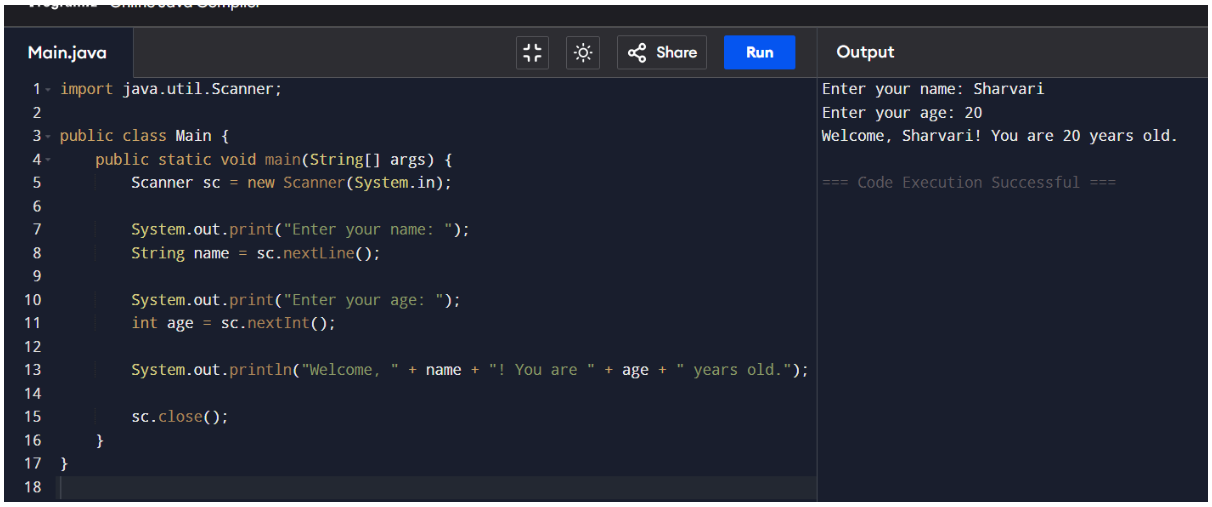
Name of the Student: Sharvari Muley

Date of Submission: 4-11-2025

Number of Programs Completed (out of 50): __34__

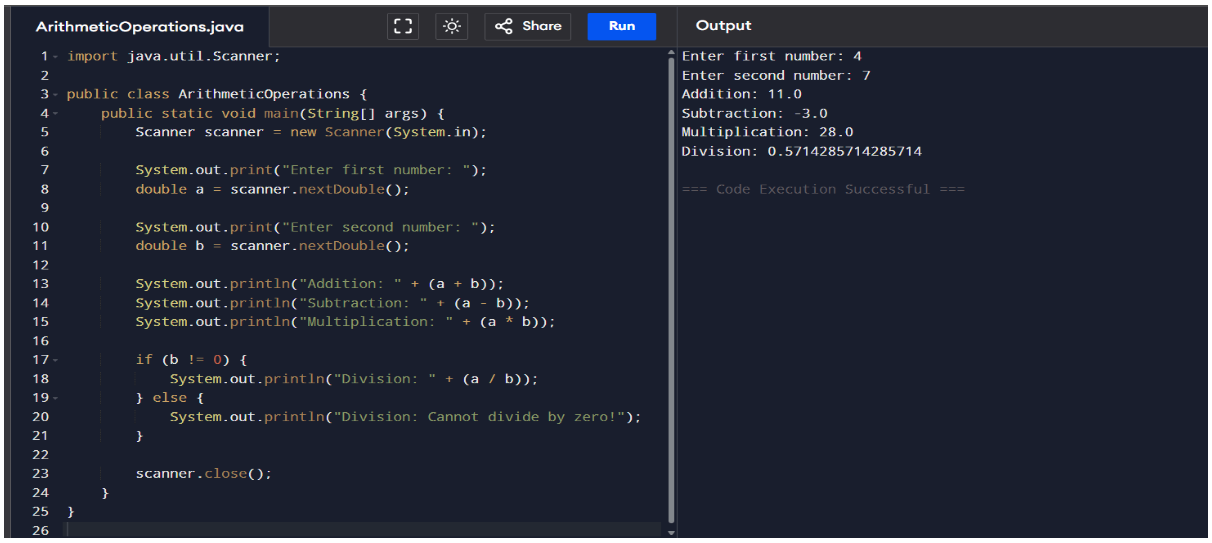
1. Develop a Java program to take user input for name and age and display a welcome message.



```java
import java.util.Scanner;

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

        System.out.print("Enter your name: ");
        String name = sc.nextLine();

        System.out.print("Enter your age: ");
        int age = sc.nextInt();

        System.out.println("Welcome, " + name + "! You are " + age + " years old.");

        sc.close();
    }
}
```

```
Enter your name: Sharvari
Enter your age: 20
Welcome, Sharvari! You are 20 years old.

=== Code Execution Successful ===
```

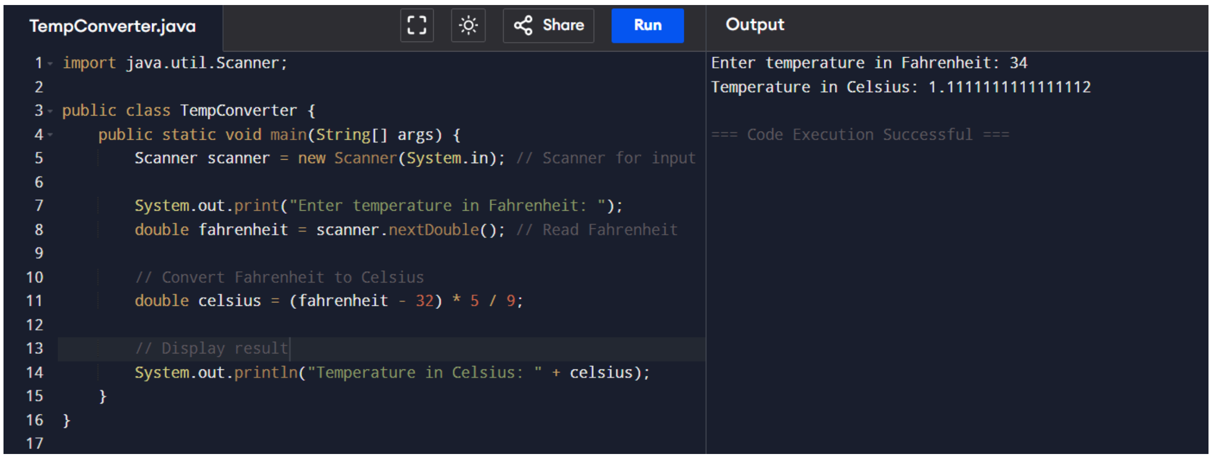
2.Write a Java program that takes two numbers and performs basic arithmetic operations (+, –, , /).



```java
import java.util.Scanner;

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

        System.out.print("Enter first number: ");
        double a = scanner.nextDouble();

        System.out.print("Enter second number: ");
        double b = scanner.nextDouble();

        System.out.println("Addition: " + (a + b));
        System.out.println("Subtraction: " + (a - b));
        System.out.println("Multiplication: " + (a * b));

        if (b != 0) {
            System.out.println("Division: " + (a / b));
        } else {
            System.out.println("Division: Cannot divide by zero!");
        }

        scanner.close();
    }
}
```

```
Enter first number: 4
Enter second number: 7
Addition: 11.0
Subtraction: -3.0
Multiplication: 28.0
Division: 0.5714285714285714

=== Code Execution Successful ===
```

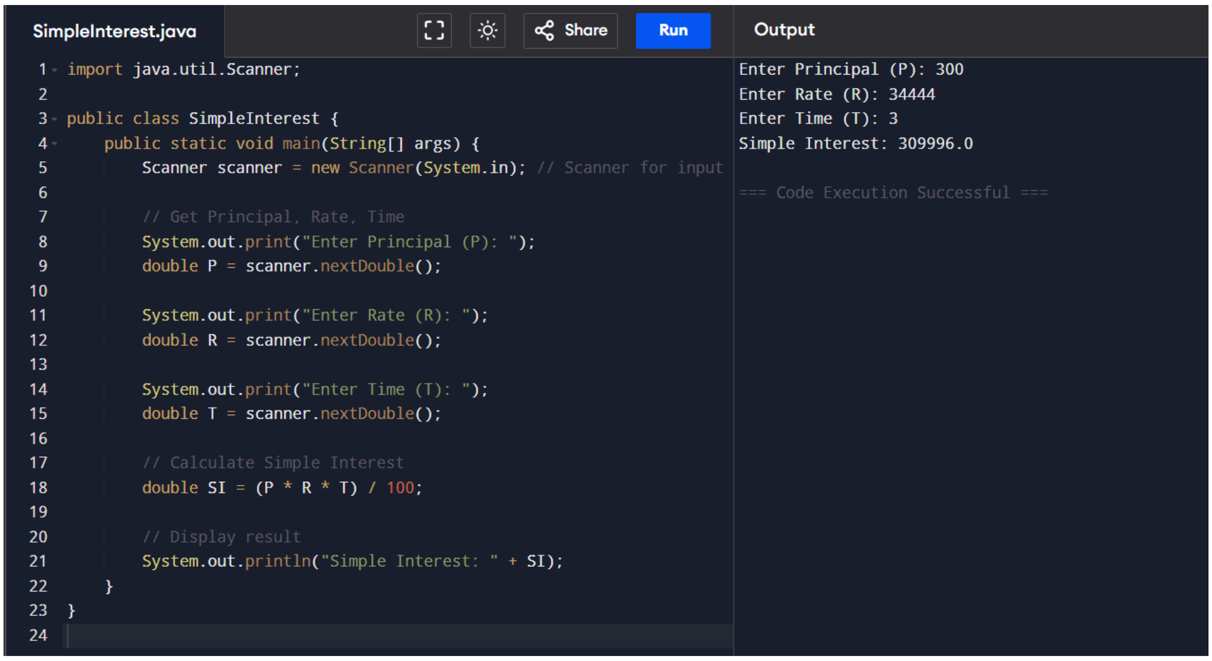
3.Create a program to convert temperature from Fahrenheit to Celsius.



```java
import java.util.Scanner;

public class TempConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        System.out.print("Enter temperature in Fahrenheit: ");
        double fahrenheit = scanner.nextDouble(); // Read Fahrenheit

        // Convert Fahrenheit to Celsius
        double celsius = (fahrenheit - 32) * 5 / 9;

        // Display result
        System.out.println("Temperature in Celsius: " + celsius);
    }
}
```

```
Enter temperature in Fahrenheit: 34
Temperature in Celsius: 1.1111111111111112

=== Code Execution Successful ===
```

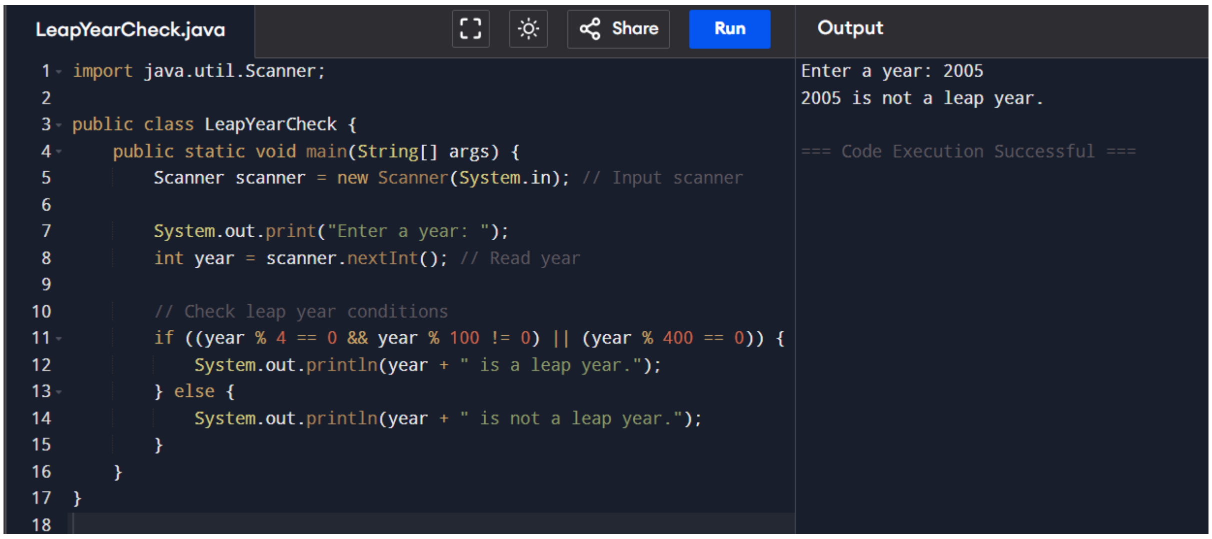
4.Design a Java application to calculate simple interest using the formula: SI = (P × R × T) / 100.



```java
import java.util.Scanner;

public class SimpleInterest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        // Get Principal, Rate, Time
        System.out.print("Enter Principal (P): ");
        double P = scanner.nextDouble();

        System.out.print("Enter Rate (R): ");
        double R = scanner.nextDouble();

        System.out.print("Enter Time (T): ");
        double T = scanner.nextDouble();

        // Calculate Simple Interest
        double SI = (P * R * T) / 100;

        // Display result
        System.out.println("Simple Interest: " + SI);
    }
}
```

```
Enter Principal (P): 300
Enter Rate (R): 34444
Enter Time (T): 3
Simple Interest: 309996.0

=== Code Execution Successful ===
```

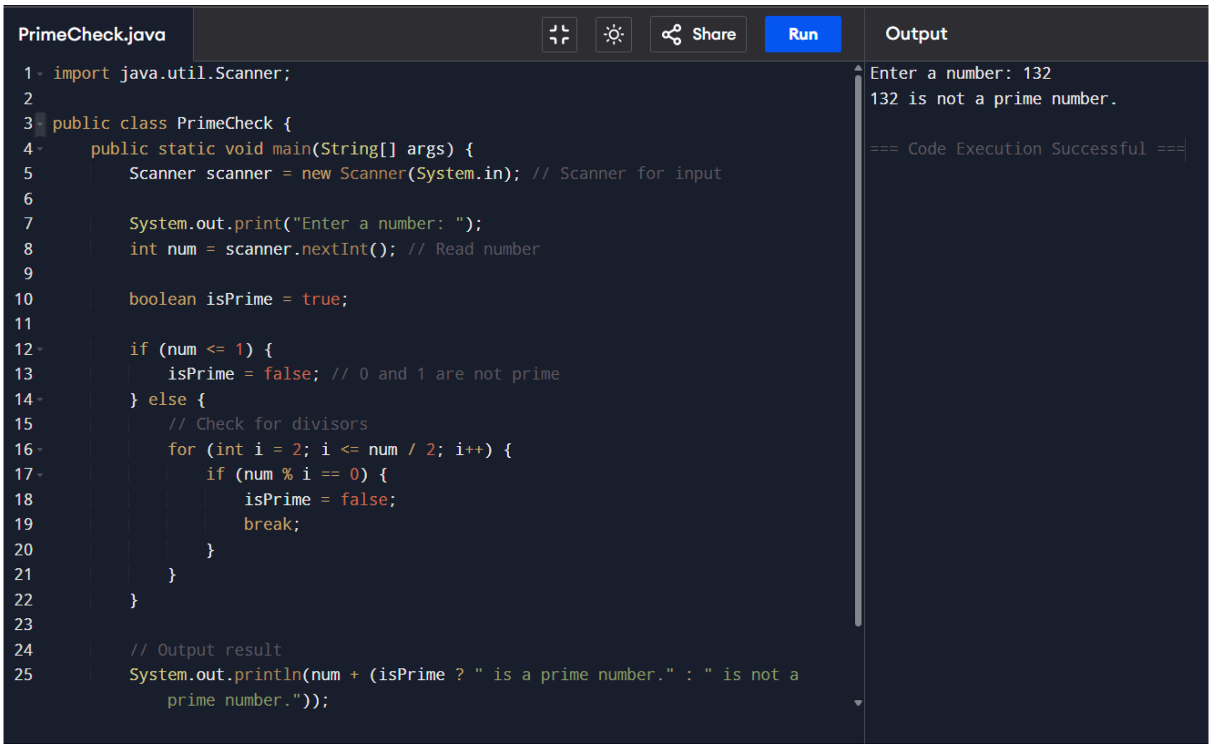
5.Write a Java program to determine whether a given year is a leap year.



```java
import java.util.Scanner;

public class LeapYearCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Input scanner

        System.out.print("Enter a year: ");
        int year = scanner.nextInt(); // Read year

        // Check leap year conditions
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
    }
}
```

```
Enter a year: 2005
2005 is not a leap year.

=== Code Execution Successful ===
```

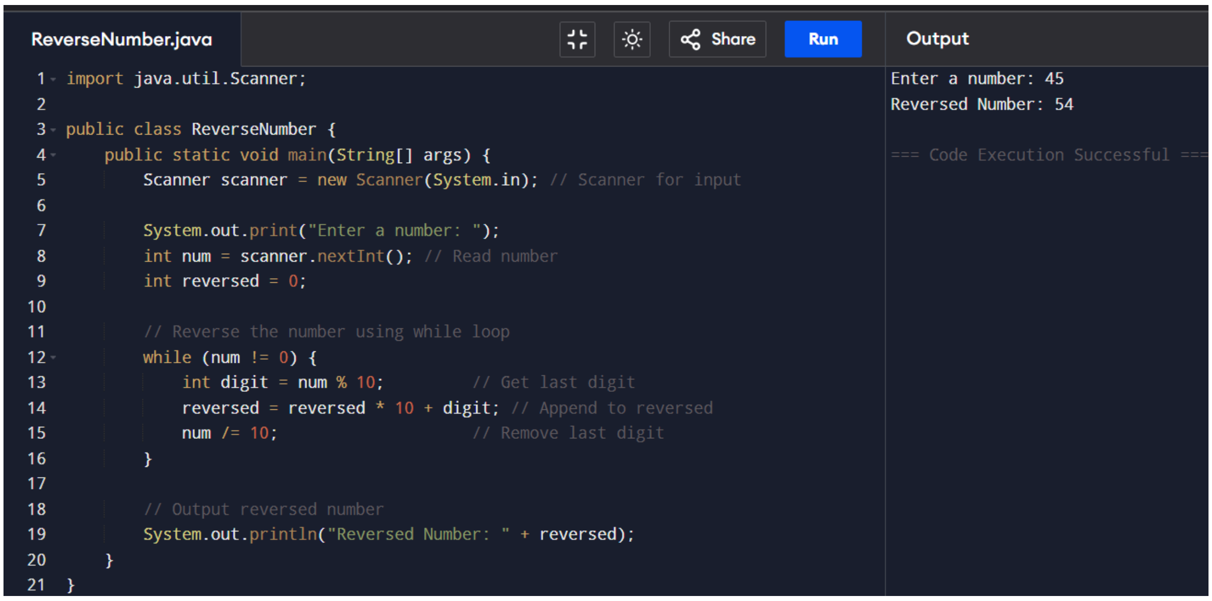
6.Develop a program to check whether an input number is prime or not using for loop..



```java
import java.util.Scanner;

public class PrimeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        System.out.print("Enter a number: ");
        int num = scanner.nextInt(); // Read number

        boolean isPrime = true;

        if (num <= 1) {
            isPrime = false; // 0 and 1 are not prime
        } else {
            // Check for divisors
            for (int i = 2; i <= num / 2; i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        // Output result
        System.out.println(num + (isPrime ? " is a prime number." : " is not a
            prime number."));
```

```
Enter a number: 132
132 is not a prime number.

=== Code Execution Successful ===
```

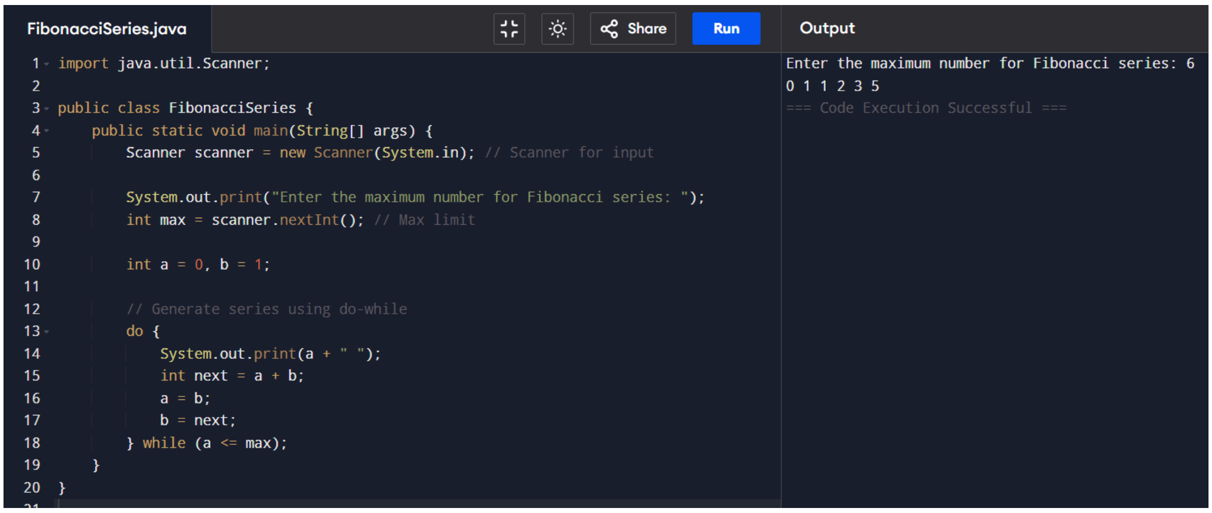
7.Write a program to reverse a number using a while loop.



```java
import java.util.Scanner;

public class ReverseNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        System.out.print("Enter a number: ");
        int num = scanner.nextInt(); // Read number
        int reversed = 0;

        // Reverse the number using while loop
        while (num != 0) {
            int digit = num % 10;          // Get last digit
            reversed = reversed * 10 + digit; // Append to reversed
            num /= 10;                     // Remove last digit
        }

        // Output reversed number
        System.out.println("Reversed Number: " + reversed);
    }
}
```

```
Enter a number: 45
Reversed Number: 54

=== Code Execution Successful ===
```

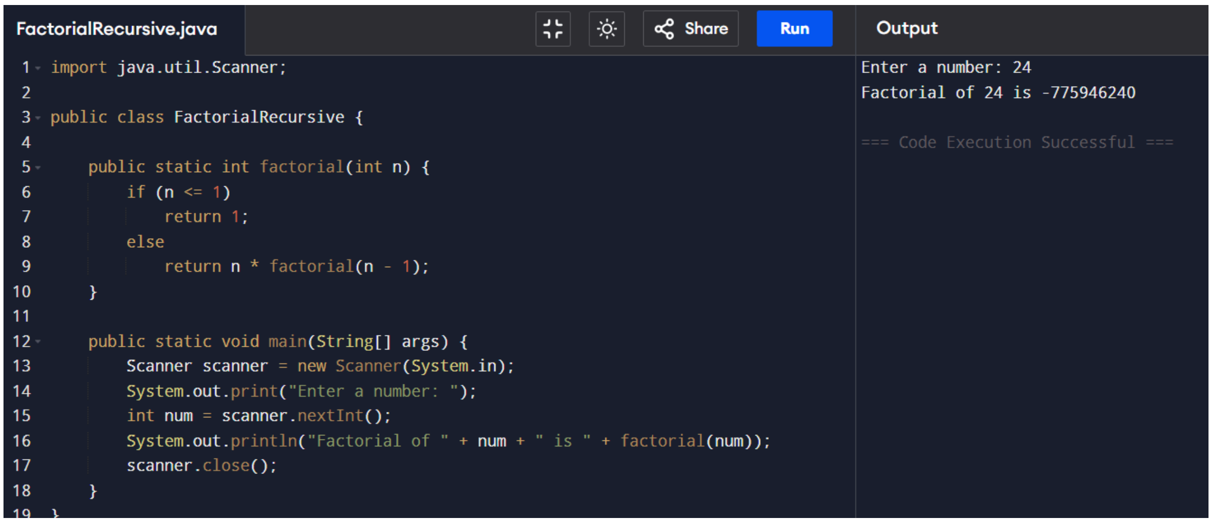
8.Create a Java application to generate Fibonacci series up to a given number using do-while loop..



```java
import java.util.Scanner;

public class FibonacciSeries {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        System.out.print("Enter the maximum number for Fibonacci series: ");
        int max = scanner.nextInt(); // Max limit

        int a = 0, b = 1;

        // Generate series using do-while
        do {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        } while (a <= max);
    }
}
```

```
Enter the maximum number for Fibonacci series: 6
0 1 1 2 3 5
=== Code Execution Successful ===
```

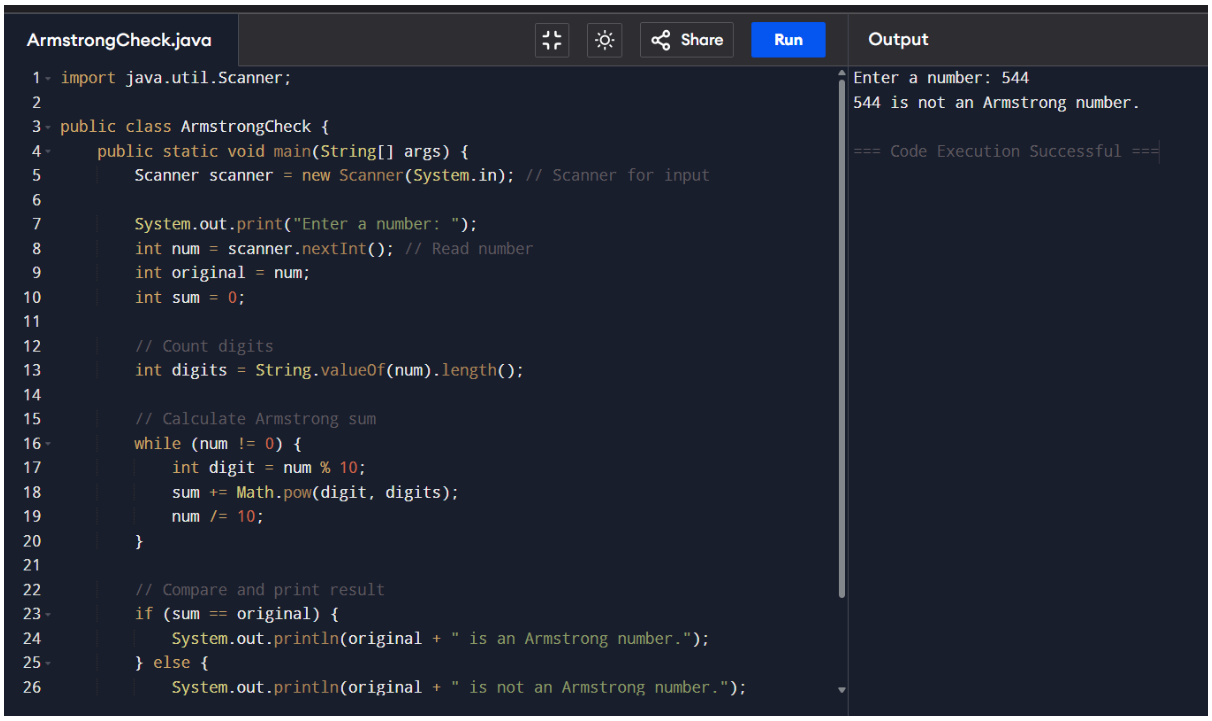
9.Design a recursive program to compute the factorial of a number using function.



```java
import java.util.Scanner;

public class FactorialRecursive {

    public static int factorial(int n) {
        if (n <= 1)
            return 1;
        else
            return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        System.out.println("Factorial of " + num + " is " + factorial(num));
        scanner.close();
    }
}
```

```
Enter a number: 24
Factorial of 24 is -775946240

=== Code Execution Successful ===
```

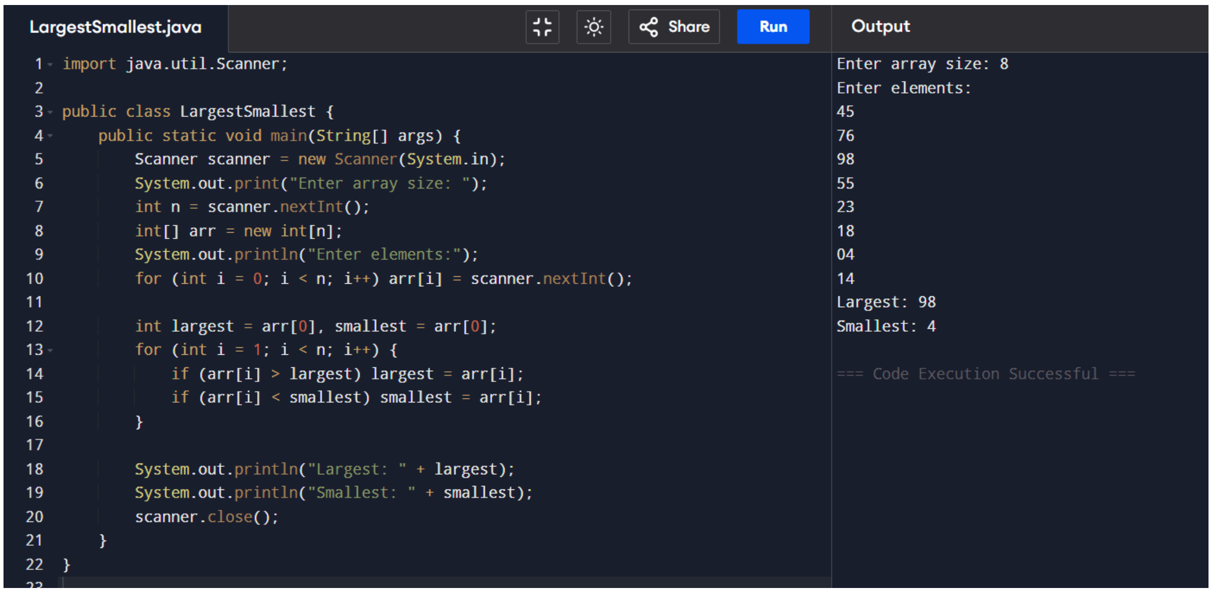
10.Implement a program to check whether a given number is an Armstrong number.



```java
import java.util.Scanner;

public class ArmstrongCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Scanner for input

        System.out.print("Enter a number: ");
        int num = scanner.nextInt(); // Read number
        int original = num;
        int sum = 0;

        // Count digits
        int digits = String.valueOf(num).length();

        // Calculate Armstrong sum
        while (num != 0) {
            int digit = num % 10;
            sum += Math.pow(digit, digits);
            num /= 10;
        }

        // Compare and print result
        if (sum == original) {
            System.out.println(original + " is an Armstrong number.");
        } else {
            System.out.println(original + " is not an Armstrong number.");
```

```
Enter a number: 544
544 is not an Armstrong number.

=== Code Execution Successful ===
```

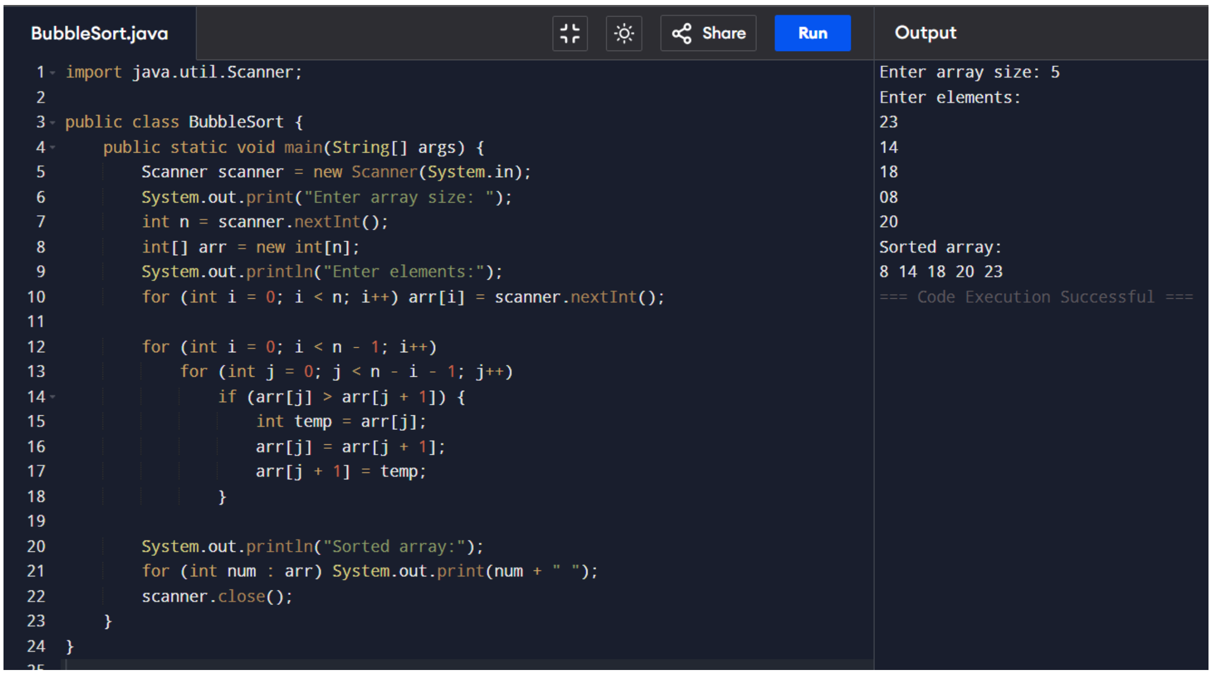
11.Write a Java program to find the largest and smallest number in an array.



```java
import java.util.Scanner;

public class LargestSmallest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) arr[i] = scanner.nextInt();

        int largest = arr[0], smallest = arr[0];
        for (int i = 1; i < n; i++) {
            if (arr[i] > largest) largest = arr[i];
            if (arr[i] < smallest) smallest = arr[i];
        }

        System.out.println("Largest: " + largest);
        System.out.println("Smallest: " + smallest);
        scanner.close();
    }
}
```

```
Enter array size: 8
Enter elements:
45
76
98
55
23
18
04
14
Largest: 98
Smallest: 4

=== Code Execution Successful ===
```

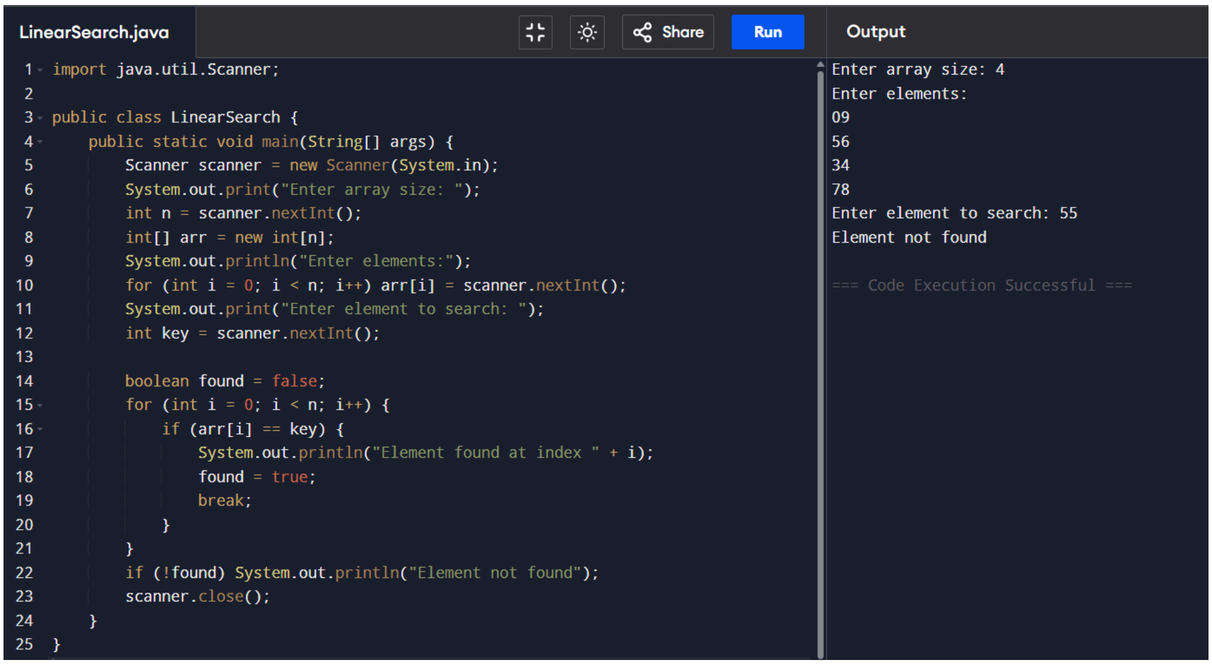
12.Develop a program to sort an array using bubble sort algorithm.



```java
import java.util.Scanner;

public class BubbleSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) arr[i] = scanner.nextInt();

        for (int i = 0; i < n - 1; i++)
            for (int j = 0; j < n - i - 1; j++)
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }

        System.out.println("Sorted array:");
        for (int num : arr) System.out.print(num + " ");
        scanner.close();
    }
}
```

```
Enter array size: 5
Enter elements:
23
14
18
08
20
Sorted array:
8 14 18 20 23
=== Code Execution Successful ===
```

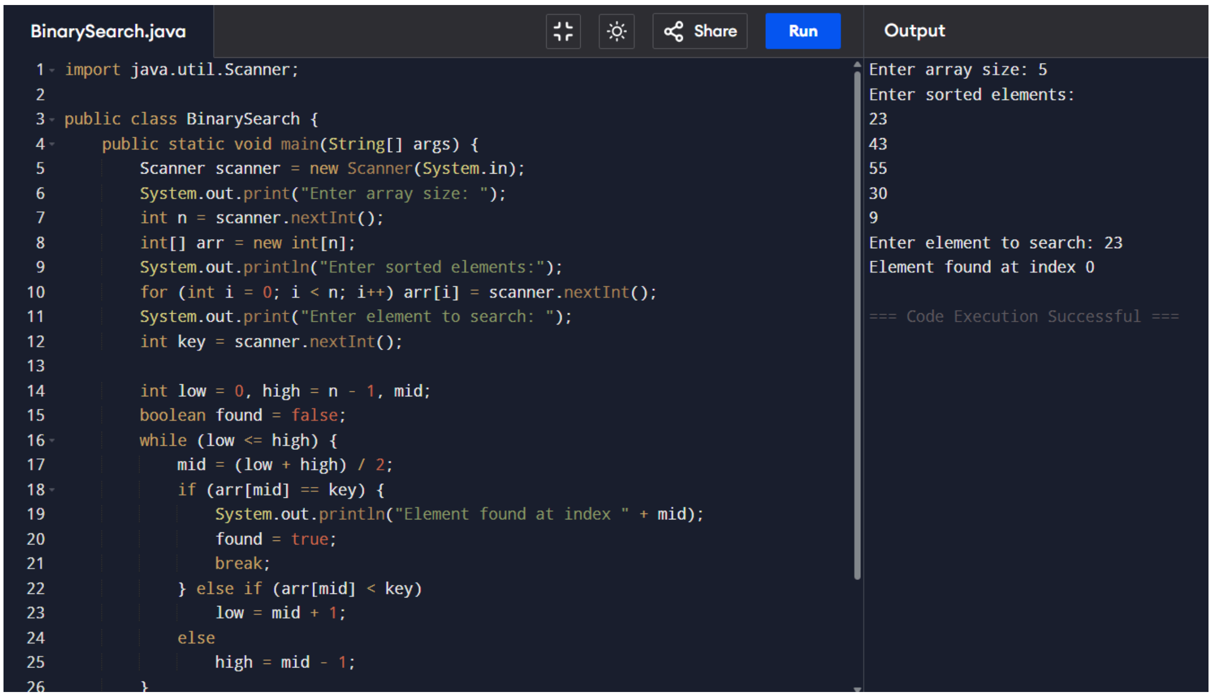
13.Implement linear search to find an element in an array.



```java
import java.util.Scanner;

public class LinearSearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) arr[i] = scanner.nextInt();
        System.out.print("Enter element to search: ");
        int key = scanner.nextInt();

        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (arr[i] == key) {
                System.out.println("Element found at index " + i);
                found = true;
                break;
            }
        }
        if (!found) System.out.println("Element not found");
        scanner.close();
    }
}
```

```
Enter array size: 4
Enter elements:
09
56
34
78
Enter element to search: 55
Element not found

=== Code Execution Successful ===
```

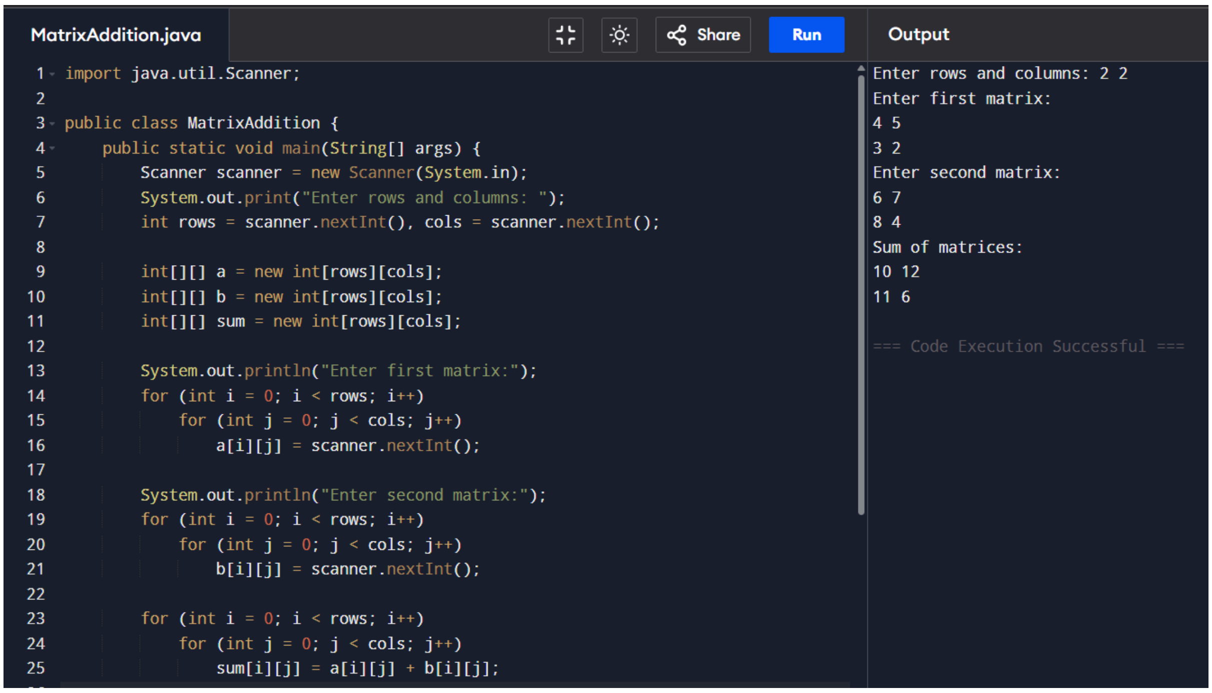
14.Implement binary search to find an element in an array.



```java
import java.util.Scanner;

public class BinarySearch {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int n = scanner.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter sorted elements:");
        for (int i = 0; i < n; i++) arr[i] = scanner.nextInt();
        System.out.print("Enter element to search: ");
        int key = scanner.nextInt();

        int low = 0, high = n - 1, mid;
        boolean found = false;
        while (low <= high) {
            mid = (low + high) / 2;
            if (arr[mid] == key) {
                System.out.println("Element found at index " + mid);
                found = true;
                break;
            } else if (arr[mid] < key)
                low = mid + 1;
            else
                high = mid - 1;
        }
```

```
Enter array size: 5
Enter sorted elements:
23
43
55
30
9
Enter element to search: 23
Element found at index 0

=== Code Execution Successful ===
```

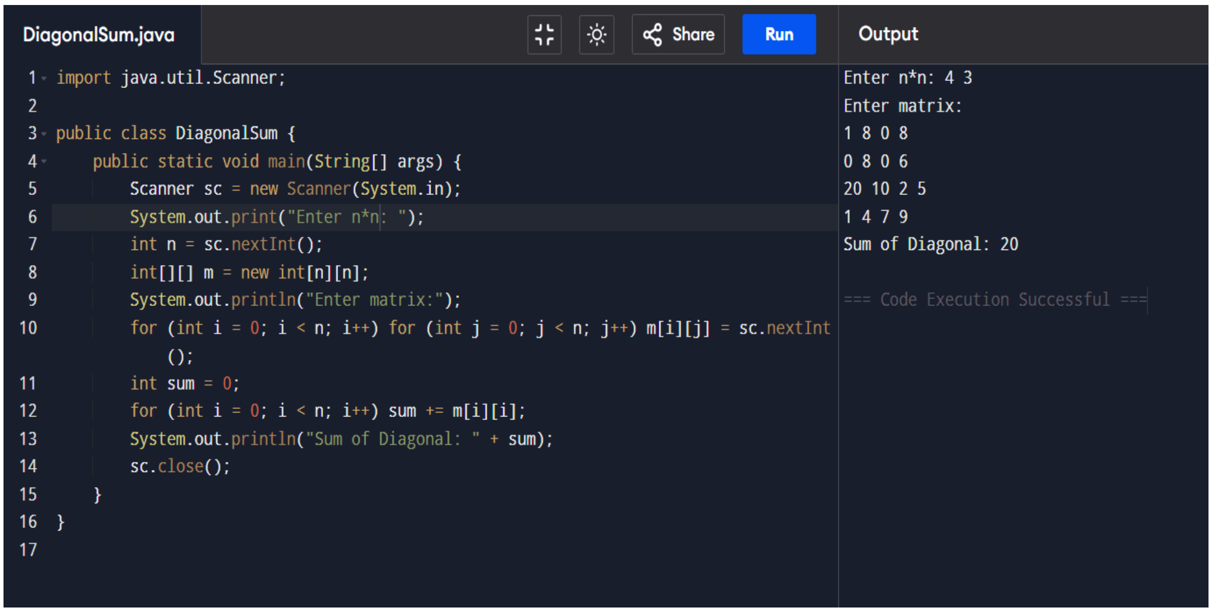
15.Write a Java program to perform matrix addition using for loop.



```java
import java.util.Scanner;

public class MatrixAddition {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter rows and columns: ");
        int rows = scanner.nextInt(), cols = scanner.nextInt();

        int[][] a = new int[rows][cols];
        int[][] b = new int[rows][cols];
        int[][] sum = new int[rows][cols];

        System.out.println("Enter first matrix:");
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                a[i][j] = scanner.nextInt();

        System.out.println("Enter second matrix:");
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                b[i][j] = scanner.nextInt();

        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                sum[i][j] = a[i][j] + b[i][j];
```

```
Enter rows and columns: 2 2
Enter first matrix:
4 5
3 2
Enter second matrix:
6 7
8 4
Sum of matrices:
10 12
11 6

=== Code Execution Successful ===
```

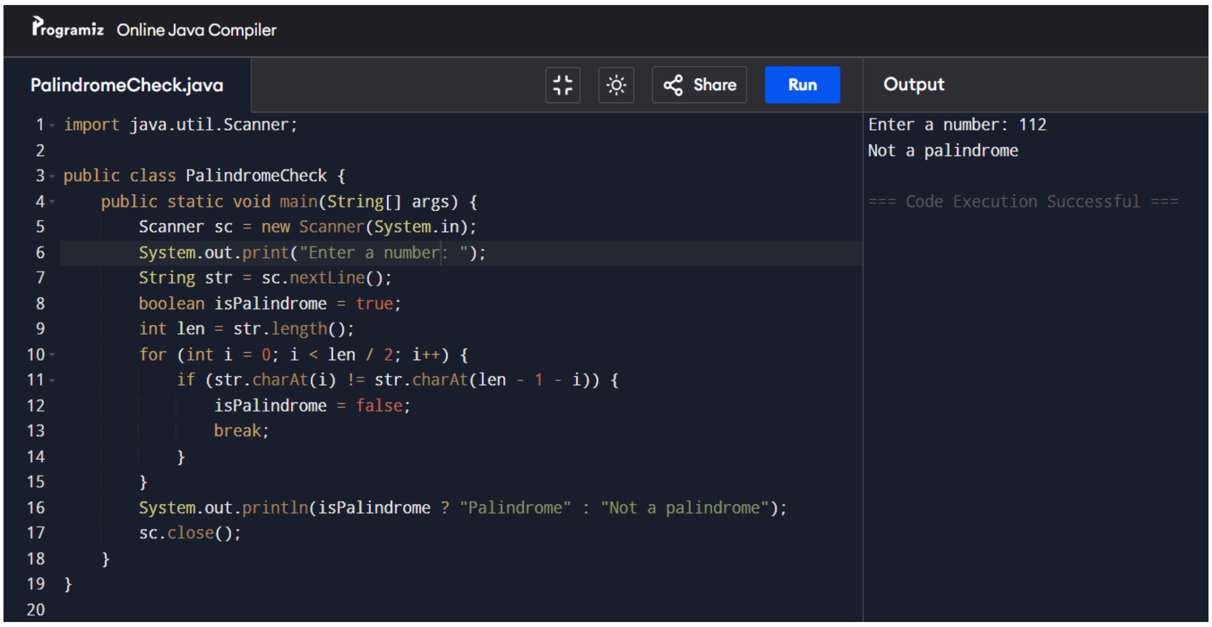
16.Write a java program to find the sum of diagonal elements in an array.



```java
import java.util.Scanner;

public class DiagonalSum {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter n*n: ");
        int n = sc.nextInt();
        int[][] m = new int[n][n];
        System.out.println("Enter matrix:");
        for (int i = 0; i < n; i++) for (int j = 0; j < n; j++) m[i][j] = sc.nextInt
            ();
        int sum = 0;
        for (int i = 0; i < n; i++) sum += m[i][i];
        System.out.println("Sum of Diagonal: " + sum);
        sc.close();
    }
}
```

```
Enter n*n: 4 3
Enter matrix:
1 8 0 8
0 8 0 6
20 10 2 5
1 4 7 9
Sum of Diagonal: 20

=== Code Execution Successful ===
```

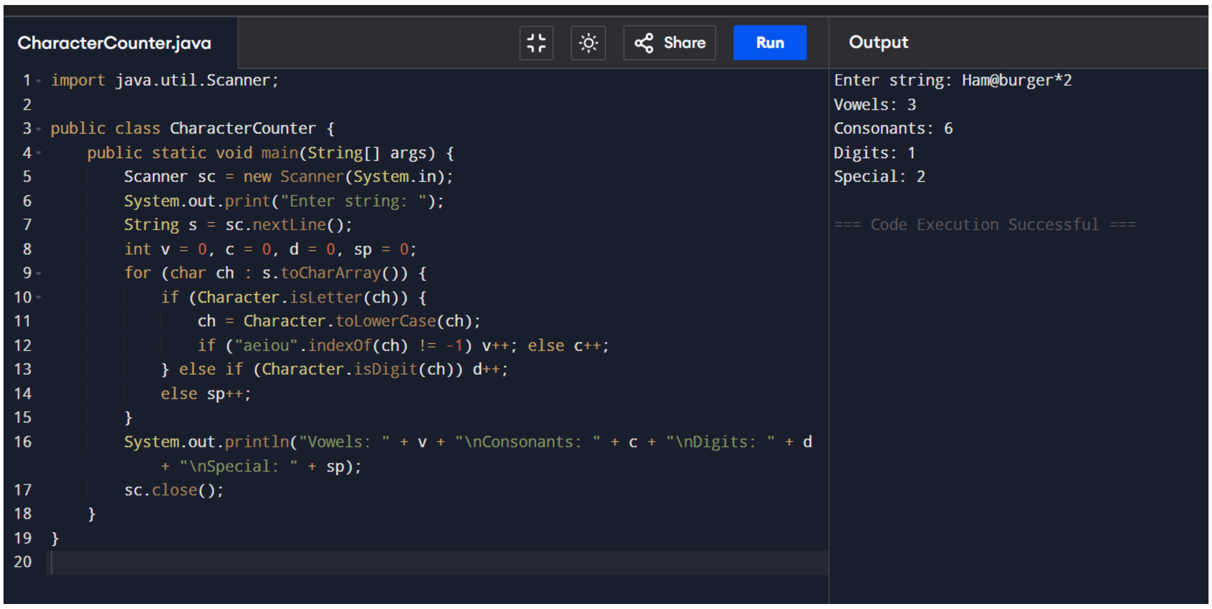
17.Check whether a given string is a palindrome.



```java
import java.util.Scanner;

public class PalindromeCheck {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        String str = sc.nextLine();
        boolean isPalindrome = true;
        int len = str.length();
        for (int i = 0; i < len / 2; i++) {
            if (str.charAt(i) != str.charAt(len - 1 - i)) {
                isPalindrome = false;
                break;
            }
        }
        System.out.println(isPalindrome ? "Palindrome" : "Not a palindrome");
        sc.close();
    }
}

```

```
Enter a number: 112
Not a palindrome

=== Code Execution Successful ===
```

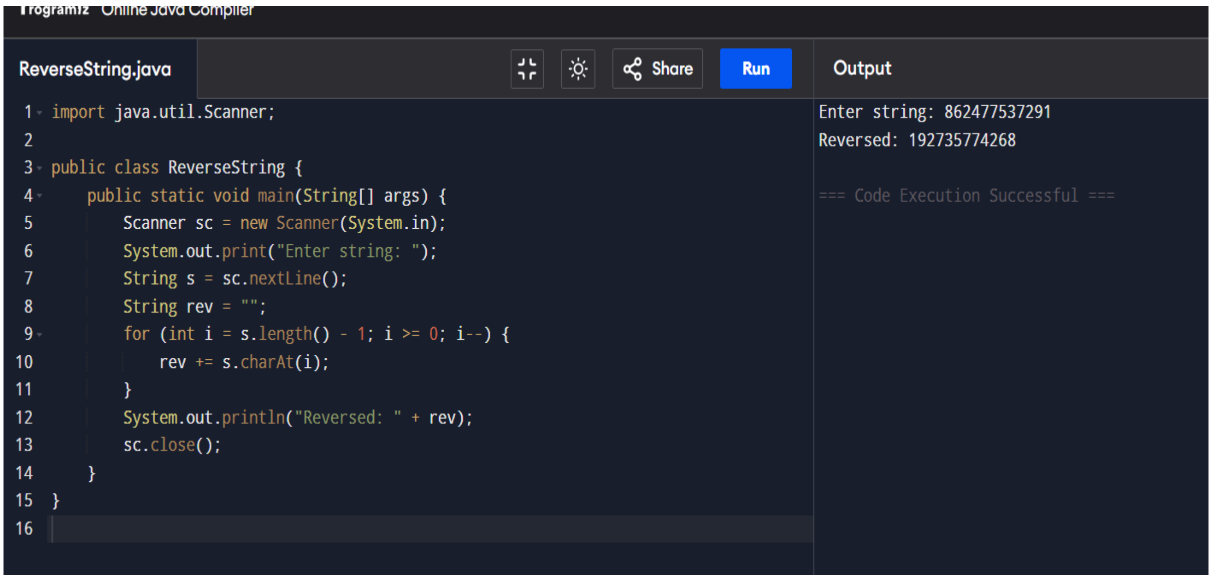
18.Count the number of vowels, consonants, digits, and special characters in a string.



```java
import java.util.Scanner;

public class CharacterCounter {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter string: ");
        String s = sc.nextLine();
        int v = 0, c = 0, d = 0, sp = 0;
        for (char ch : s.toCharArray()) {
            if (Character.isLetter(ch)) {
                ch = Character.toLowerCase(ch);
                if ("aeiou".indexOf(ch) != -1) v++; else c++;
            } else if (Character.isDigit(ch)) d++;
            else sp++;
        }
        System.out.println("Vowels: " + v + "\nConsonants: " + c + "\nDigits: " + d
            + "\nSpecial: " + sp);
        sc.close();
    }
}

```

```
Enter string: Ham@burger*2
Vowels: 3
Consonants: 6
Digits: 1
Special: 2

=== Code Execution Successful ===
```

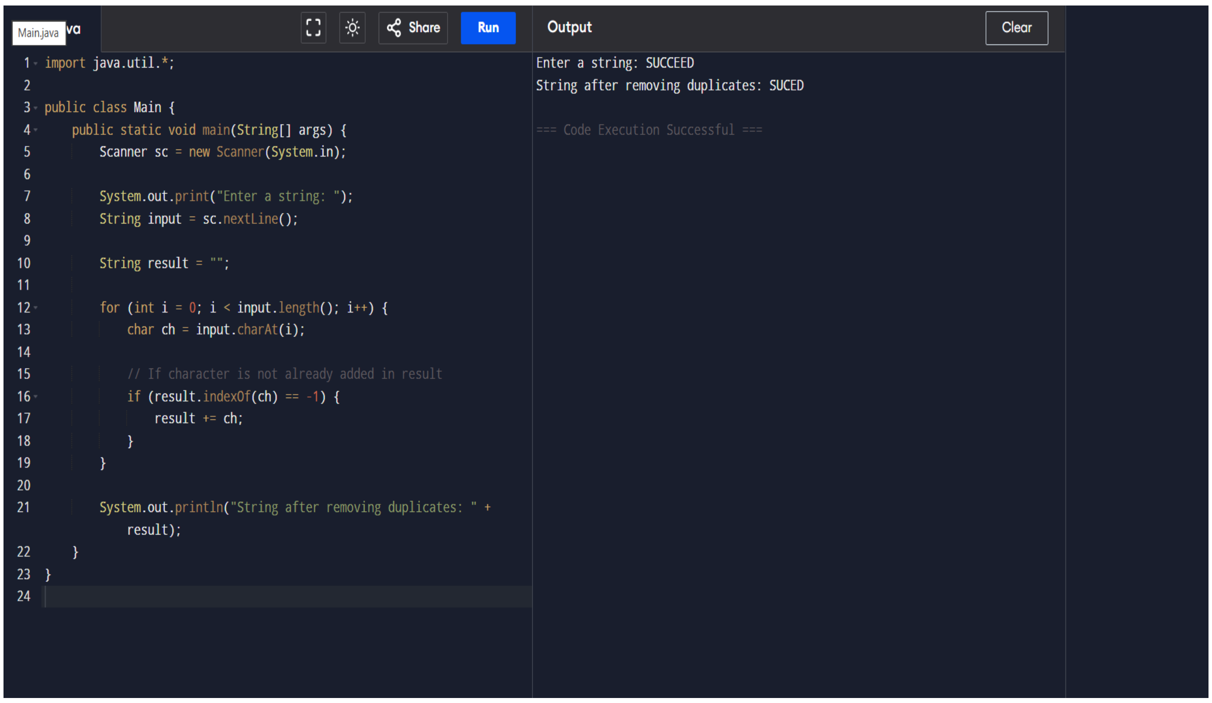
19.Program to reverse the string using predefined methods in String class.



```java
import java.util.Scanner;

public class ReverseString {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter string: ");
        String s = sc.nextLine();
        String rev = "";
        for (int i = s.length() - 1; i >= 0; i--) {
            rev += s.charAt(i);
        }
        System.out.println("Reversed: " + rev);
        sc.close();
    }
}
```

```
Enter string: 862477537291
Reversed: 192735774268

=== Code Execution Successful ===
```

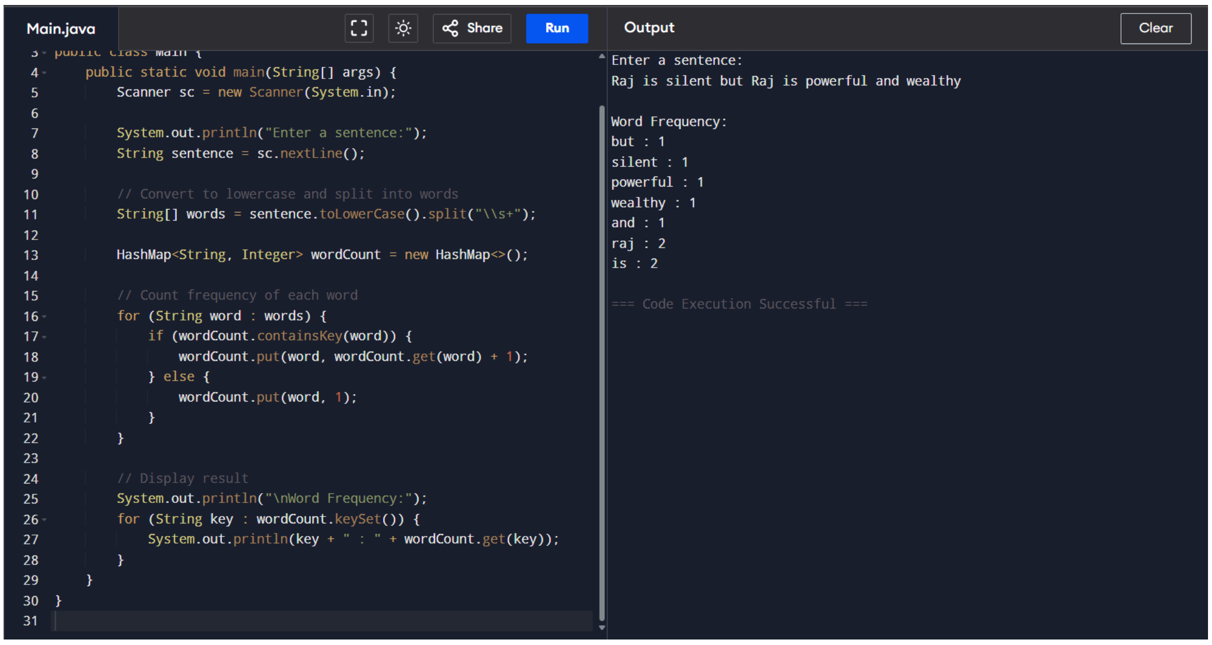
20.Write a program to remove duplicate characters from a string.



```java
import java.util.*;

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

        System.out.print("Enter a string: ");
        String input = sc.nextLine();

        String result = "";

        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);

            // If character is not already added in result
            if (result.indexOf(ch) == -1) {
                result += ch;
            }
        }

        System.out.println("String after removing duplicates: " +
            result);
    }
}
```

```
Enter a string: SUCCEED
String after removing duplicates: SUCED

=== Code Execution Successful ===
```

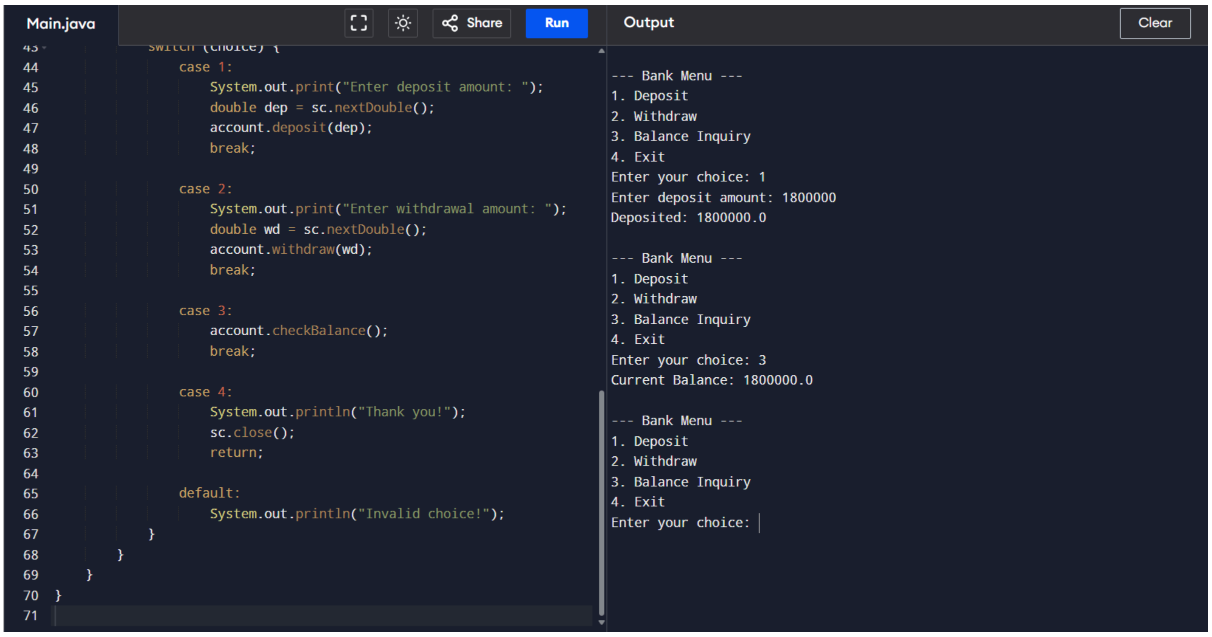
21.Develop a Java program to count the frequency of each word in a sentence.



```java
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a sentence:");
        String sentence = sc.nextLine();

        // Convert to lowercase and split into words
        String[] words = sentence.toLowerCase().split("\\s+");

        HashMap<String, Integer> wordCount = new HashMap<>();

        // Count frequency of each word
        for (String word : words) {
            if (wordCount.containsKey(word)) {
                wordCount.put(word, wordCount.get(word) + 1);
            } else {
                wordCount.put(word, 1);
            }
        }

        // Display result
        System.out.println("\nWord Frequency:");
        for (String key : wordCount.keySet()) {
            System.out.println(key + " : " + wordCount.get(key));
        }
    }
}
```

```
Enter a sentence:
Raj is silent but Raj is powerful and wealthy

Word Frequency:
but : 1
silent : 1
powerful : 1
wealthy : 1
and : 1
raj : 2
is : 2

=== Code Execution Successful ===
```

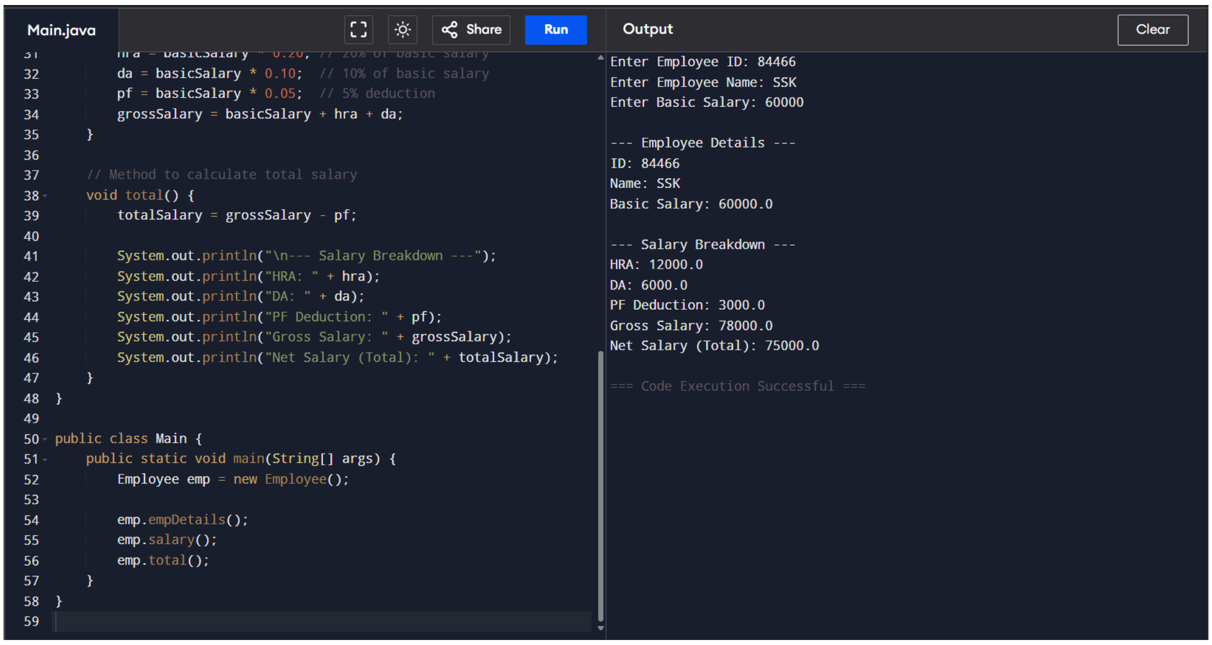
22.Design a class BankAccount with methods for deposit, withdraw, and balance inquiry.



```java
            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double dep = sc.nextDouble();
                    account.deposit(dep);
                    break;

                case 2:
                    System.out.print("Enter withdrawal amount: ");
                    double wd = sc.nextDouble();
                    account.withdraw(wd);
                    break;

                case 3:
                    account.checkBalance();
                    break;

                case 4:
                    System.out.println("Thank you!");
                    sc.close();
                    return;

                default:
                    System.out.println("Invalid choice!");
            }
        }
    }
}
```

```
--- Bank Menu ---
1. Deposit
2. Withdraw
3. Balance Inquiry
4. Exit
Enter your choice: 1
Enter deposit amount: 1800000
Deposited: 1800000.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Balance Inquiry
4. Exit
Enter your choice: 3
Current Balance: 1800000.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Balance Inquiry
4. Exit
Enter your choice:
```

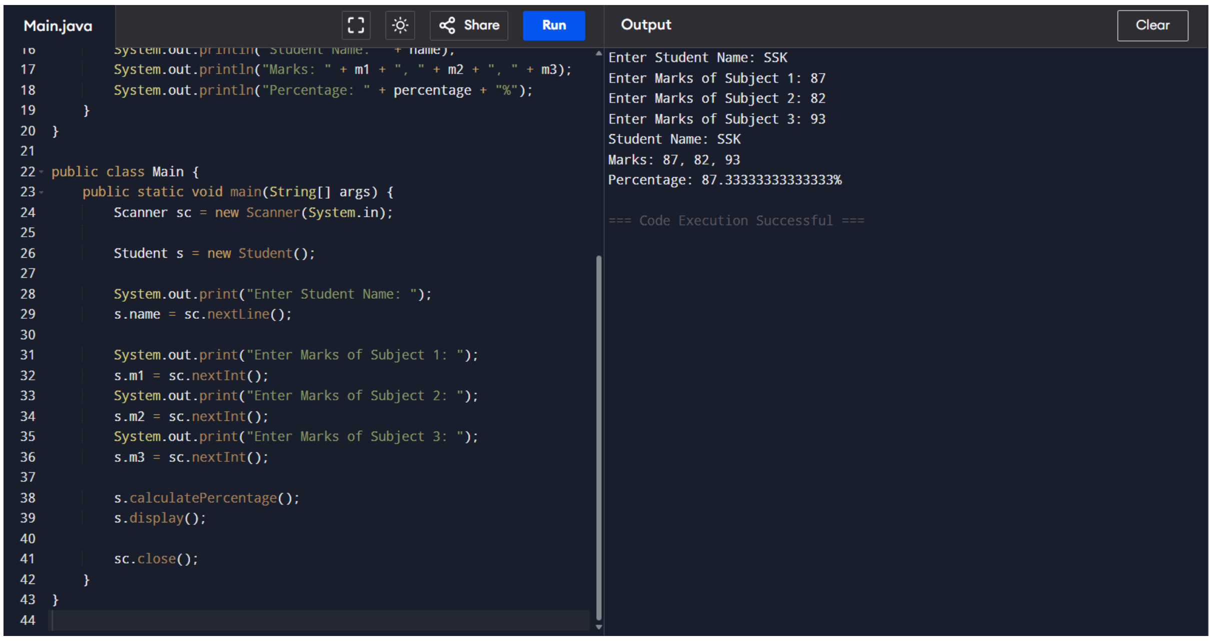
23.Design a Java class Employee with the following:A method `empDetails()` to accept and display employee details.A method `salary()` to compute basic salary components.A method `total()` to calculate the total salary (including allowances/deductions).


```java
        hra = basicSalary * 0.20; // 20% of basic salary
        da = basicSalary * 0.10;  // 10% of basic salary
        pf = basicSalary * 0.05;  // 5% deduction
        grossSalary = basicSalary + hra + da;
    }

    // Method to calculate total salary
    void total() {
        totalSalary = grossSalary - pf;

        System.out.println("\n--- Salary Breakdown ---");
        System.out.println("HRA: " + hra);
        System.out.println("DA: " + da);
        System.out.println("PF Deduction: " + pf);
        System.out.println("Gross Salary: " + grossSalary);
        System.out.println("Net Salary (Total): " + totalSalary);
    }
}

public class Main {
    public static void main(String[] args) {
        Employee emp = new Employee();

        emp.empDetails();
        emp.salary();
        emp.total();
    }
}
```

Output

```
Enter Employee ID: 84466
Enter Employee Name: SSK
Enter Basic Salary: 60000

--- Employee Details ---
ID: 84466
Name: SSK
Basic Salary: 60000.0

--- Salary Breakdown ---
HRA: 12000.0
DA: 6000.0
PF Deduction: 3000.0
Gross Salary: 78000.0
Net Salary (Total): 75000.0

=== Code Execution Successful ===
```

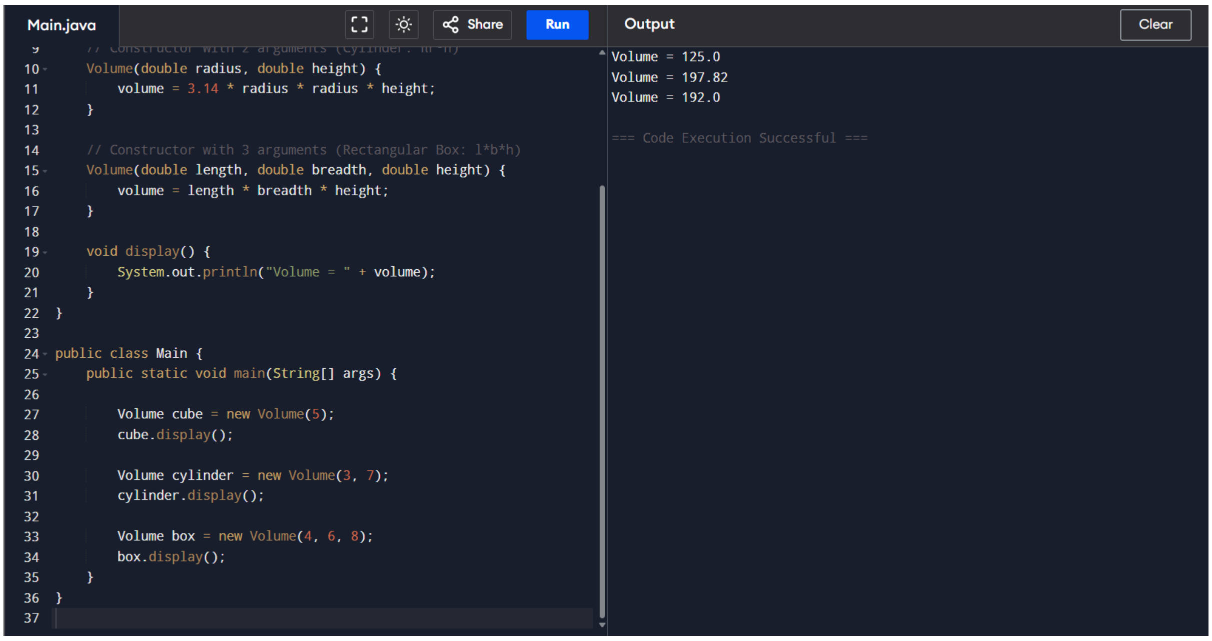
24.Create a Student class with marks in 3 subjects and compute the result with percentage.


```java
        System.out.println("Student Name: " + name);
        System.out.println("Marks: " + m1 + ", " + m2 + ", " + m3);
        System.out.println("Percentage: " + percentage + "%");
    }
}

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

        Student s = new Student();

        System.out.print("Enter Student Name: ");
        s.name = sc.nextLine();

        System.out.print("Enter Marks of Subject 1: ");
        s.m1 = sc.nextInt();
        System.out.print("Enter Marks of Subject 2: ");
        s.m2 = sc.nextInt();
        System.out.print("Enter Marks of Subject 3: ");
        s.m3 = sc.nextInt();

        s.calculatePercentage();
        s.display();

        sc.close();
    }
}
```

Output

```
Enter Student Name: SSK
Enter Marks of Subject 1: 87
Enter Marks of Subject 2: 82
Enter Marks of Subject 3: 93
Student Name: SSK
Marks: 87, 82, 93
Percentage: 87.33333333333333%

=== Code Execution Successful ===
```

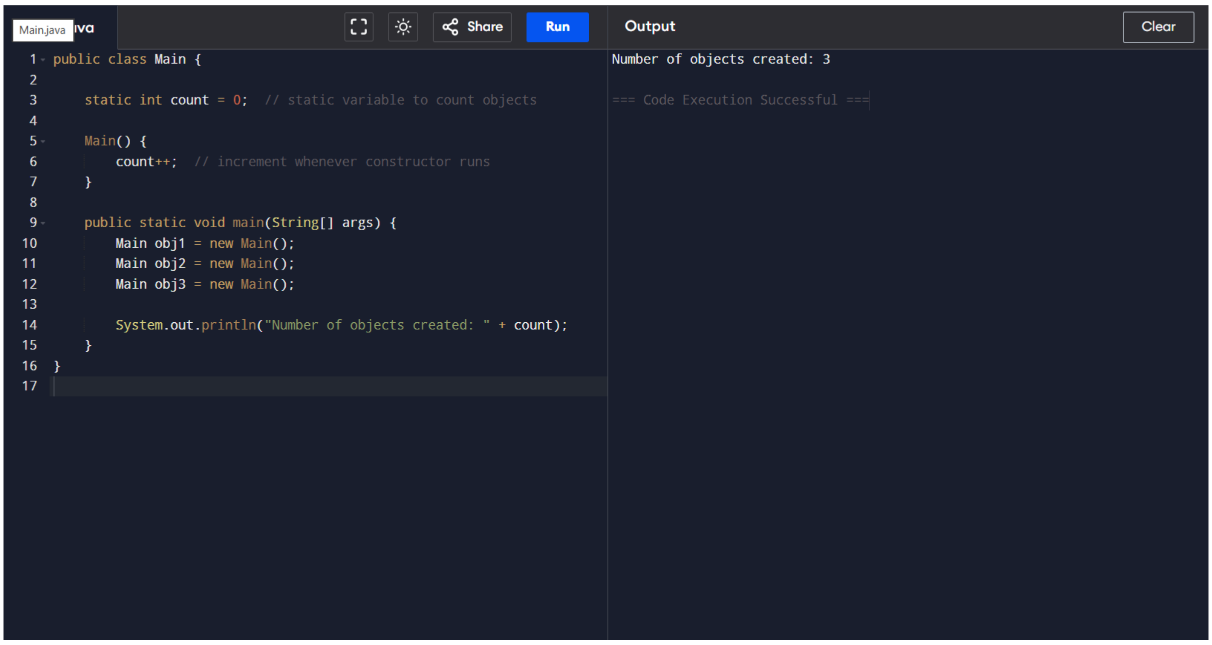
25.Create a class Volume and create three constructor with one arg,two arg and three arg with the help of constructor overloading concept.



```java
    // Constructor with 2 arguments (Cylinder: πr^2h)
    Volume(double radius, double height) {
        volume = 3.14 * radius * radius * height;
    }

    // Constructor with 3 arguments (Rectangular Box: l*b*h)
    Volume(double length, double breadth, double height) {
        volume = length * breadth * height;
    }

    void display() {
        System.out.println("Volume = " + volume);
    }
}

public class Main {
    public static void main(String[] args) {

        Volume cube = new Volume(5);
        cube.display();

        Volume cylinder = new Volume(3, 7);
        cylinder.display();

        Volume box = new Volume(4, 6, 8);
        box.display();
    }
}
```

```
Volume = 125.0
Volume = 197.82
Volume = 192.0

=== Code Execution Successful ===
```

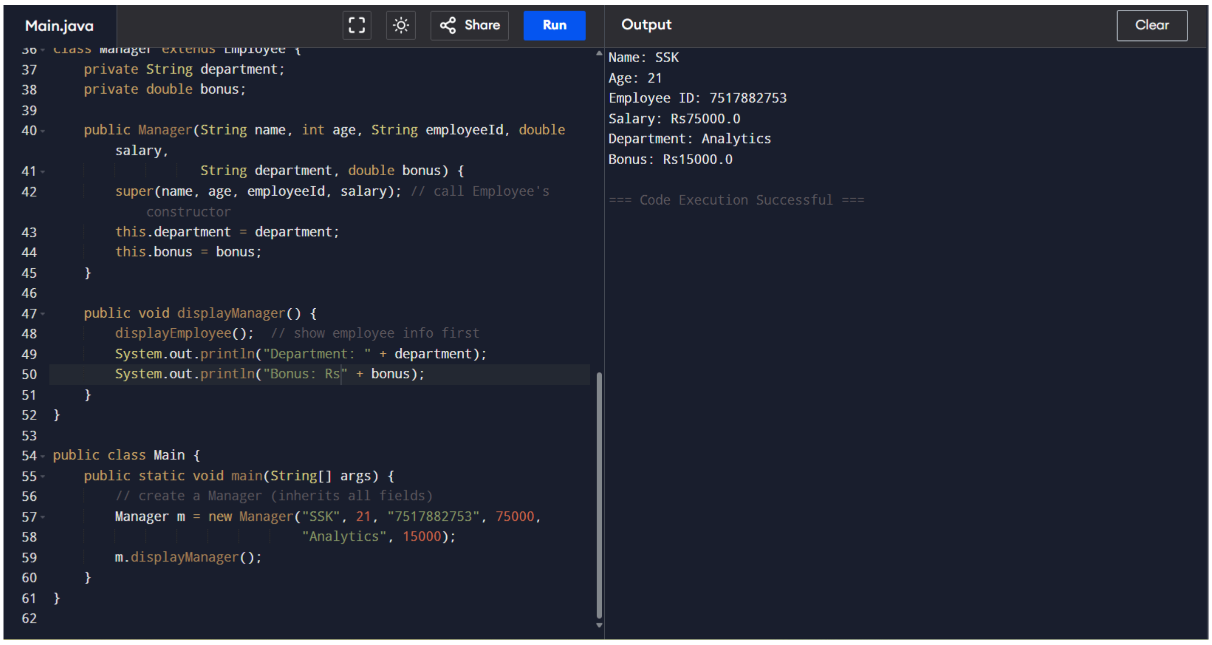
26.Write a program to count number of object/instances created in a class.



```java
public class Main {

    static int count = 0;  // static variable to count objects

    Main() {
        count++;  // increment whenever constructor runs
    }

    public static void main(String[] args) {
        Main obj1 = new Main();
        Main obj2 = new Main();
        Main obj3 = new Main();

        System.out.println("Number of objects created: " + count);
    }
}
```

```
Number of objects created: 3

=== Code Execution Successful ===
```

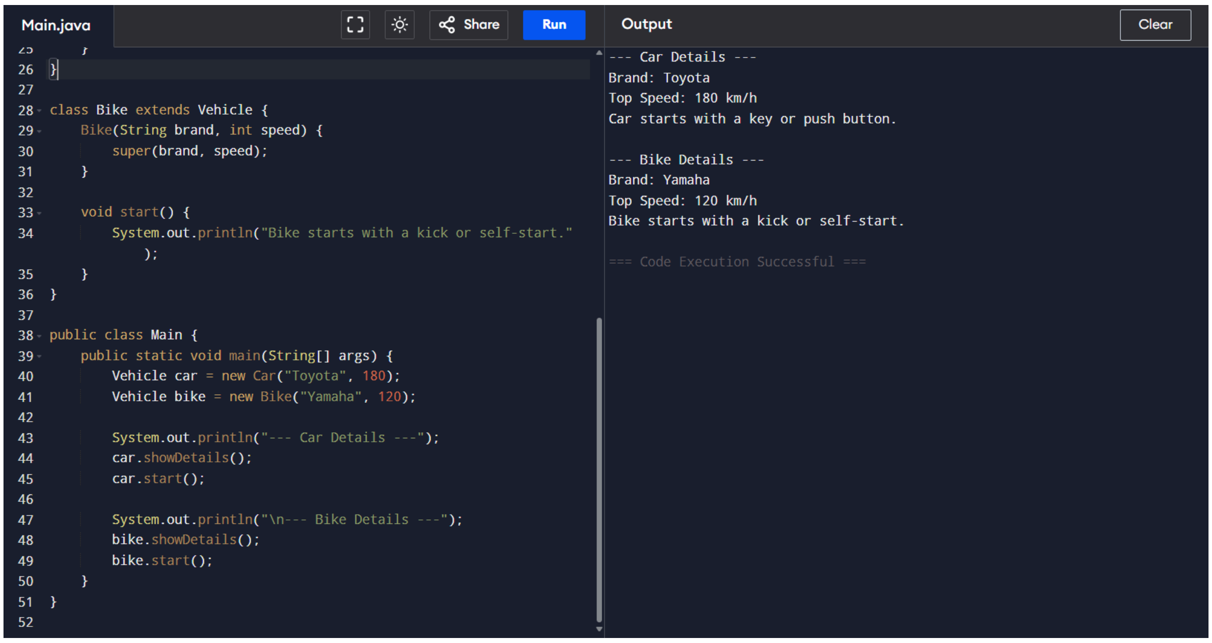
27.Design a class hierarchy using inheritance: Person → Employee → Manager.



```java
class Manager extends Employee {
    private String department;
    private double bonus;

    public Manager(String name, int age, String employeeId, double salary,
                   String department, double bonus) {
        super(name, age, employeeId, salary); // call Employee's constructor
        this.department = department;
        this.bonus = bonus;
    }

    public void displayManager() {
        displayEmployee();  // show employee info first
        System.out.println("Department: " + department);
        System.out.println("Bonus: Rs" + bonus);
    }
}

public class Main {
    public static void main(String[] args) {
        // create a Manager (inherits all fields)
        Manager m = new Manager("SSK", 21, "7517882753", 75000,
                                "Analytics", 15000);
        m.displayManager();
    }
}
```

```
Name: SSK
Age: 21
Employee ID: 7517882753
Salary: Rs75000.0
Department: Analytics
Bonus: Rs15000.0

=== Code Execution Successful ===
```

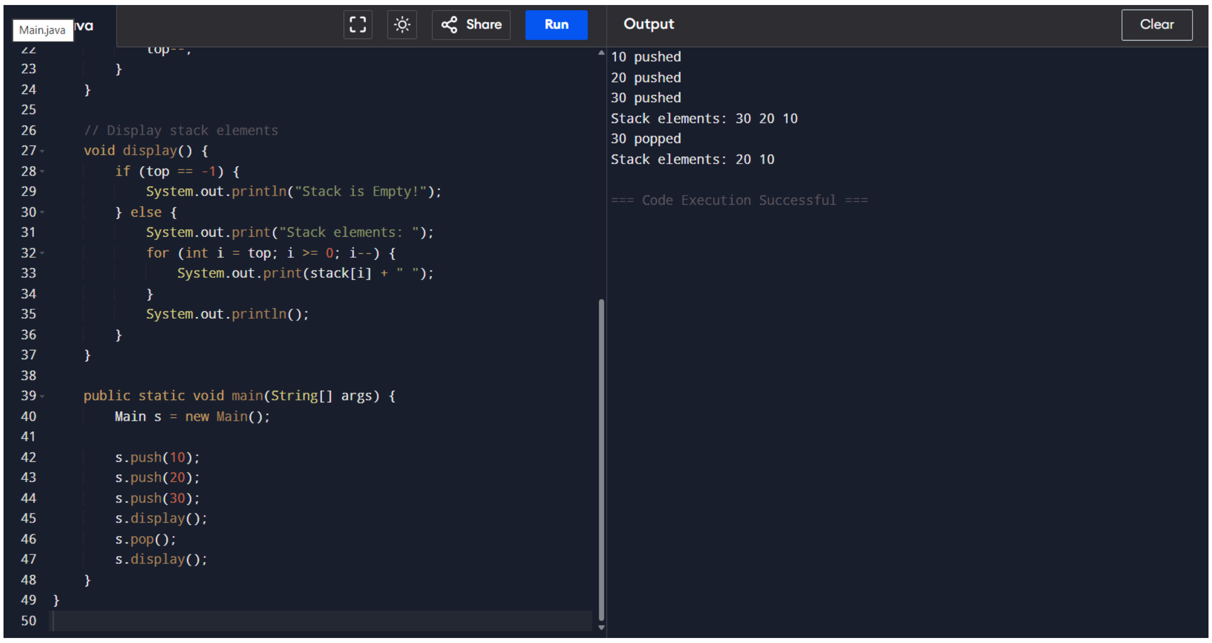
29.Use abstract classes to design a Vehicle class with car and bike subclasses.



```java
    }
}

class Bike extends Vehicle {
    Bike(String brand, int speed) {
        super(brand, speed);
    }

    void start() {
        System.out.println("Bike starts with a kick or self-start.");
    }
}

public class Main {
    public static void main(String[] args) {
        Vehicle car = new Car("Toyota", 180);
        Vehicle bike = new Bike("Yamaha", 120);

        System.out.println("--- Car Details ---");
        car.showDetails();
        car.start();

        System.out.println("\n--- Bike Details ---");
        bike.showDetails();
        bike.start();
    }
}
```

```
--- Car Details ---
Brand: Toyota
Top Speed: 180 km/h
Car starts with a key or push button.

--- Bike Details ---
Brand: Yamaha
Top Speed: 120 km/h
Bike starts with a kick or self-start.

=== Code Execution Successful ===
```

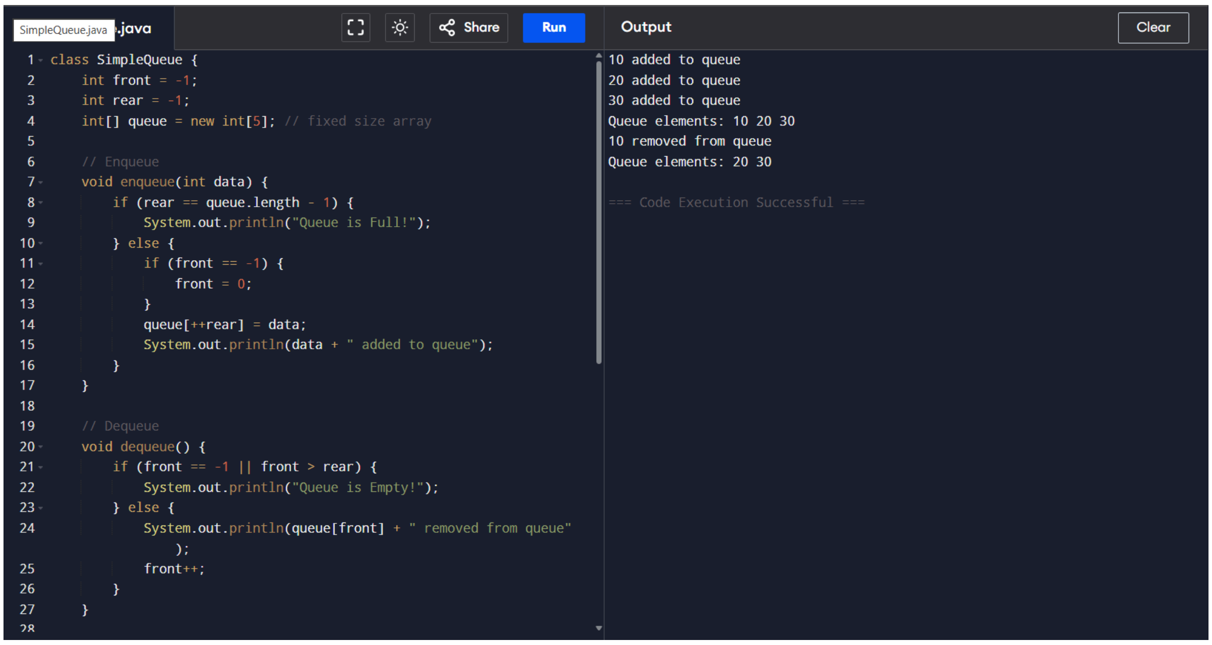
30.Implement a stack using an array with push, pop, and display operations.



```java
            top--;
        }
    }

    // Display stack elements
    void display() {
        if (top == -1) {
            System.out.println("Stack is Empty!");
        } else {
            System.out.print("Stack elements: ");
            for (int i = top; i >= 0; i--) {
                System.out.print(stack[i] + " ");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Main s = new Main();

        s.push(10);
        s.push(20);
        s.push(30);
        s.display();
        s.pop();
        s.display();
    }
}
```

```
10 pushed
20 pushed
30 pushed
Stack elements: 30 20 10
30 popped
Stack elements: 20 10

=== Code Execution Successful ===
```

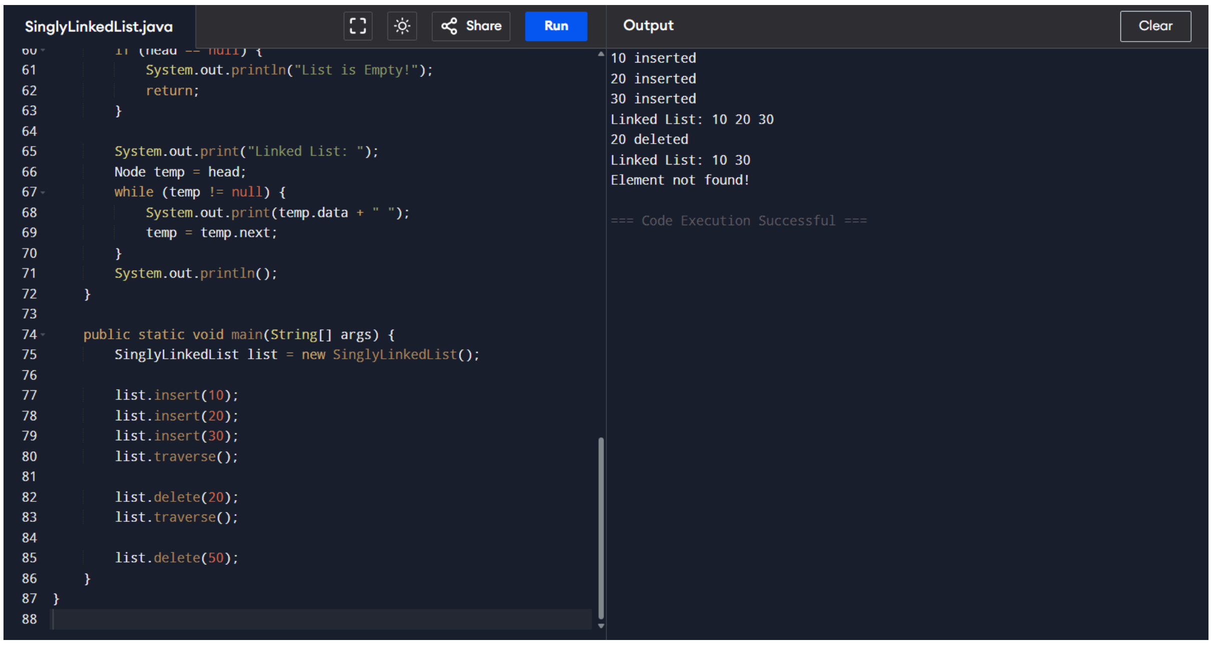
31.Create a queue using an array with enqueue and dequeue operations.



```java
class SimpleQueue {
    int front = -1;
    int rear = -1;
    int[] queue = new int[5]; // fixed size array

    // Enqueue
    void enqueue(int data) {
        if (rear == queue.length - 1) {
            System.out.println("Queue is Full!");
        } else {
            if (front == -1) {
                front = 0;
            }
            queue[++rear] = data;
            System.out.println(data + " added to queue");
        }
    }

    // Dequeue
    void dequeue() {
        if (front == -1 || front > rear) {
            System.out.println("Queue is Empty!");
        } else {
            System.out.println(queue[front] + " removed from queue"
                );
            front++;
        }
    }
```

```
10 added to queue
20 added to queue
30 added to queue
Queue elements: 10 20 30
10 removed from queue
Queue elements: 20 30

=== Code Execution Successful ===
```

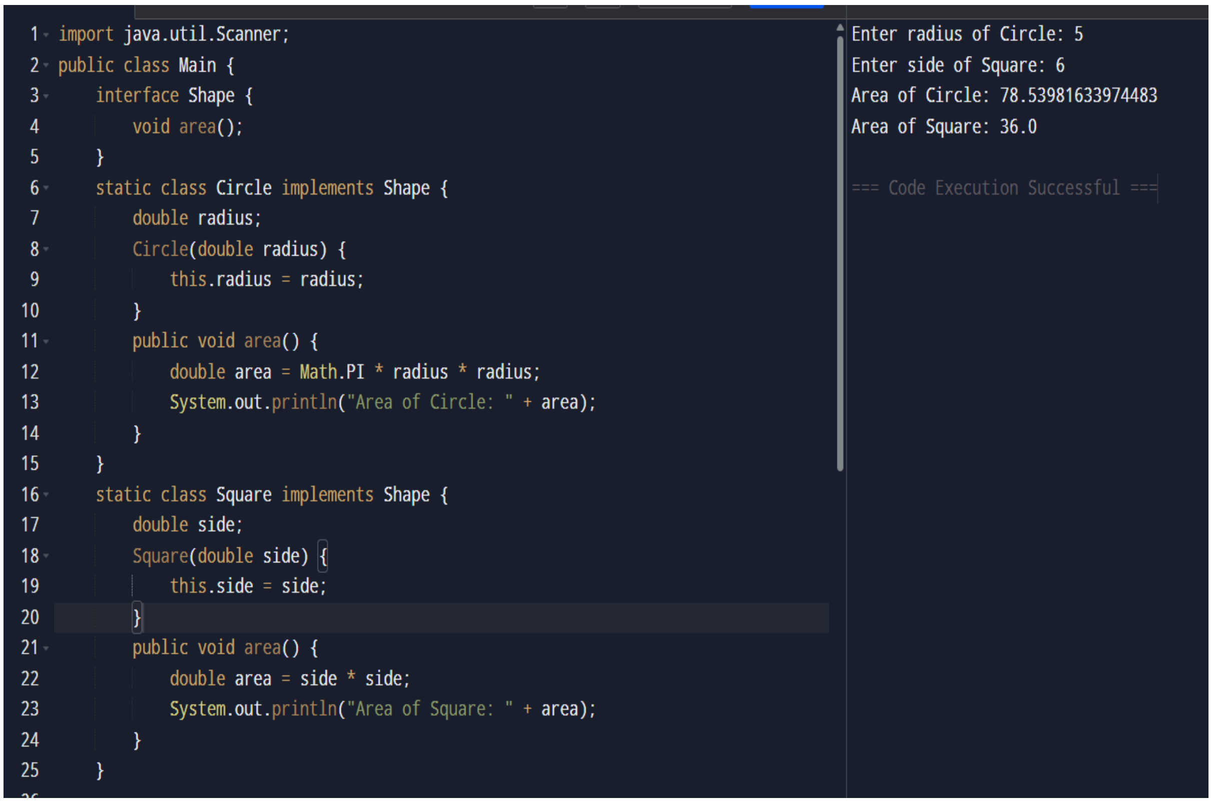
32.Design a singly linked list with insert, delete, and traverse methods.



```java
        if (head == null) {
            System.out.println("List is Empty!");
            return;
        }

        System.out.print("Linked List: ");
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {
        SinglyLinkedList list = new SinglyLinkedList();

        list.insert(10);
        list.insert(20);
        list.insert(30);
        list.traverse();

        list.delete(20);
        list.traverse();

        list.delete(50);
    }
}

```

```
10 inserted
20 inserted
30 inserted
Linked List: 10 20 30
20 deleted
Linked List: 10 30
Element not found!

=== Code Execution Successful ===
```

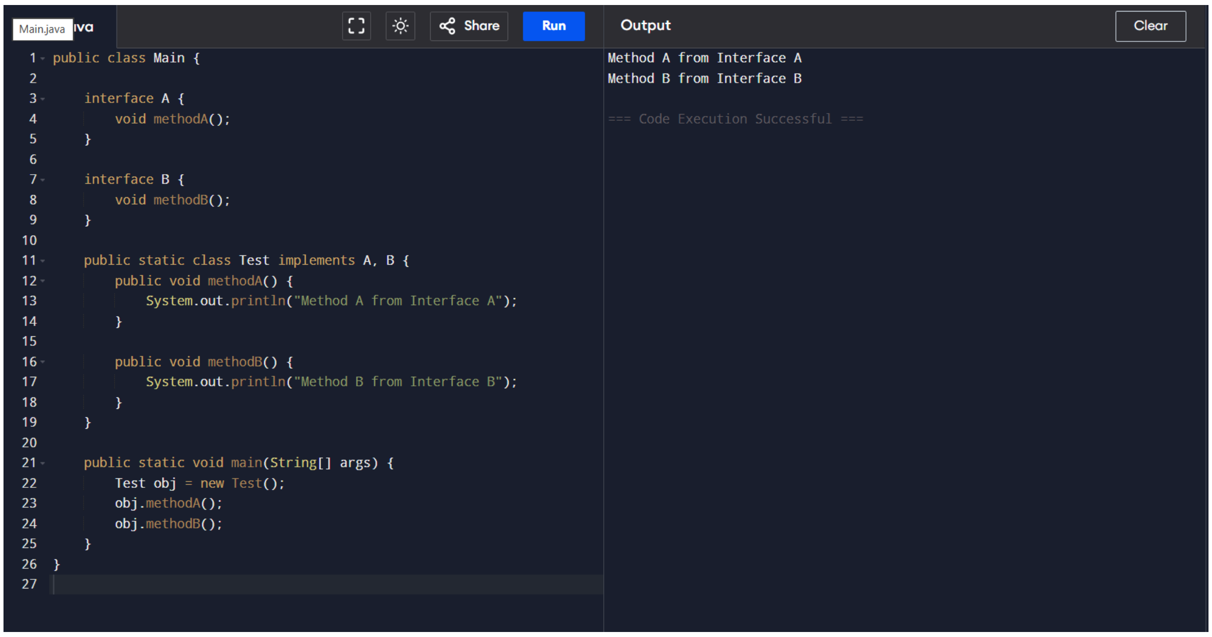
33.Create an interface `Shape` with a method `area()` and implement it in `Circle`, `Square`.



```java
import java.util.Scanner;
public class Main {
    interface Shape {
        void area();
    }
    static class Circle implements Shape {
        double radius;
        Circle(double radius) {
            this.radius = radius;
        }
        public void area() {
            double area = Math.PI * radius * radius;
            System.out.println("Area of Circle: " + area);
        }
    }
    static class Square implements Shape {
        double side;
        Square(double side) {
            this.side = side;
        }
        public void area() {
            double area = side * side;
            System.out.println("Area of Square: " + area);
        }
    }
```

```
Enter radius of Circle: 5
Enter side of Square: 6
Area of Circle: 78.53981633974483
Area of Square: 36.0

=== Code Execution Successful ===
```

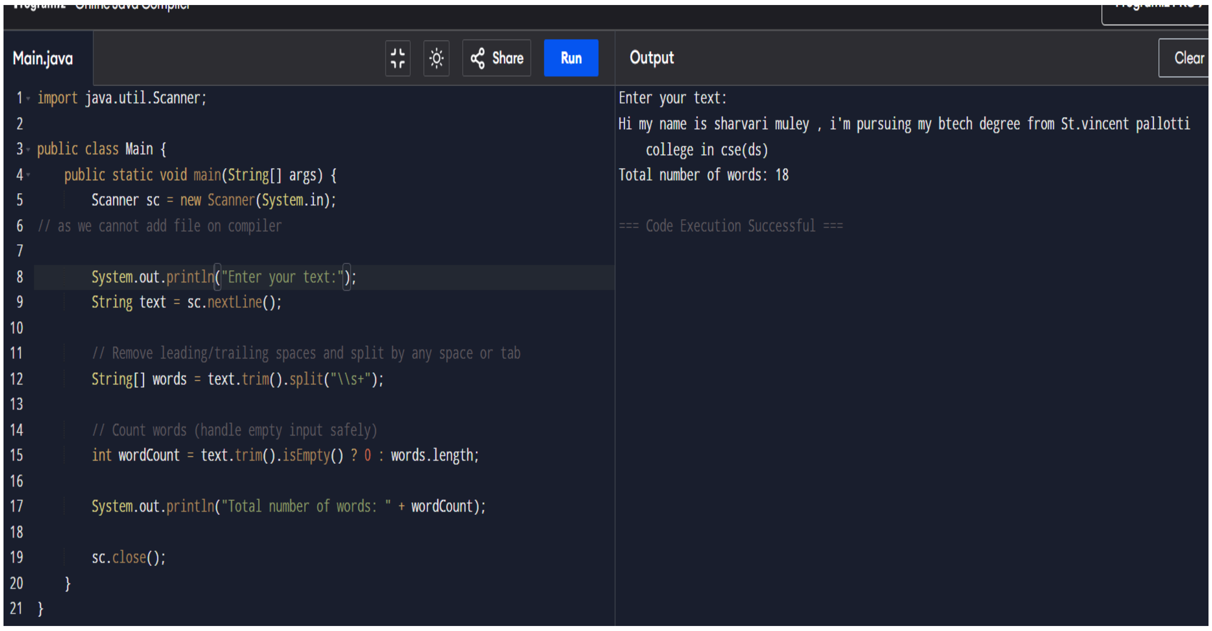
34.Demonstrate multiple inheritance in Java using two interfaces and one implementing class.



```java
public class Main {

    interface A {
        void methodA();
    }

    interface B {
        void methodB();
    }

    public static class Test implements A, B {
        public void methodA() {
            System.out.println("Method A from Interface A");
        }

        public void methodB() {
            System.out.println("Method B from Interface B");
        }
    }

    public static void main(String[] args) {
        Test obj = new Test();
        obj.methodA();
        obj.methodB();
    }
}
```

```
Method A from Interface A
Method B from Interface B

=== Code Execution Successful ===
```

36.Develop a program to count the number of words in a text file.



```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
// as we cannot add file on compiler

        System.out.println("Enter your text:");
        String text = sc.nextLine();

        // Remove leading/trailing spaces and split by any space or tab
        String[] words = text.trim().split("\\s+");

        // Count words (handle empty input safely)
        int wordCount = text.trim().isEmpty() ? 0 : words.length;

        System.out.println("Total number of words: " + wordCount);

        sc.close();
    }
}
```

```
Enter your text:
Hi my name is sharvari muley , i'm pursuing my btech degree from St.vincent pallotti
    college in cse(ds)
Total number of words: 18

=== Code Execution Successful ===
```