**PROGRAM – CAN GIVE VOTE**

**Q1:** 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 =>**

1. Define an array of 10 integer elements and take user input for the student's age.
2. 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.

**CODE:**

import java.util.\*;

public class CanStudentsVote {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

int ages[] = new int[10];

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

System.out.println("ENTER THE AGE FOR THE " + (i + 1) + " STUDENT : ");

ages[i] = sc.nextInt();

}

String r = "";

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

if (ages[i] < 0) {

System.err.println("INVALID AGE!");

} else {

r = (ages[i] > 18) ? "THE STUDENT WITH AGE " + ages[i] + " CAN GIVE VOTE" : "THE STUDENT WITH AGE " + ages[i] + " CANNOT GIVE VOTE";

System.out.println(r);

}

}

} catch (InputMismatchException e) {

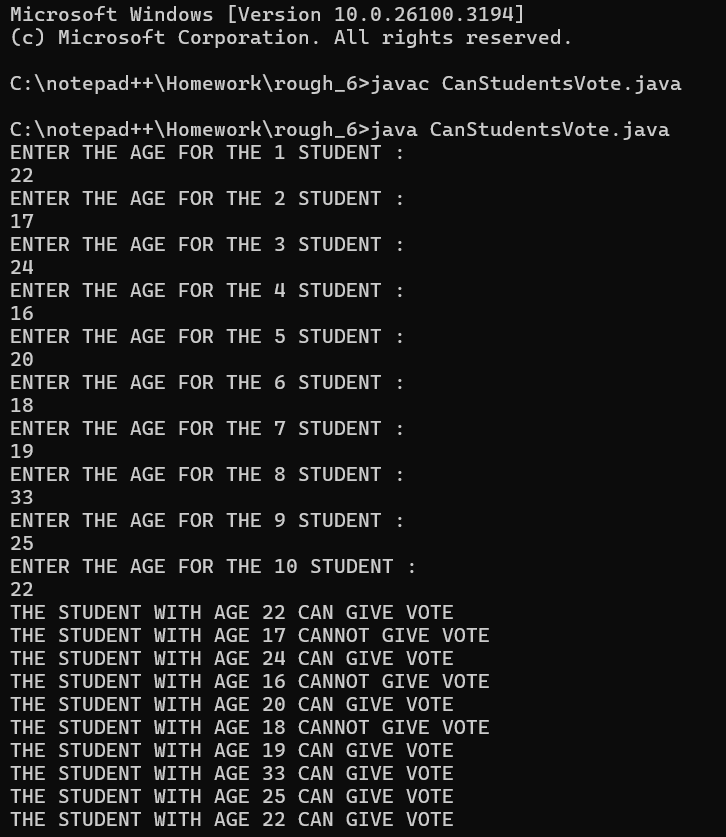
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – CHECK WHETHER NUMBER IS POSITIVE, NEGATIVE OR ZERO**

**Q2:** 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 =>**

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

**CODE:**

import java.util.\*;

public class NegPosiZeroEvenOddEqualGreaterLess {

public static void input(int num[], Scanner sc) {

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

num[i] = sc.nextInt();

}

}

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

int num[] = new int[5];

System.out.println("ENTER THE 5 NUMBERS : ");

input(num, sc);

String r = "";

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

if (num[i] > 0) {

System.out.println(num[i] + " IS A POSITIVE ELEMENT");

r = (num[i] % 2 == 0) ? num[i] + " IS AN EVEN ELEMENT" : num[i] + " IS AN ODD ELEMENT";

System.out.println(r);

} else if (num[i] < 0) {

System.out.println(num[i] + " IS A NEGATIVE ELEMENT");

} else {

System.out.println(num[i] + " IS A ZERO ELEMENT");

}

}

String p = (num[0] == num[4]) ? "THE FIRST & LAST ELEMENTS ARE EQUAL WITH EACH OTHERS" : (num[0] > num[4]) ? "THE FIRST ELEMENT IS GREATER THEN THE LAST ELEMENT" : "THE FIRST ELEMENT IS LESSER THEN THE LAST ELEMENT";

System.out.println(p);

} catch (InputMismatchException e) {

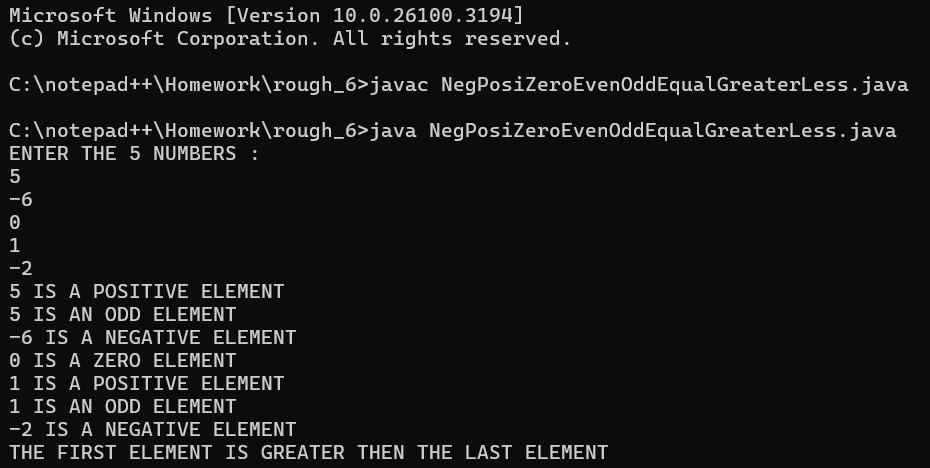
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – MULTIPLICATION TABLE**

**Q3:**  Create a program to print a multiplication table of a number.

**Hint =>**

1. 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
2. Run a loop from 1 to 10 and store the results in the multiplication table array
3. Finally, display the result from the array in the format number \* i = \_\_\_

**CODE:**

import java.util.\*;

public class MultiplicationTableArray {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER THE NUMBER : ");

int n = sc.nextInt();

int result[] = new int[10];

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

result[i] = (i + 1) \* n;

}

System.out.println("THE MULTIPLICATION TABLE OF " + n + " IS : ");

for (int j = 0; j < result.length; j++) {

System.out.println(result[j]);

}

} catch (InputMismatchException e) {

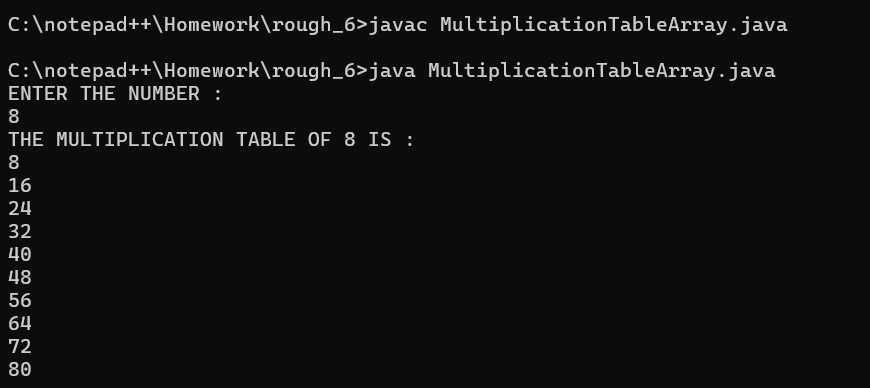
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – SUM OF ELEMENT**

**Q4:** Write a Program to check if the given number is a prime number or not

**Hint =>** a. A number that can be divided exactly only by itself and 1 are Prime Numbers, b. Prime Numbers checks are done for number greater than 1 c. Loop through all the numbers from 2 to the user input number and check if the reminder is zero. If the reminder is zero break out from the loop as the number is divisible by some other number and is not a prime number. d. Use isPrime boolean variable to store the result

**CODE:**

import java.util.\*;

public class SumOfElement {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

double num[] = new double[10];

double totalSum = 0;

int i = 0;

while (true) {

System.out.println("ENTER THE NUMBER (0 OR NEGATIVE NUMBER FOR EXIT) : ");

double n = sc.nextDouble();

if (n <= 0) {

break;

}

num[i] = n;

i++;

if (i == 10) {

break;

}

}

System.out.println("THE ELEMETS ARE : ");

for (int j = 0; j < num.length; j++) {

System.out.println(num[j] + " ");

totalSum += num[j];

}

System.out.println("THE TOTAL SUM OF ALL ELEMENT IN OUR ARRAY IS : " + totalSum);

} catch (InputMismatchException e) {

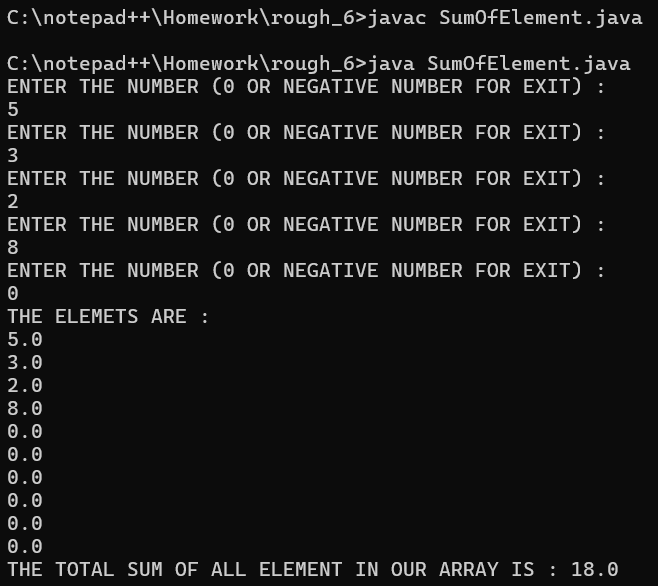
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – MULTIPLICATION TABLE FROM 6 TO 9**

**Q5:** Create a program to find the multiplication table of a number entered by the user from 6 to 9 and display the result

**Hint =>**

1. 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
2. Using a for loop, find the multiplication table of numbers from 6 to 9 and save the result in the array
3. Finally, display the result from the array in the format number \* i = \_\_\_

**CODE:**

import java.util.\*;

public class MultiplicationTabale {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER THE NUMBER : ");

int n = sc.nextInt();

int result[] = new int[4];

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

result[i] = (i + 6) \* n;

}

System.out.println("THE MULTIPLICATION TABLE (6-9) OF " + n + " IS : ");

for (int j = 0; j < result.length; j++) {

System.out.println(result[j]);

}

} catch (InputMismatchException e) {

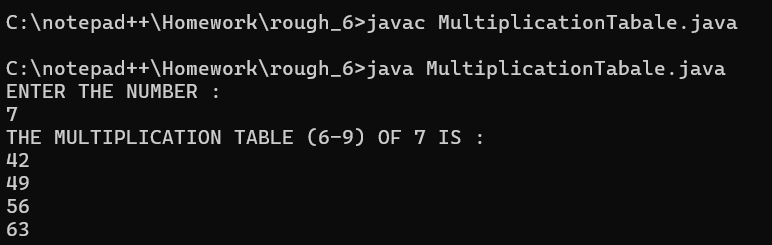
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – HEIGHT OF PLAYERS**

**Q6:**  Create a program to find the mean height of players present in a football team.

**Hint =>**

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

**CODE:**

import java.util.\*;

public class FootballerHeight {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

int heights[] = new int[11];

int totalHeight = 0;

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

System.out.println("ENTER THE HEIGHT OF FOOTBALLER - " + (i + 1) + " IS : ");

heights[i] = sc.nextInt();

}

for (int j = 0; j < heights.length; j++) {

totalHeight += heights[j];

}

System.out.println("THE SUM OF ALL FOOTBALLER'S HEIGHT IS : " + totalHeight);

double mean = totalHeight / 11.0;

System.out.println("THE MEAN HEIGHT OF PLAYERS PRESENT IN THAT FOOTBALL TEAM IS : " + mean);

} catch (InputMismatchException e) {

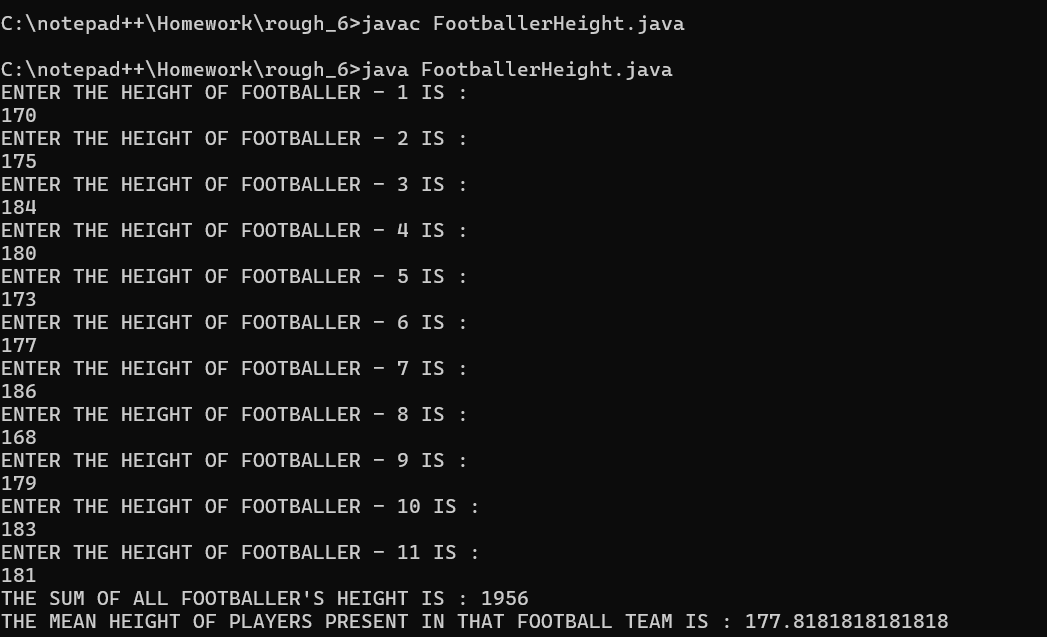
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – ODD & EVEN NUMBER**

**Q7:** 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 =>

1. 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
2. Create an integer array for even and odd numbers with size = number / 2 + 1
3. Create index variables for odd and even numbers and initialize them to zero
4. 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
5. Finally, print the odd and even numbers array using the odd and even index

**CODE:**

import java.util.\*;

public class OddEven {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER THE UPPER LIMIT OF THE EVEN-ODD SERIES : ");

int n = sc.nextInt();

if (n < 1) {

System.out.println("ERROR: ENTER A NATURAL NUMBER GREATER THAN 0.");

// sc.close();

// return;

System.exit(0);

}

int evenSize = n / 2 + 1;

int oddSize = n / 2 + 1;

int evenNumbers[] = new int[evenSize];

int oddNumbers[] = new int[oddSize];

int evenIndex = 0, oddIndex = 0;

for (int i = 1; i <= n; i++) {

if (i % 2 == 0) {

evenNumbers[evenIndex++] = i;

} else {

oddNumbers[oddIndex++] = i;

}

}

System.out.println("THE EVEN NUMBERS ARE : ");

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

System.out.print(evenNumbers[i] + " ");

}

System.out.println("\nTHE ODD NUMBERS ARE : ");

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

System.out.print(oddNumbers[i] + " ");

}

} catch (InputMismatchException e) {

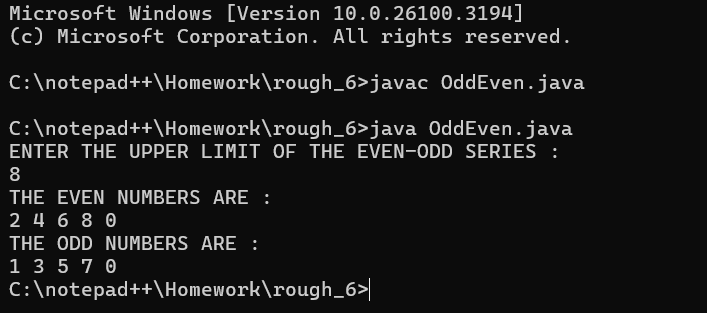
System.out.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – FACTORS IN AN ARRAY**

**Q8:** 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 =>**

1. Take the input for a number
2. 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.
3. 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
4. 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
5. Finally, Display the factors of the number

**CODE:**

import java.util.\*;

public class Factorial {

public static void main(String args[]) {

try (Scanner sc = new Scanner(System.in)) {

System.out.println("ENTER A NUMBER : ");

int n = sc.nextInt();

int maxFactor = 10;

int factors[] = new int[maxFactor];

int index = 0;

for (int i = 1; i <= n; i++) {

if (n % 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;

}

}

System.out.println("\nFACTORS OF " + n + " IS : ");

for (int i = 0; i < index; i++) {

System.out.print(factors[i] + " ");

}

} catch (InputMismatchException e) {

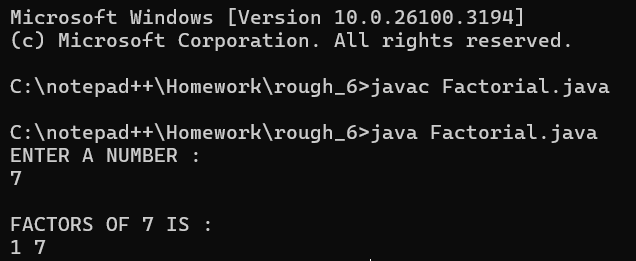
System.err.println("INVALID INPUT!");

}

}

}

**OUTPUT:**

****

**PROGRAM – COPYING AN ARRAY**

**Q9:** 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 =>**

1. Take user input for rows and columns, create a 2D array (Matrix), and take the user input
2. 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];
3. 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
4. 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

**CODE:**

import java.util.\*;

public class CopyArray {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("ENTER THE NUMBER OF ROWS : ");

int rows = sc.nextInt();

System.out.println("ENTER THE NUMBER OF COLUMNS : ");

int columns = sc.nextInt();

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

int array[] = new int[rows \* columns];

System.out.println("ENTER THE ELEMENTS OF THE MATRIX : ");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

matrix[i][j] = sc.nextInt();

}

}

int index = 0;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < columns; j++) {

array[index++] = matrix[i][j];

}

}

System.out.println("\n1D ARRAY : ");

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

System.out.println(array[i] + " ");

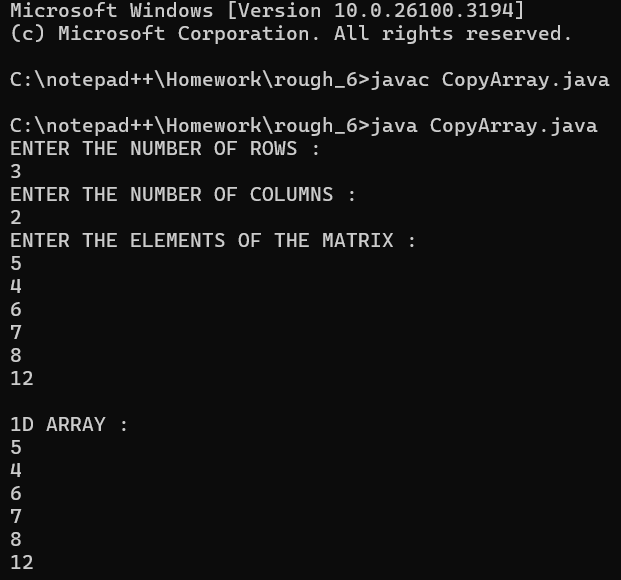
}

sc.close();

}

}

**OUTPUT:**

****

**PROGRAM – FIZZBUZZ**

**Q10:** 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 =>**

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

.

**CODE:**

import java.util.\*;

public class FizzBuzz {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("ENTER A POSITIVE INTEGER NUMBER : ");

int n = sc.nextInt();

if (n < 0) {

System.err.println("ERROR! : PLEASE ENTER A POSITIVE NUMBER.");

System.exit(0);

}

String results[] = new String[n + 1];

for (int i = 0; i <= n; i++) {

if (i % 3 == 0 && i % 5 == 0) {

results[i] = "FizzBuzz";

} else if (i % 3 == 0) {

results[i] = "Fizz";

} else if (i % 5 == 0) {

results[i] = "Buzz";

} else {

results[i] = String.valueOf(i);

}

}

System.out.println("\nFizzBuzz Results : ");

for (int i = 0; i <= n; i++) {

System.out.println("Position " + i + " = " + results[i]);

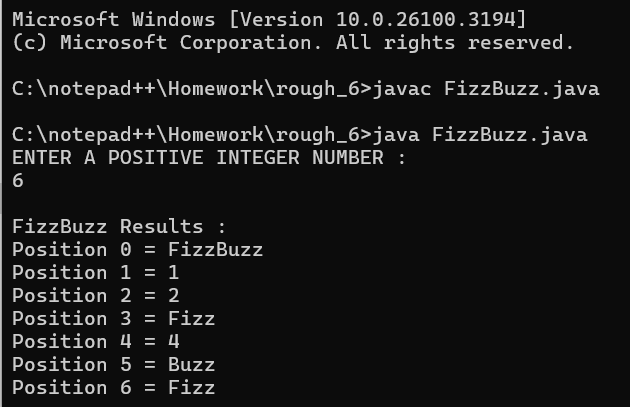
}

sc.close();

}

}

**OUTPUT:**

****