1. Write a java program to read a character until a \* is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users, java code, pseudo code include start, end and procedure inside it

Pseudo Code:

START

PROCEDURE countCharacters()

INITIALIZE uppercaseCount TO 0

INITIALIZE lowercaseCount TO 0

INITIALIZE numberCount TO 0

DECLARE char AS CHARACTER

PRINT "Enter characters, end with '\*'"

DO

READ char // Read character from user

IF char IS UPPERCASE

INCREMENT uppercaseCount

ELSE IF char IS LOWERCASE

INCREMENT lowercaseCount

ELSE IF char IS DIGIT

INCREMENT numberCount

END IF

WHILE char IS NOT '\*'

PRINT "Uppercase count: " + uppercaseCount

PRINT "Lowercase count: " + lowercaseCount

PRINT "Number count: " + numberCount

END PROCEDURE

PROCEDURE Main()

CALL countCharacters()

END PROCEDURE

END

Java code:

import java.util.Scanner;

public class CharacterCount {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int upperCount = 0, lowerCount = 0, numberCount = 0;

char input;

System.out.println("Enter characters (enter '\*' to stop):");

while (true) {

input = scanner.next().charAt(0);

if (input == '\*') break;

if (Character.isUpperCase(input)) upperCount++;

else if (Character.isLowerCase(input)) lowerCount++;

else if (Character.isDigit(input)) numberCount++;

}

System.out.println("Uppercase letters: " + upperCount);

System.out.println("Lowercase letters: " + lowerCount);

System.out.println("Numbers: " + numberCount);

}

}

2. Write a Program to create an array with the First Element as the Number and Second Element as the Square of the Number.

Pseudo code:

START

READ number

DECLARE array AS INTEGER[2]

SET array[0] TO number

SET array[1] TO number \* number

PRINT array[0], array[1]

END

Java code:

import java.util.Scanner;

public class NumberSquare {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

int[] array = {number, number \* number};

System.out.println("Number: " + array[0]);

System.out.println("Square: " + array[1]);

}

}

3. Develop a JAVA code to display the balance. Include the following members: Design a class to represent a bank account.

Data Members: Name of the depositor, Account number, Type of account(Savings/Current), Balance amount in the account(Minimum balance is Rs.500.00)

Methods: To read account number, Depositor name, Type of account.To deposit an amount (Deposited amount should be added with it)

To withdraw an amount after checking balance(Minimum balance must be Rs.500.00 Note : Assume that balance amount = 10000

Pseudo code:

START

CLASS BankAccount

DECLARE depositorName, accountNumber, accountType AS STRING

DECLARE balance AS DOUBLE = 10000.0

PROCEDURE readAccountDetails()

INPUT accountNumber, depositorName, accountType

END PROCEDURE

PROCEDURE deposit(amount)

balance += amount

PRINT "Deposited: " + amount

END PROCEDURE

PROCEDURE withdraw(amount)

IF balance - amount < 500 THEN

PRINT "Cannot withdraw. Minimum balance is Rs.500.00"

ELSE

balance -= amount

PRINT "Withdrawn: " + amount

END IF

END PROCEDURE

PROCEDURE displayBalance()

PRINT "Current Balance: " + balance

END PROCEDURE

END CLASS

PROCEDURE Main()

DECLARE account AS BankAccount

CALL account.readAccountDetails()

CALL account.deposit(2000)

CALL account.withdraw(500)

CALL account.displayBalance()

END PROCEDURE

END

Java code:

import java.util.Scanner;

class BankAccount {

private String depositorName;

private String accountNumber;

private String accountType;

private double balance = 10000.0;

public void readAccountDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Account Number: ");

accountNumber = scanner.nextLine();

System.out.print("Enter Depositor Name: ");

depositorName = scanner.nextLine();

System.out.print("Enter Account Type (Savings/Current): ");

accountType = scanner.nextLine();

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount);

}

public void withdraw(double amount) {

if (balance - amount < 500) {

System.out.println("Cannot withdraw. Minimum balance must be Rs.500.00");

} else {

balance -= amount;

System.out.println("Withdrawn: " + amount);

}

}

public void displayBalance() {

System.out.println("Current Balance: " + balance);

}

}

public class Main {

public static void main(String[] args) {

BankAccount account = new BankAccount();

account.readAccountDetails();

account.deposit(2000);

account.withdraw(500);

account.displayBalance();

}

}

4. Write a program that accepts a string from user and displays the same string after removing vowels from it.

Pseudo code:

START

READ input STRING

SET output TO input with vowels removed

PRINT output

END

Java code:

import java.util.Scanner;

public class RemoveVowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

String output = input.replaceAll("[AEIOUaeiou]", "");

System.out.println("String after removing vowels: " + output);

}

}

5. Develop a code to Reverse and Add a Number until you get a Palindrome.

Pseudo code:

START

READ number

WHILE TRUE

SET reversed TO reverse(number)

number += reversed

PRINT "New number: " + number

IF number is palindrome THEN BREAK

END WHILE

END

Java code:

import java.util.Scanner;

public class ReverseAndAdd {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

while (true) {

int reversed = reverse(number);

number += reversed;

System.out.println("New number: " + number);

if (isPalindrome(number)) break;

}

}

private static int reverse(int num) {

int rev = 0;

while (num != 0) {

rev = rev \* 10 + num % 10;

num /= 10;

}

return rev;

}

private static boolean isPalindrome(int num) {

return num == reverse(num);

}

}

6. Bank is a class that provides method to get the rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS banks are providing 8.4%, 7.3% and 9.7% rate of interest. Write a

Java program for above scenario.

Pseudo code:

START

CLASS Bank

PROCEDURE getRateOfInterest()

RETURN 0.0

END PROCEDURE

END CLASS

CLASS SBI EXTENDS Bank

PROCEDURE getRateOfInterest()

RETURN 8.4

END PROCEDURE

END CLASS

CLASS ICICI EXTENDS Bank

PROCEDURE getRateOfInterest()

RETURN 7.3

END PROCEDURE

END CLASS

CLASS AXIS EXTENDS Bank

PROCEDURE getRateOfInterest()

RETURN 9.7

END PROCEDURE

END CLASS

PROCEDURE Main()

DECLARE sbi AS SBI

DECLARE icici AS ICICI

DECLARE axis AS AXIS

PRINT sbi.getRateOfInterest()

PRINT icici.getRateOfInterest()

PRINT axis.getRateOfInterest()

END PROCEDURE

END

Java code:

class Bank {

public double getRateOfInterest() {

return 0.0;

}

}

class SBI extends Bank {

@Override

public double getRateOfInterest() {

return 8.4;

}

}

class ICICI extends Bank {

@Override

public double getRateOfInterest() {

return 7.3;

}

}

class AXIS extends Bank {

@Override

public double getRateOfInterest() {

return 9.7;

}

}

public class Main {

public static void main(String[] args) {

Bank sbi = new SBI();

Bank icici = new ICICI();

Bank axis = new AXIS();

System.out.println("SBI Rate of Interest: " + sbi.getRateOfInterest());

System.out.println("ICICI Rate of Interest: " + icici.getRateOfInterest());

System.out.println("AXIS Rate of Interest: " + axis.getRateOfInterest());

}

}

7. Create Customer class with deposit() and withdraw() as synchronized methods. Declare AccountNo, AccName and Balance as Instance Variables inside the class. From the main class, Input the amount for withdraw() operation and if requested amount is not available in existing Balance amount, withdraw() method should be temporarily suspended using wait() method until deposit() method receives the input for amount, to be added in the existing Balance amount and then withdraw() would be completed in a successful manner. Develop the above scenario using Synchronization and Inter-Thread Communication.

Note : existing Bank balance amount 10000

Pseudo code:

START

CLASS Customer

DECLARE accountNo, accName AS STRING

DECLARE balance AS DOUBLE = 10000

PROCEDURE withdraw(amount)

WHILE amount > balance

PRINT "Insufficient balance. Waiting for deposit..."

WAIT

END WHILE

balance -= amount

PRINT "Withdrawn: " + amount

END PROCEDURE

PROCEDURE deposit(amount)

balance += amount

PRINT "Deposited: " + amount

NOTIFY

END PROCEDURE

END CLASS

PROCEDURE Main()

DECLARE customer AS Customer

START NEW THREAD for withdraw

CALL customer.deposit(5000) AFTER DELAY

END PROCEDURE

END

Java code:

import java.util.Scanner;

class Customer {

private String accountNo;

private String accName;

private double balance = 10000;

public synchronized void withdraw(double amount) throws InterruptedException {

while (amount > balance) {

System.out.println("Insufficient balance. Waiting for deposit...");

wait(); // Wait for deposit

}

balance -= amount;

System.out.println("Withdrawn: " + amount);

}

public synchronized void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount);

notify(); // Notify waiting withdraw

}

}

public class Main {

public static void main(String[] args) {

Customer customer = new Customer();

new Thread(() -> {

try {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter amount to withdraw: ");

double amount = scanner.nextDouble();

customer.withdraw(amount);

} catch (InterruptedException e) {

e.printStackTrace();

}

}).start();

try {

Thread.sleep(1000); // Simulating delay

customer.deposit(5000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

8. Given an integer n, return a string array answer (1-indexed) where: answer[i] == "FizzBuzz" if i is divisible by 3 and 5. answer[i] == "Fizz" if i is divisible by 3.

answer[i] == "Buzz" if i is divisible by 5.

answer[i] == i (as a string) if none of the above conditions are true.

Pseudo code:

FUNCTION fizzBuzz(n)

CREATE array result of size n

FOR i FROM 1 TO n

IF i MOD 3 = 0 AND i MOD 5 = 0 THEN

result[i - 1] = "FizzBuzz"

ELSE IF i MOD 3 = 0 THEN

result[i - 1] = "Fizz"

ELSE IF i MOD 5 = 0 THEN

result[i - 1] = "Buzz"

ELSE

result[i - 1] = i as STRING

END IF

END FOR

RETURN result

END FUNCTION

Java code:

import java.util.ArrayList;

public class FizzBuzz {

public static String[] fizzBuzz(int n) {

String[] result = new String[n];

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

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

result[i - 1] = "FizzBuzz";

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

result[i - 1] = "Fizz";

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

result[i - 1] = "Buzz";

} else {

result[i - 1] = String.valueOf(i);

}

}

return result;

}

}

9. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Pseudo code:

START

READ salary, grade

IF grade = 'A' THEN

bonus = salary \* 0.05

ELSE IF grade = 'B' THEN

bonus = salary \* 0.10

ELSE

PRINT "Invalid grade."

RETURN

END IF

IF salary < 10000 THEN

bonus += salary \* 0.02

END IF

totalSalary = salary + bonus

PRINT bonus, totalSalary

END

Java code:

import java.util.Scanner;

public class BonusCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter salary: ");

double salary = scanner.nextDouble();

System.out.print("Enter grade (A/B): ");

char grade = scanner.next().charAt(0);

double bonus;

if (grade == 'A') {

bonus = salary \* 0.05;

} else if (grade == 'B') {

bonus = salary \* 0.10;

} else {

System.out.println("Invalid grade.");

return;

}

if (salary < 10000) {

bonus += salary \* 0.02;

}

double totalSalary = salary + bonus;

System.out.println("Bonus: " + bonus);

System.out.println("Total Salary: " + totalSalary);

}

}

10. Given a string s consisting of words and spaces, return the length of the last word in the string. A word is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.

Pseudo code:  
FUNCTION lengthOfLastWord(s)

TRIM s

SPLIT s INTO words

RETURN LENGTH of last element in words

END FUNCTION

Java code:

public class LengthOfLastWord {

public static int lengthOfLastWord(String s) {

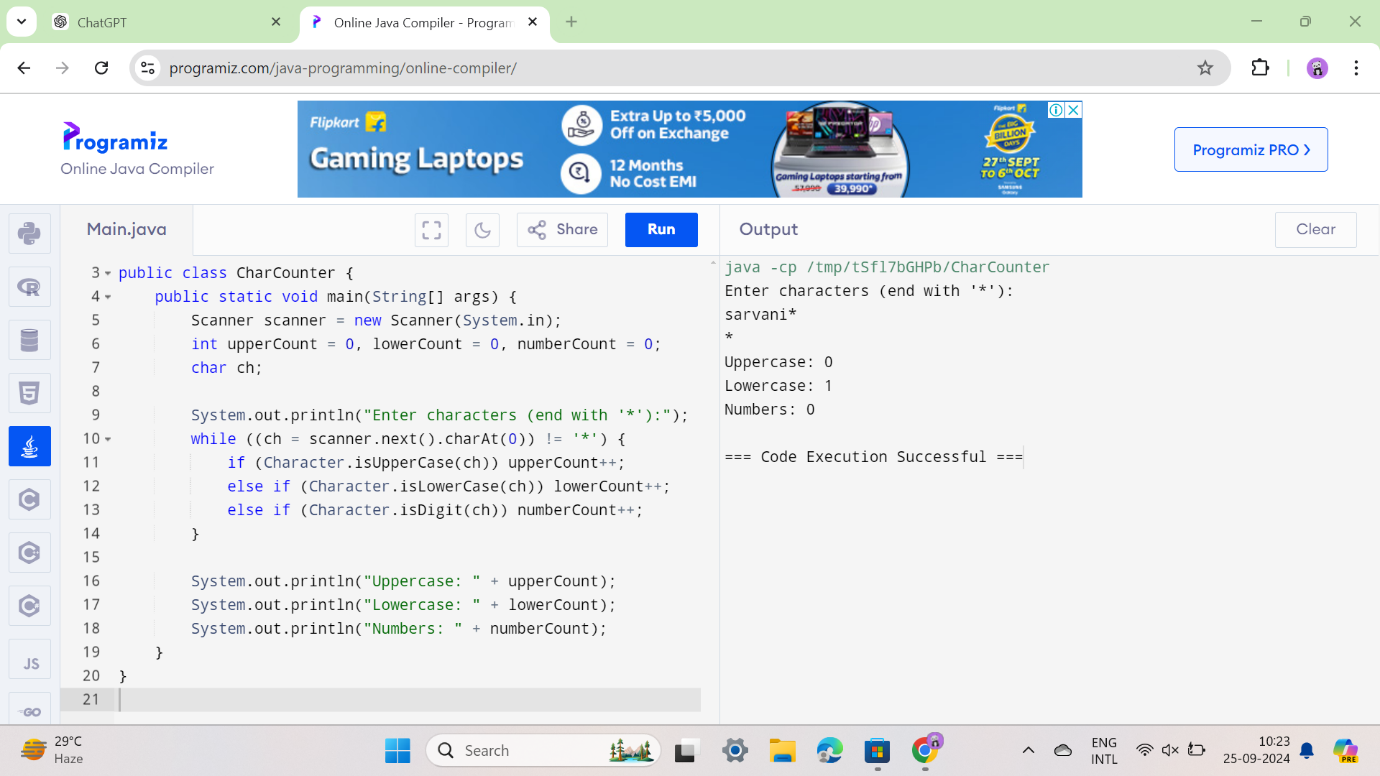
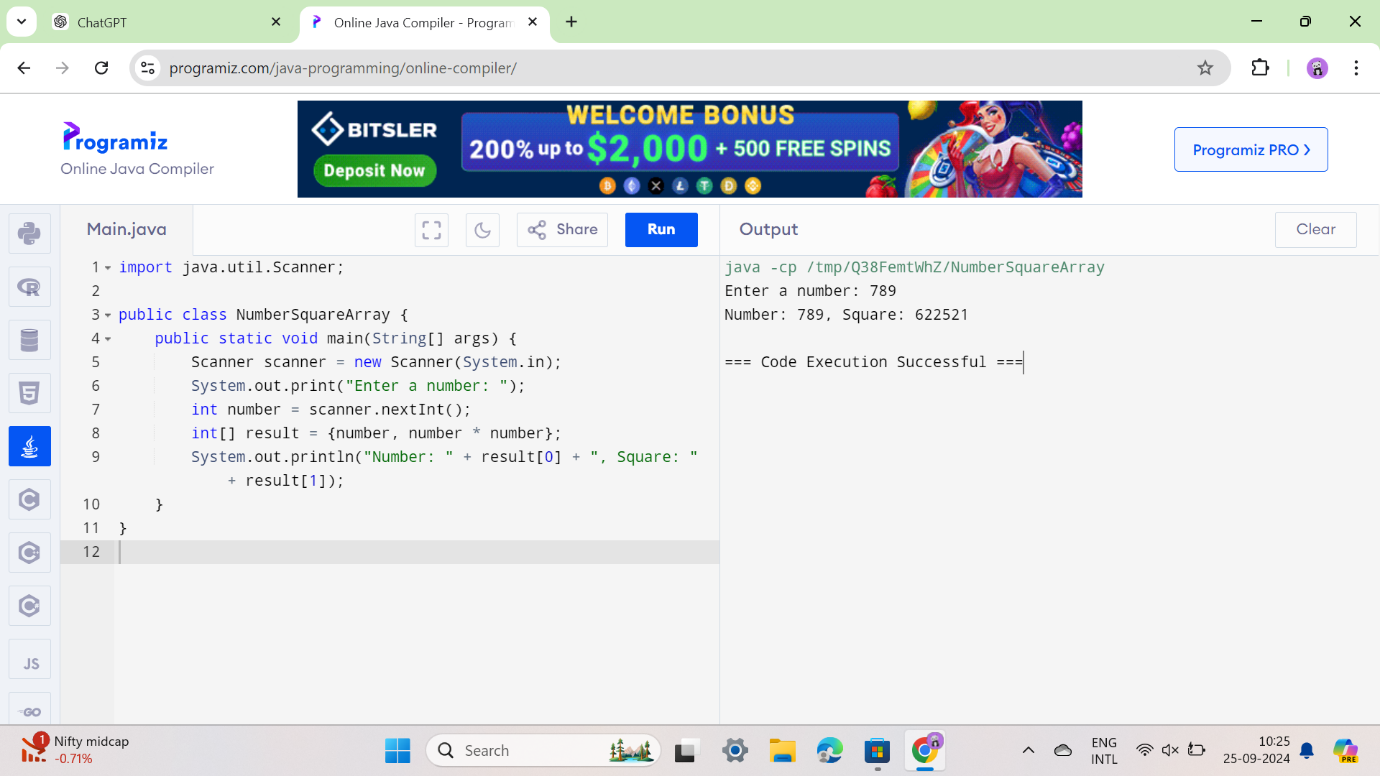
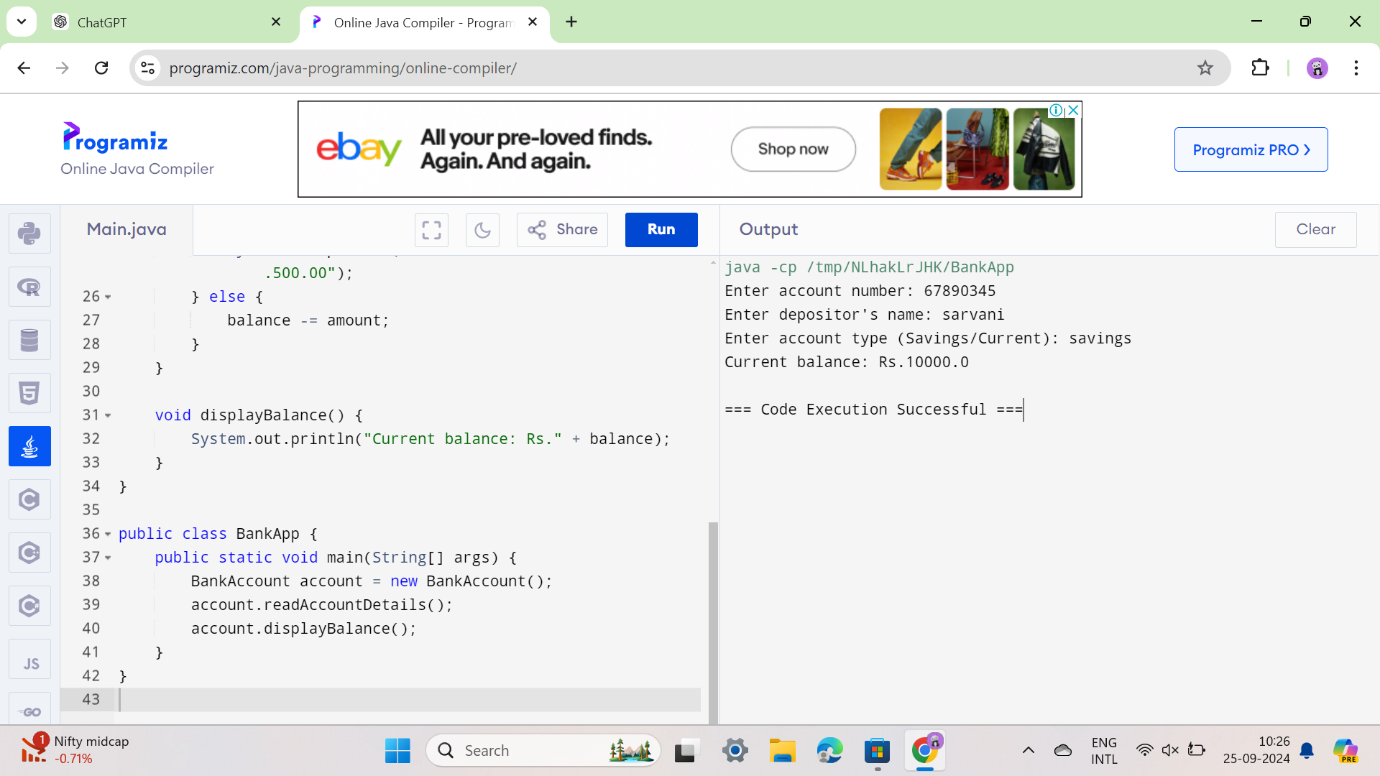
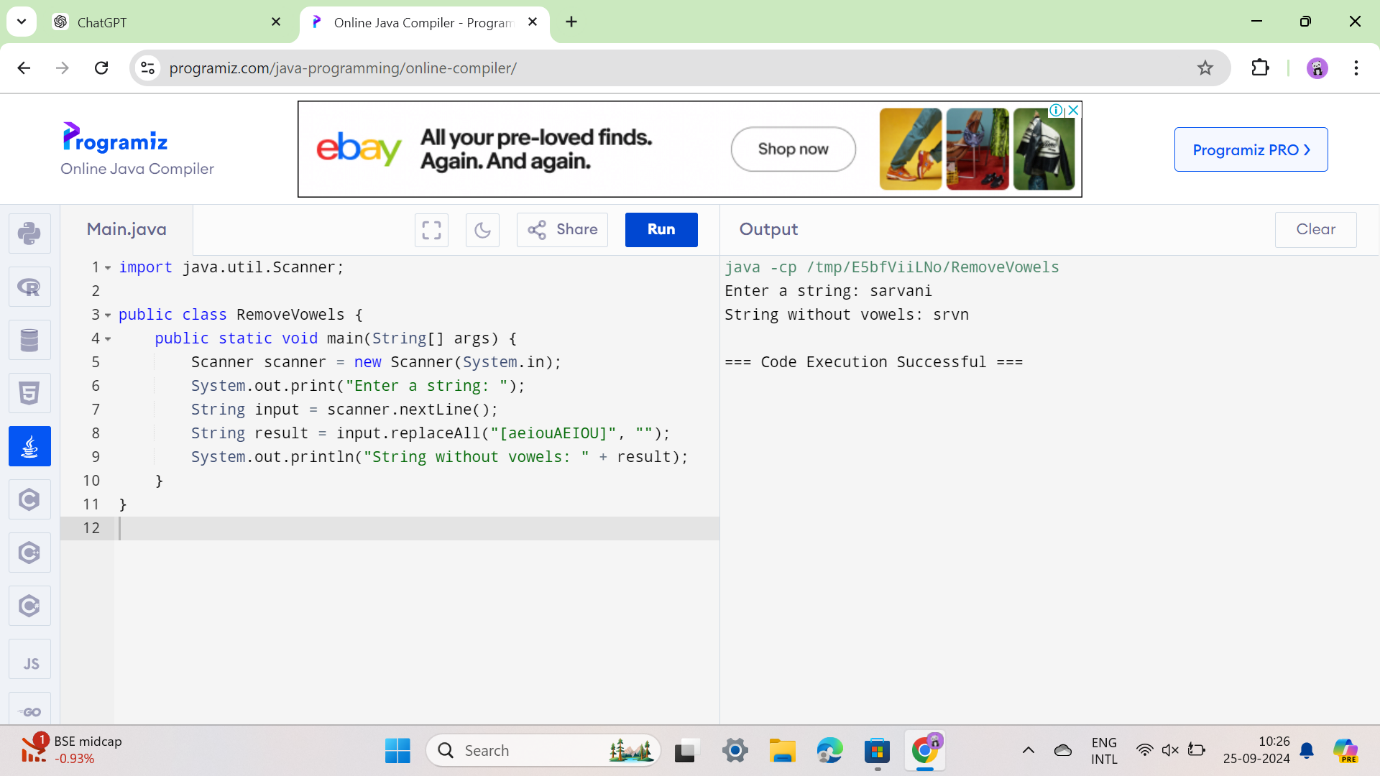
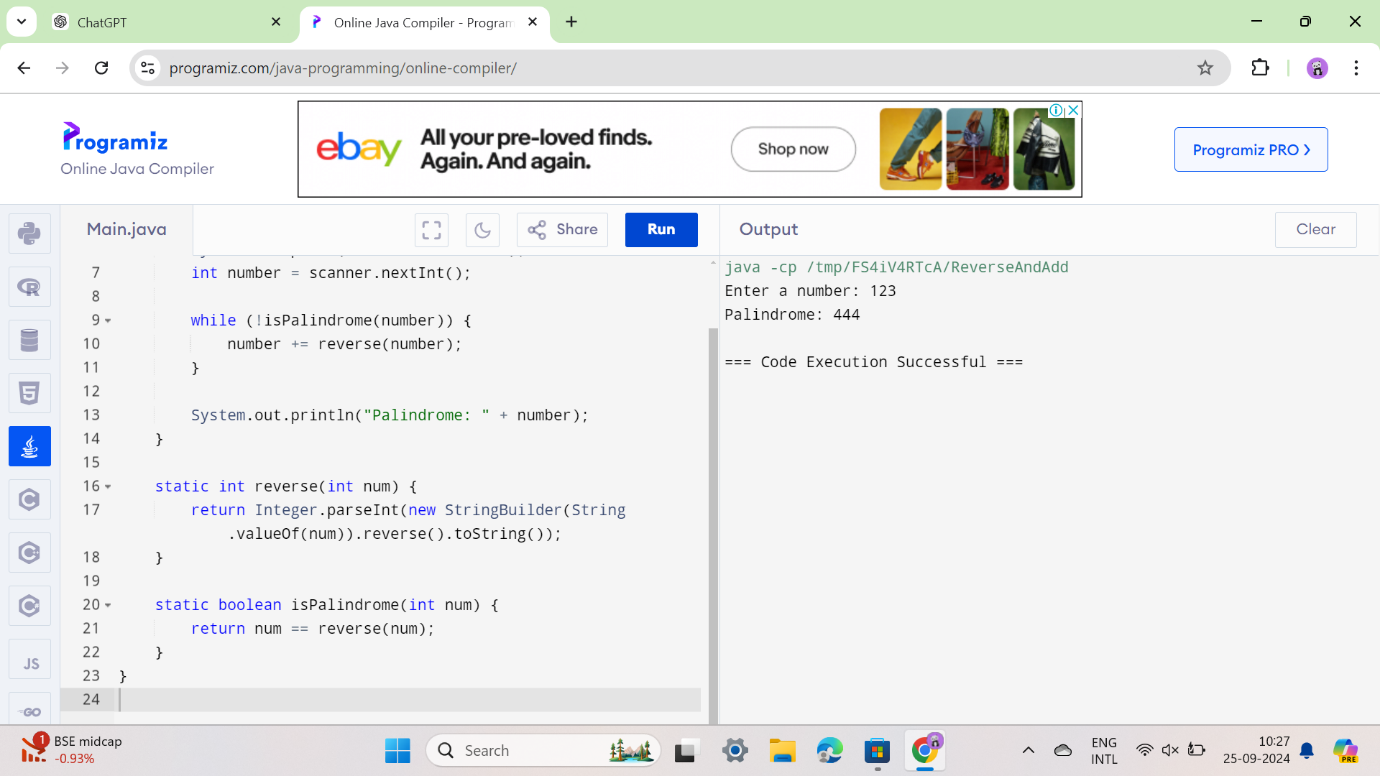
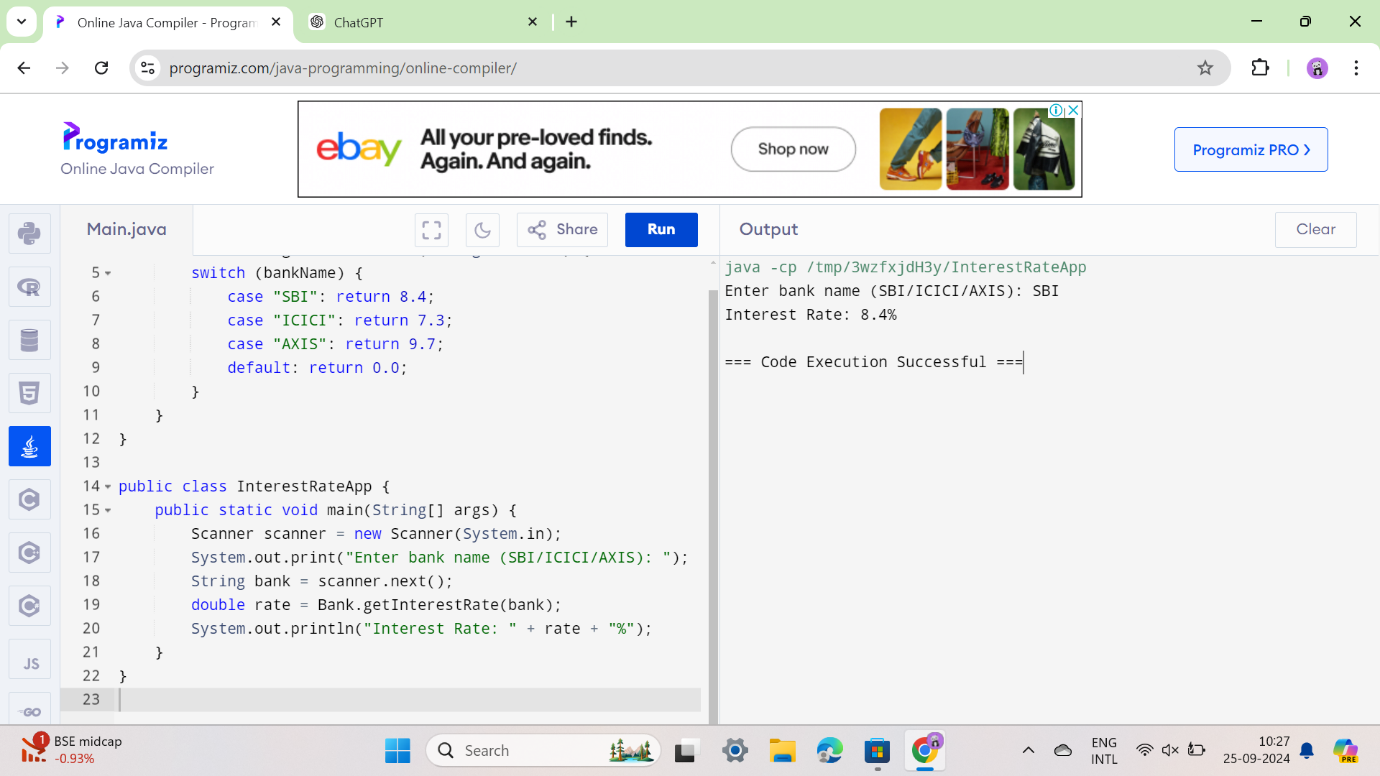
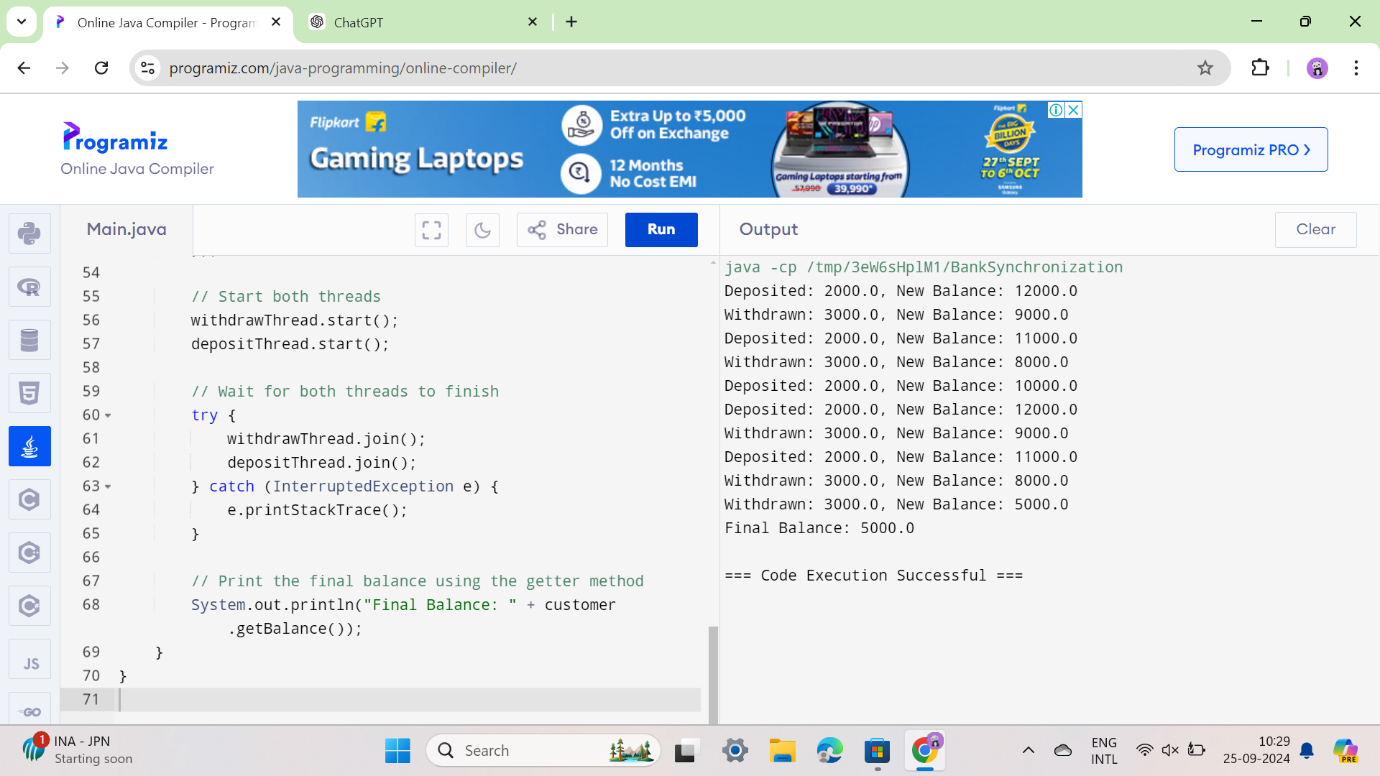
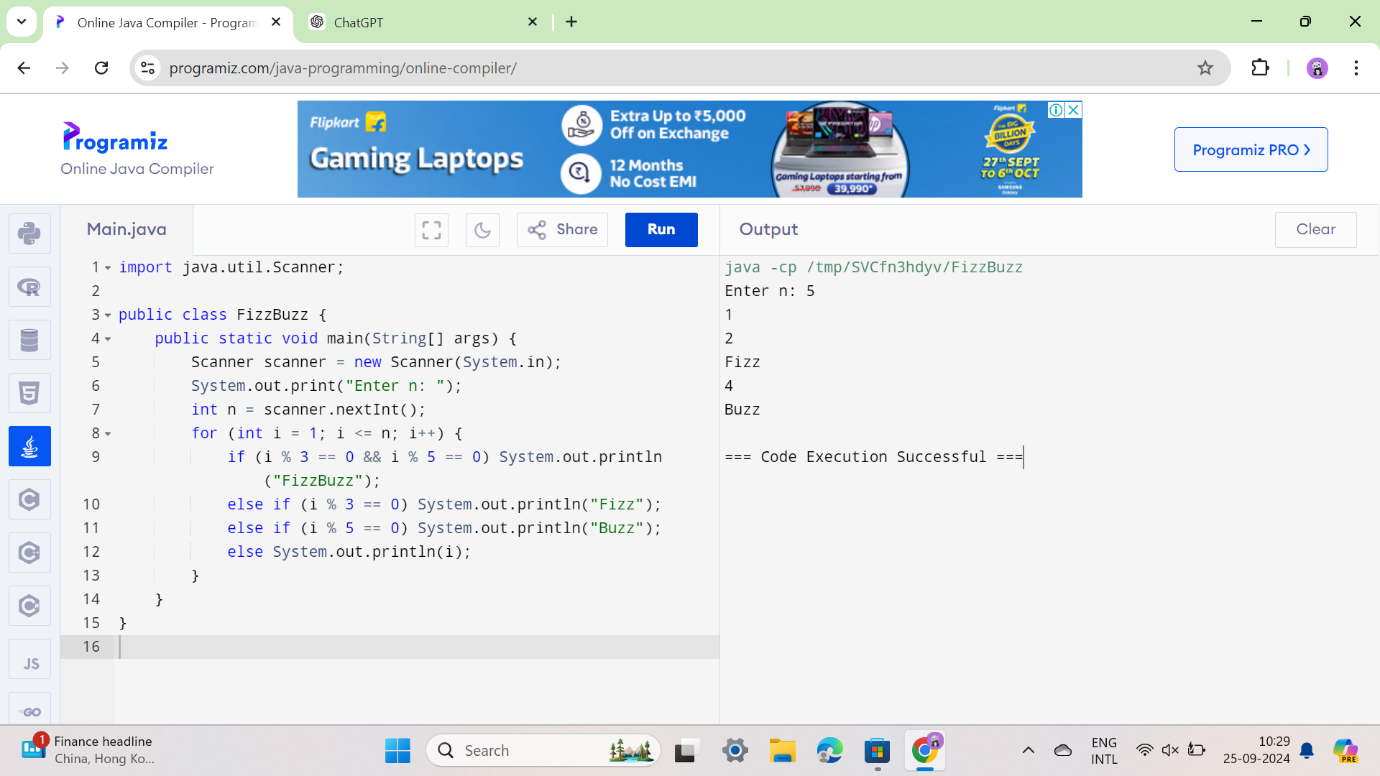
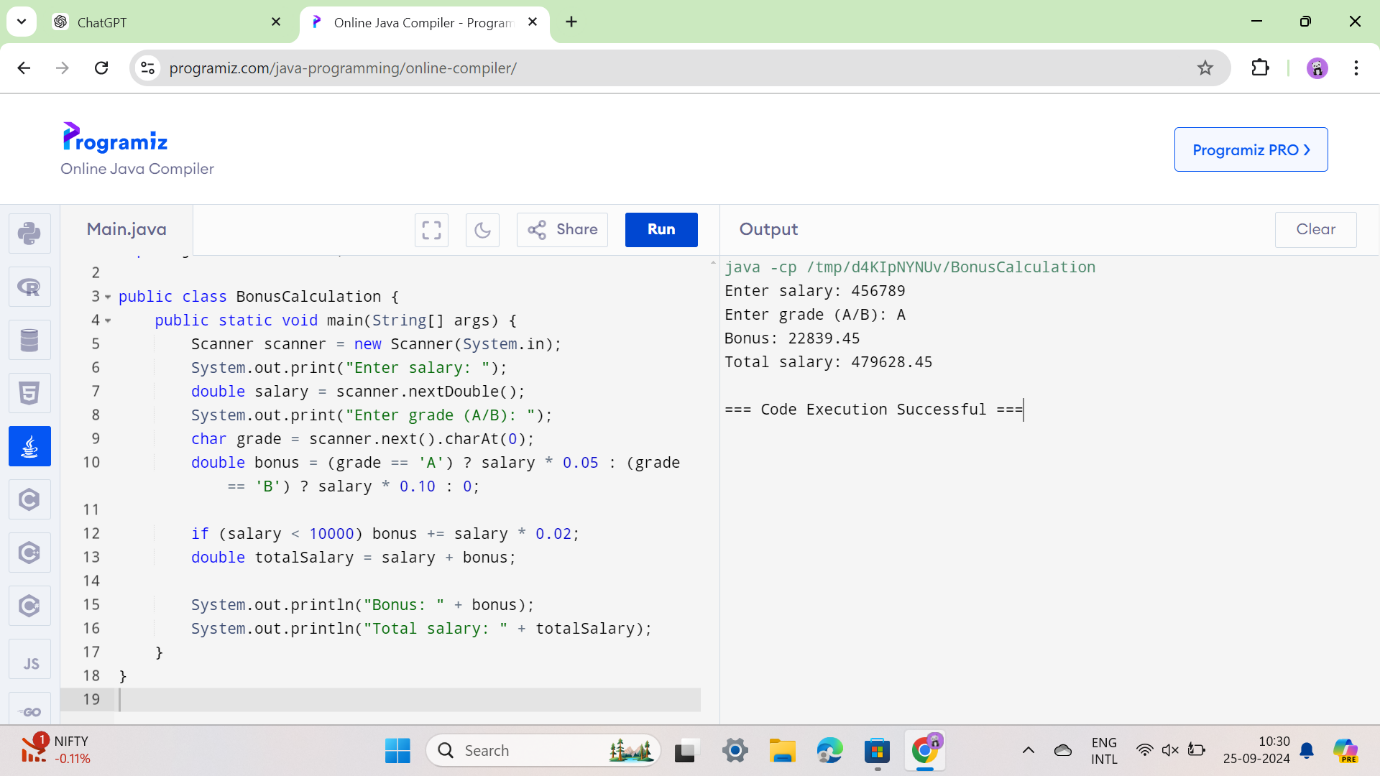
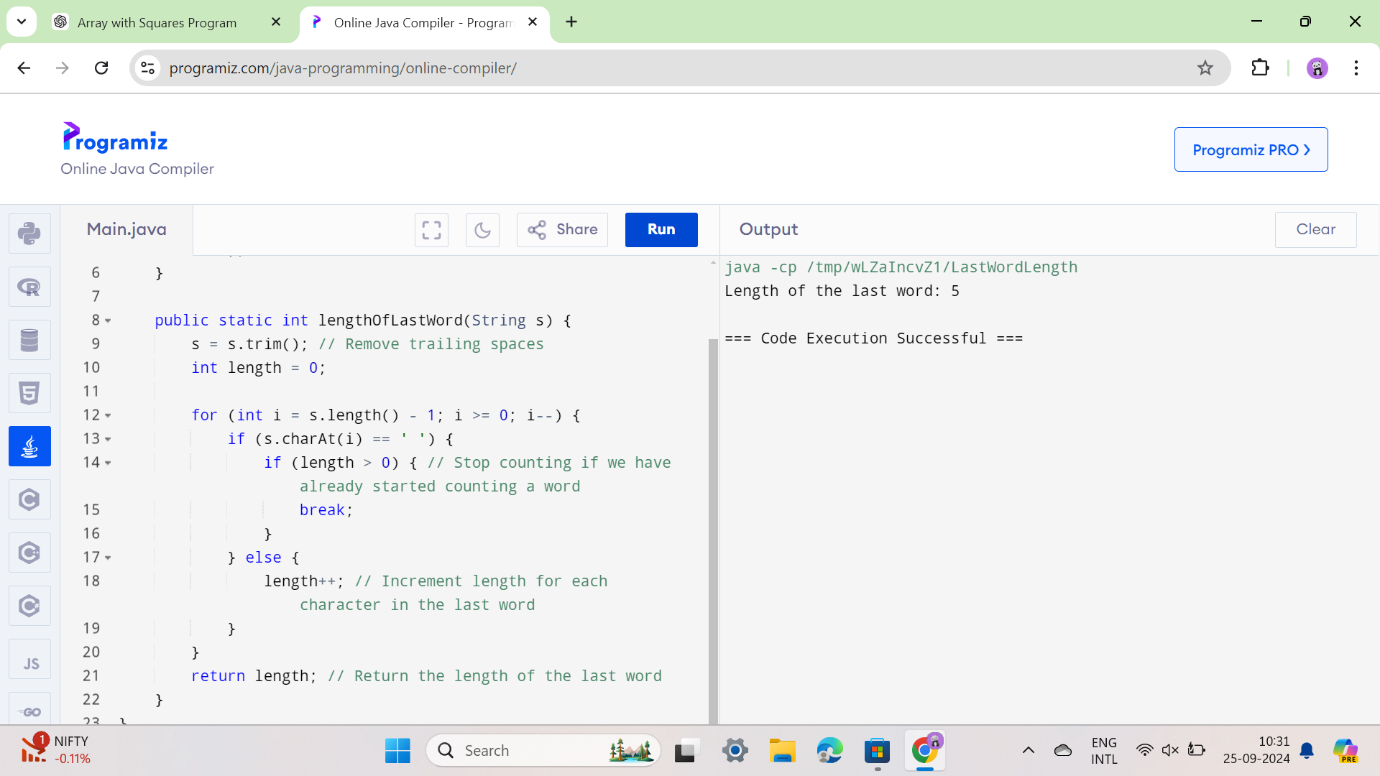
String[] words = s.trim().split(" ");

return words[words.length - 1].length();

}

}

**OUTPUTS-**

****