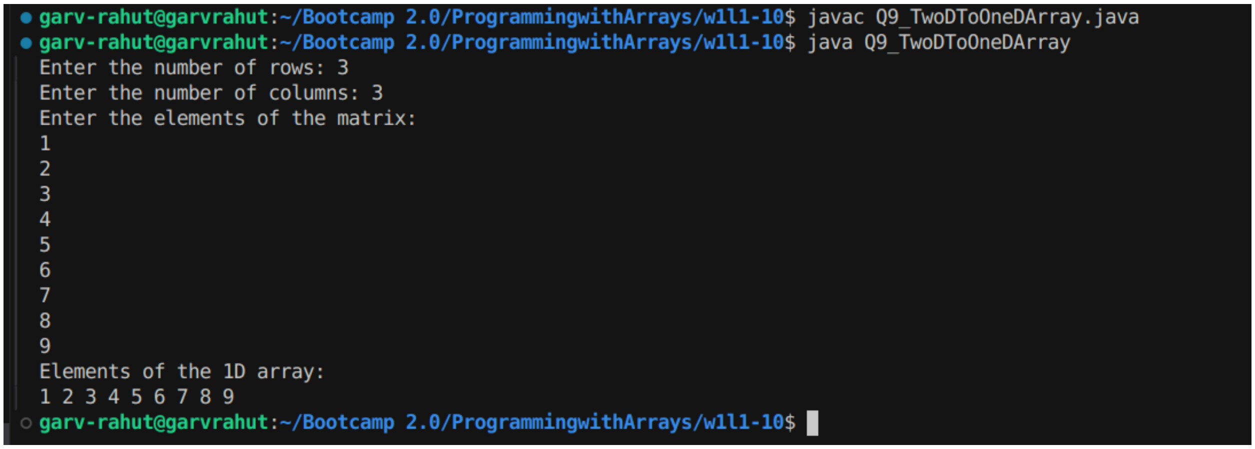
   a. Take user input for rows and columns, create a 2D array (Matrix), and take the user input

b. Copy the elements of the matrix to a 1D array. For this create a 1D array of size rows*columns as in `int[] array = new int[rows * columns];`
c. Define the index variable and Loop through the 2D array. Copy every element of the 2D array into the 1D array and increment the index
d. Note: For looping through the 2D array, you will need Nested for loop, Outer for loop for rows, and the inner for loops to access each element

```
import java.util.Scanner;

public class Q9_TwoDToOneDArray {

   public static void main(String[] args) {
       Scanner input = new Scanner(System.in);

       System.out.print("Enter the number of rows: ");
       int rows = input.nextInt();

       System.out.print("Enter the number of columns: ");
       int cols = input.nextInt();

       int[][] matrix = new int[rows][cols];

       System.out.println("Enter the elements of the matrix:");
       for (int i = 0; i < rows; i++) {
           for (int j = 0; j < cols; j++) {
               matrix[i][j] = input.nextInt();
           }
       }

       int[] array = new int[rows * cols];
       int index = 0;
```



```
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                array[index] = matrix[i][j];
                index++;
            }
        }

        System.out.println("Elements of the 1D array:");
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();

        input.close();
    }
}
```



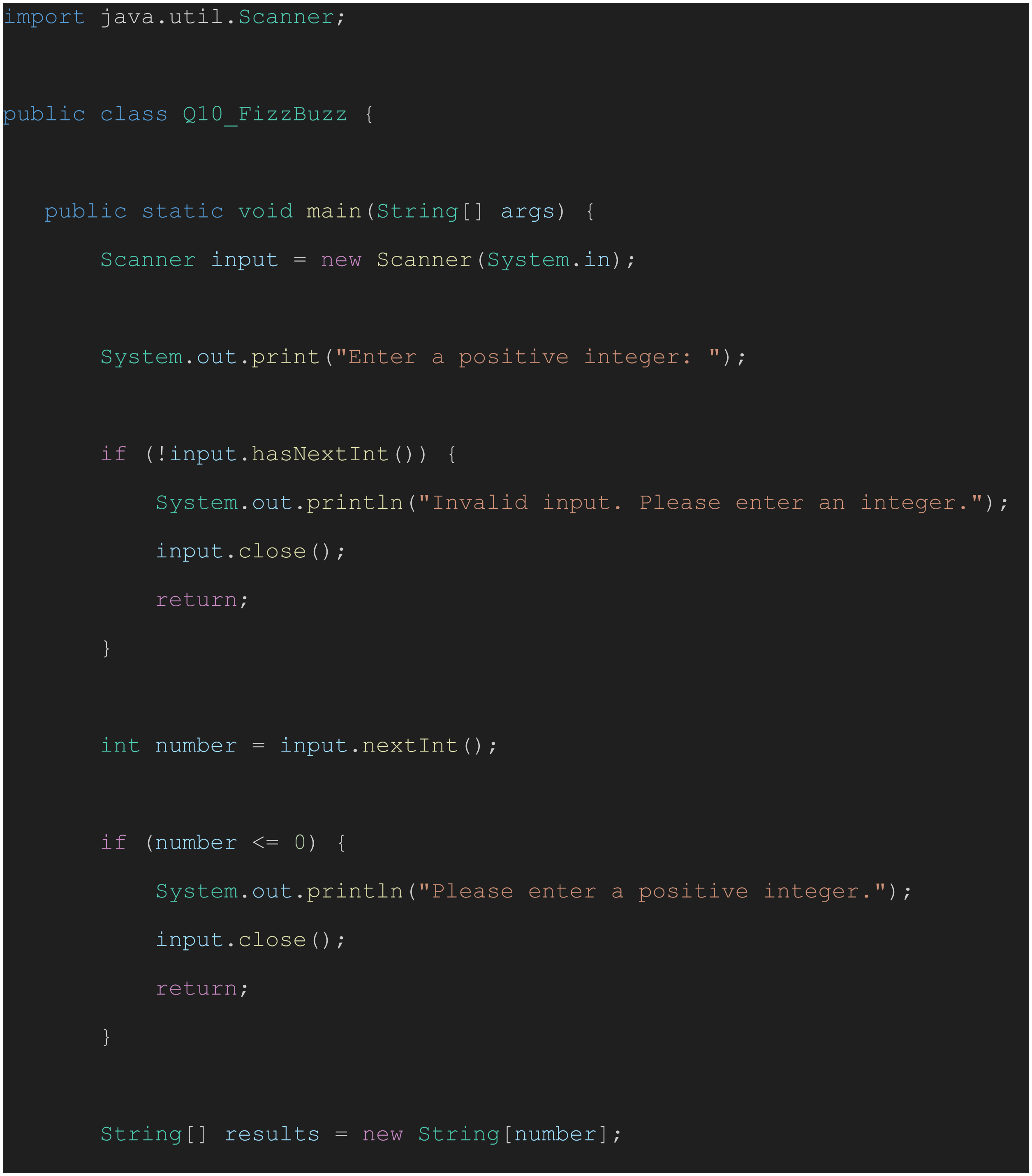
```
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q9_TwoDToOneDArray.java
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q9_TwoDToOneDArray
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
1
2
3
4
5
6
7
8
9
Elements of the 1D array:
1 2 3 4 5 6 7 8 9
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ 
```

10. Write a program FizzBuzz, take a number as user input and if it is a positive integer loop from 0 to the number and save the number, but for multiples of 3 save "Fizz" instead of the number, for multiples of 5 save "Buzz", and for multiples of both save "FizzBuzz". Finally,

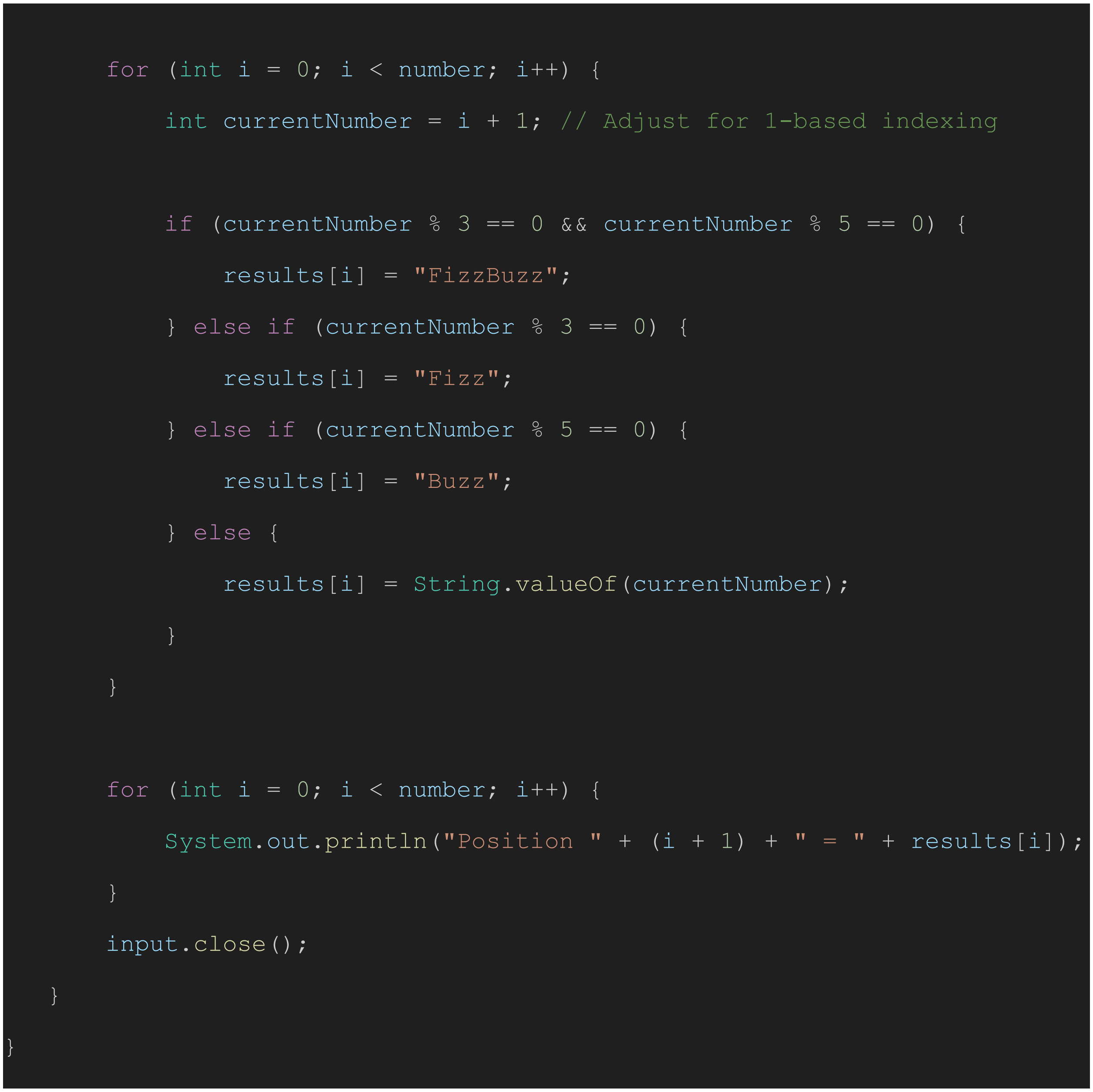
print the array results for each index position in the format Position 1 = 1, …, Position 3 = Fizz,...

**Hint =>**

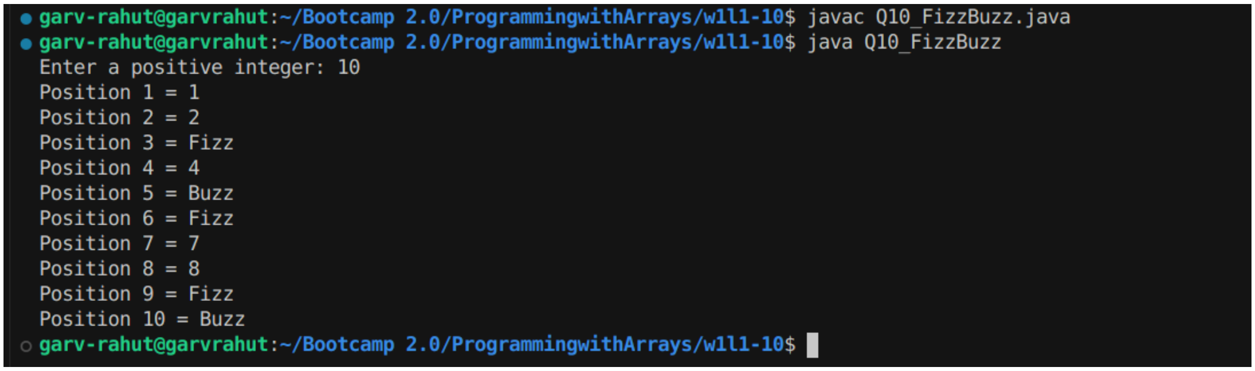
a. Create a String Array to save the results and
b. Finally, loop again to show the results of the array based on the index position



```java
import java.util.Scanner;


public class Q10_FizzBuzz {


    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);


        System.out.print("Enter a positive integer: ");


        if (!input.hasNextInt()) {

            System.out.println("Invalid input. Please enter an integer.");

            input.close();

            return;

        }


        int number = input.nextInt();


        if (number <= 0) {

            System.out.println("Please enter a positive integer.");

            input.close();

            return;

        }


        String[] results = new String[number];
```



```java
        for (int i = 0; i < number; i++) {
            int currentNumber = i + 1; // Adjust for 1-based indexing

            if (currentNumber % 3 == 0 && currentNumber % 5 == 0) {
                results[i] = "FizzBuzz";
            } else if (currentNumber % 3 == 0) {
                results[i] = "Fizz";
            } else if (currentNumber % 5 == 0) {
                results[i] = "Buzz";
            } else {
                results[i] = String.valueOf(currentNumber);
            }
        }

        for (int i = 0; i < number; i++) {
            System.out.println("Position " + (i + 1) + " = " + results[i]);
        }
        input.close();
    }
}
```



```
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ javac Q10_FizzBuzz.java
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$ java Q10_FizzBuzz
Enter a positive integer: 10
Position 1 = 1
Position 2 = 2
Position 3 = Fizz
Position 4 = 4
Position 5 = Buzz
Position 6 = Fizz
Position 7 = 7
Position 8 = 8
Position 9 = Fizz
Position 10 = Buzz
garv-rahut@garvrahut:~/Bootcamp 2.0/ProgrammingwithArrays/w1l1-10$
```