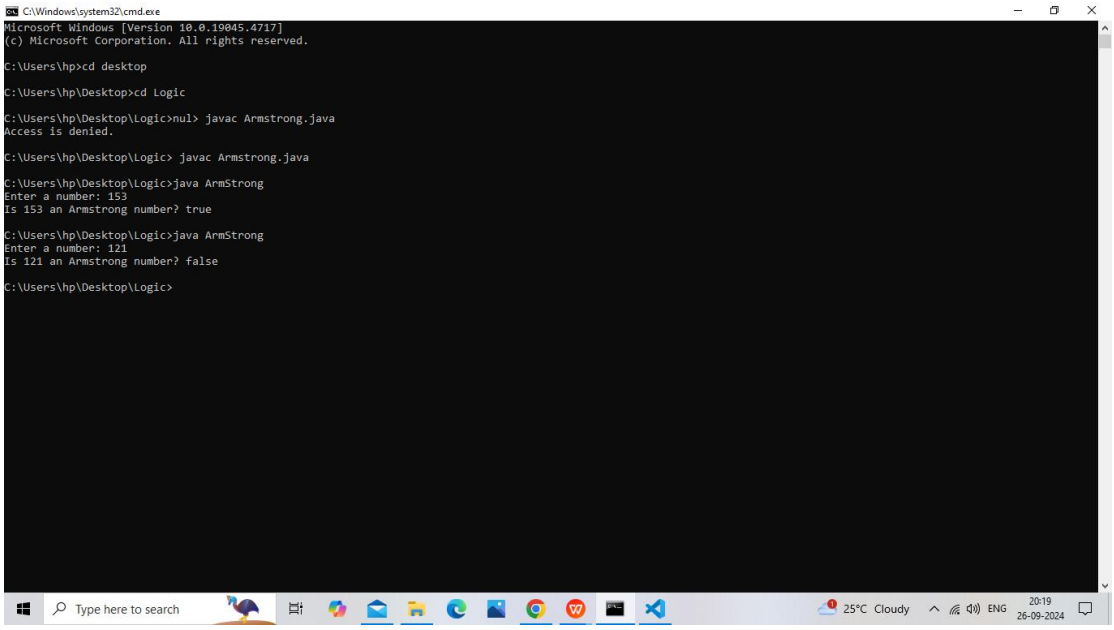
Question - 1

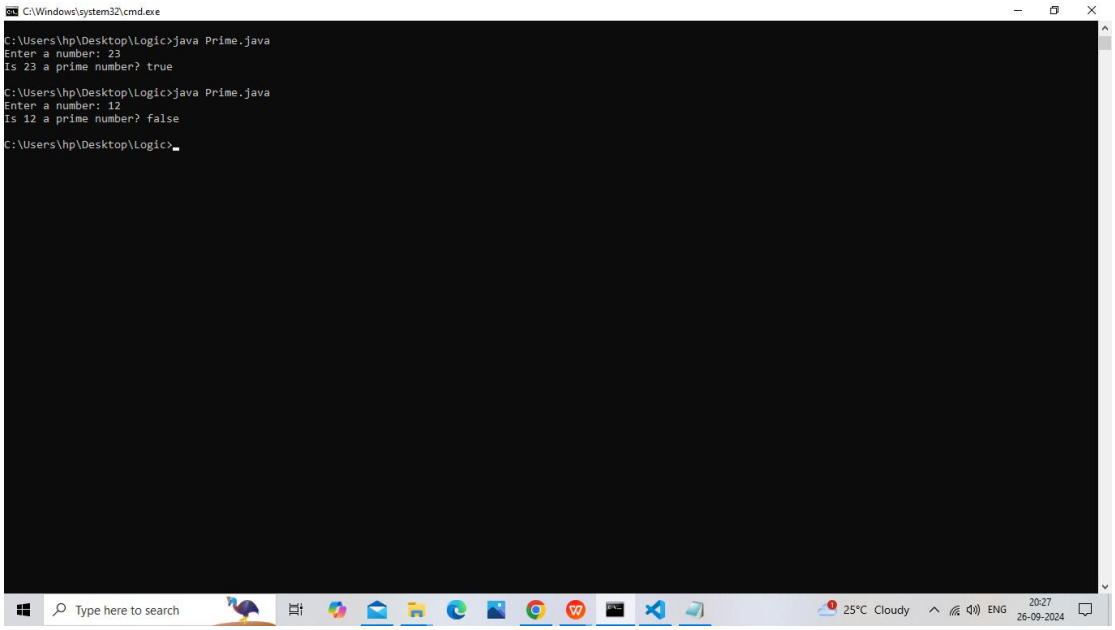
```
import java.util.Scanner;

public class ArmStrong {


   public static boolean isArmstrong(int number) {
      int originalNumber = number;
      int sum = 0;
      int digits = String.valueOf(number).length();


      while (number > 0) {
         int digit = number % 10;
         sum += Math.pow(digit, digits);
         number /= 10;
      }

      return sum == originalNumber;
   }

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


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


      boolean result = isArmstrong(number);


      System.out.println("Is " + number + " an Armstrong number? " + result);
   }
}
```



Question -2

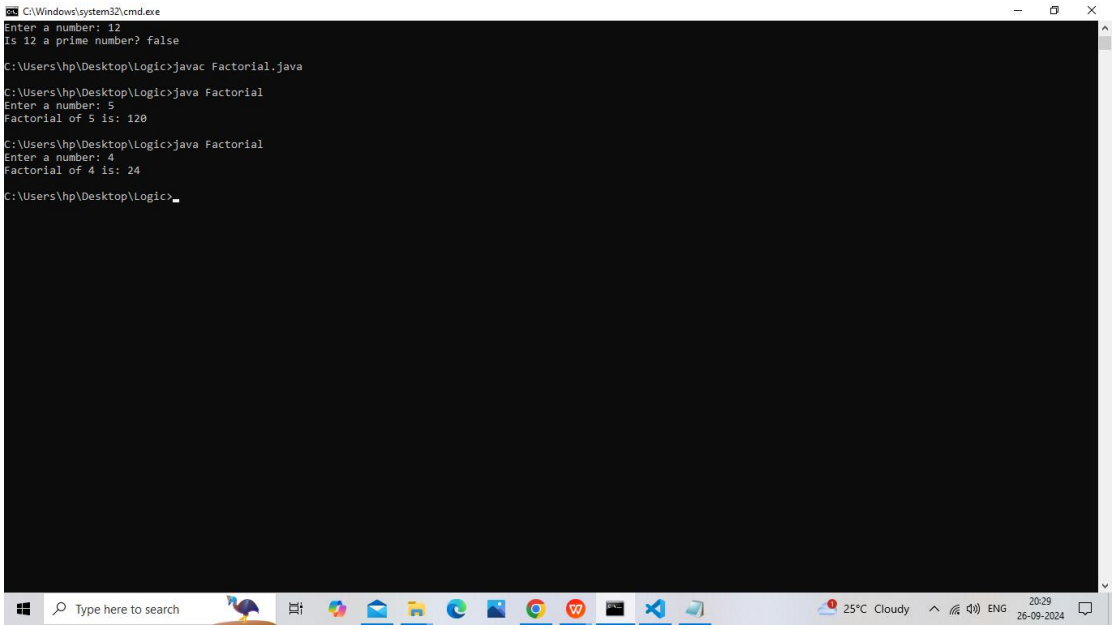
```
import java.util.Scanner;

public class Prime {


   public static boolean isPrime(int number) {

      if (number <= 1) {
         return false;
      }


      for (int i = 2; i <= Math.sqrt(number); i++) {
         if (number % i == 0) {
            return false;
         }
      }

      return true;
   }

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


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


      boolean result = isPrime(number);


      System.out.println("Is " + number + " a prime number? " + result);
```

```
    }
}
```



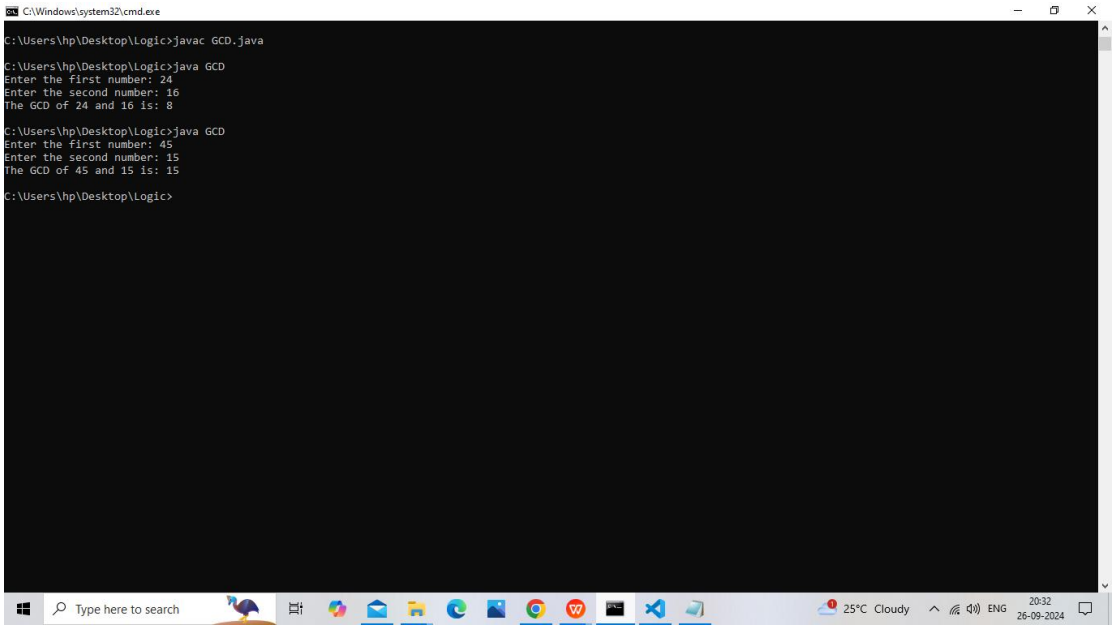
Question - 3

```
import java.util.Scanner;

public class Factorial {


   public static long factorial(int number) {
      long result = 1;


      for (int i = 2; i <= number; i++) {
         result *= i;
      }

      return result;
   }

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


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


      long result = factorial(number);


      System.out.println("Factorial of " + number + " is: " + result);
   }
}
```



Question - 4

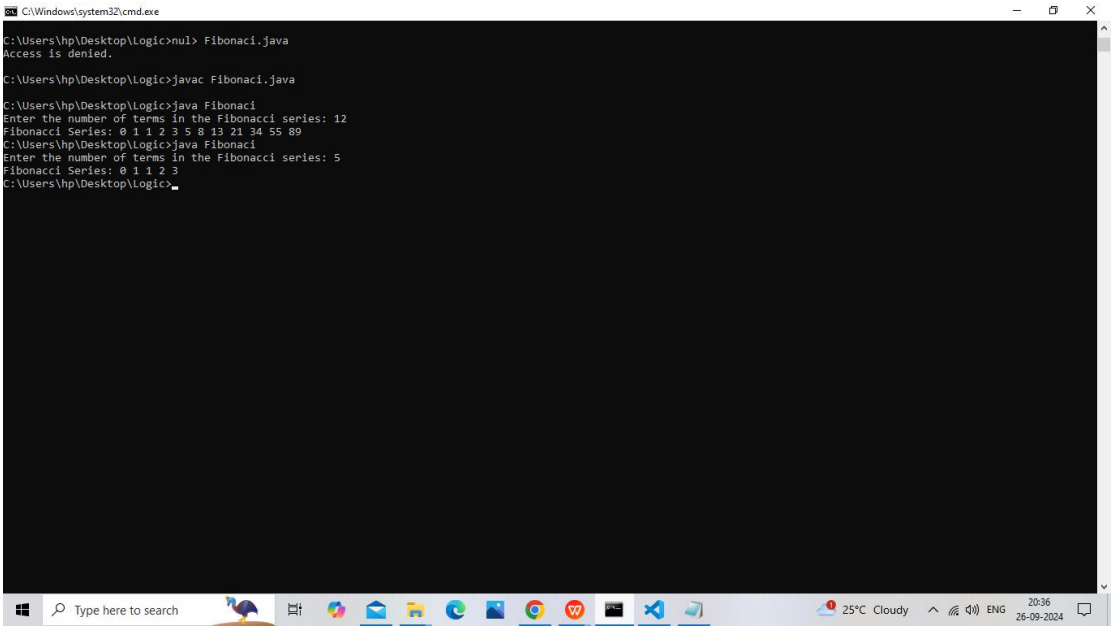
```
import java.util.Scanner;

public class GCD {


   public static int findGCD(int a, int b) {
      while (b != 0) {
         int temp = b;
         b = a % b;
         a = temp;
      }
      return a;
   }

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


      System.out.print("Enter the first number: ");
      int a = scanner.nextInt();
      System.out.print("Enter the second number: ");
      int b = scanner.nextInt();


      int gcd = findGCD(a, b);
      System.out.println("The GCD of " + a + " and " + b + " is: " + gcd);
   }
}
```



Question - 5

```
import java.util.Scanner;

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


      System.out.print("Enter the number of terms in the Fibonacci series: ");
      int n = scanner.nextInt();


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


      System.out.print("Fibonacci Series: ");
      printFibonacci(n);

      scanner.close();
   }


   static void printFibonacci(int n) {
      int a = 0, b = 1;
      for (int i = 1; i <= n; i++) {
         System.out.print(a + " ");
         int next = a + b;
         a = b;
         b = next;
      }
```

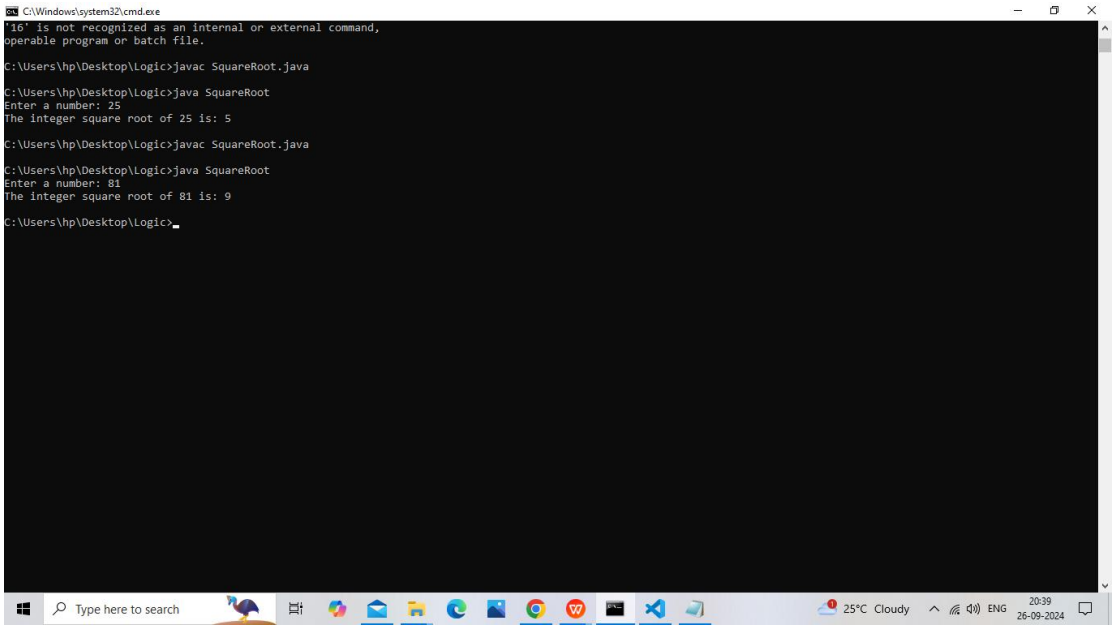
```
    }
}
```



Question - 6

```
import java.util.Scanner;

public class SquareRoot {


   public static int integerSquareRoot(int number) {
      if (number < 0) {
         throw new IllegalArgumentException("Square root of negative numbers is undefined.");
      }

      if (number == 0 || number == 1) {
         return number;
      }

      int low = 1, high = number, result = 0;

      while (low <= high) {
         int mid = (low + high) / 2;
         long square = (long) mid * mid;
         if (square == number) {
            return mid;
         } else if (square < number) {
            low = mid + 1;
            result = mid;
         } else {
            high = mid - 1;
         }
      }

      return result;
```

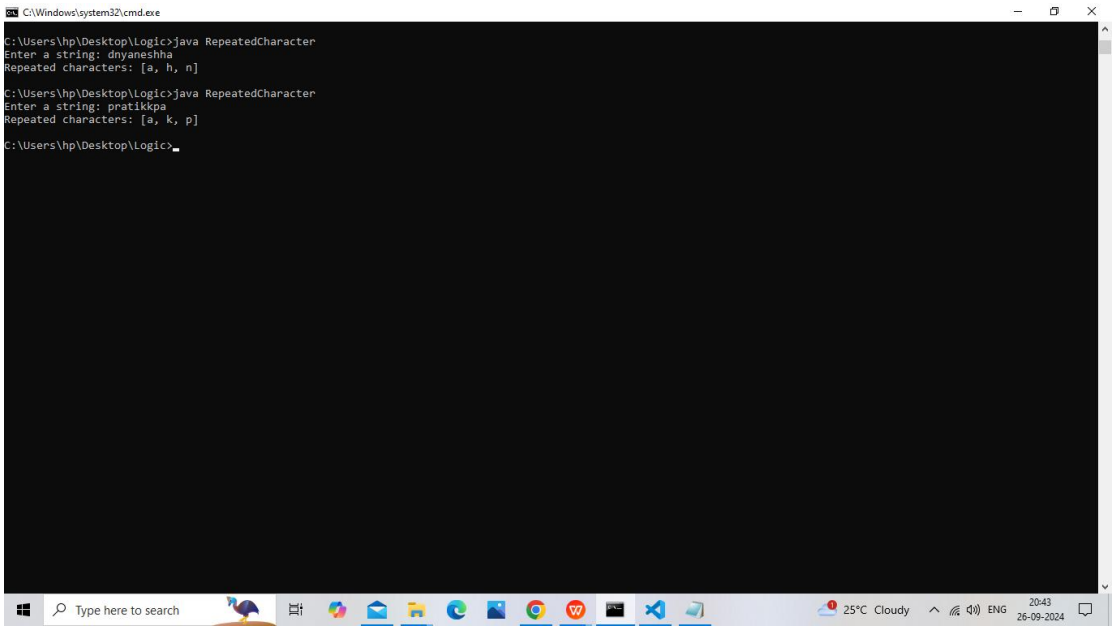
```
    }

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


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


        int sqrt = integerSquareRoot(number);


        System.out.println("The integer square root of " + number + " is: " + sqrt);
    }
}
```



Question - 7

```
import java.util.ArrayList;
import java.util.Scanner;

public class RepeatedCharacter {


    public static ArrayList<Character> findRepeatedCharacters(String input) {
        int[] charCount = new int[256];
        ArrayList<Character> repeatedChars = new ArrayList<>();


        for (char c : input.toCharArray()) {
            charCount[c]++;
        }
```

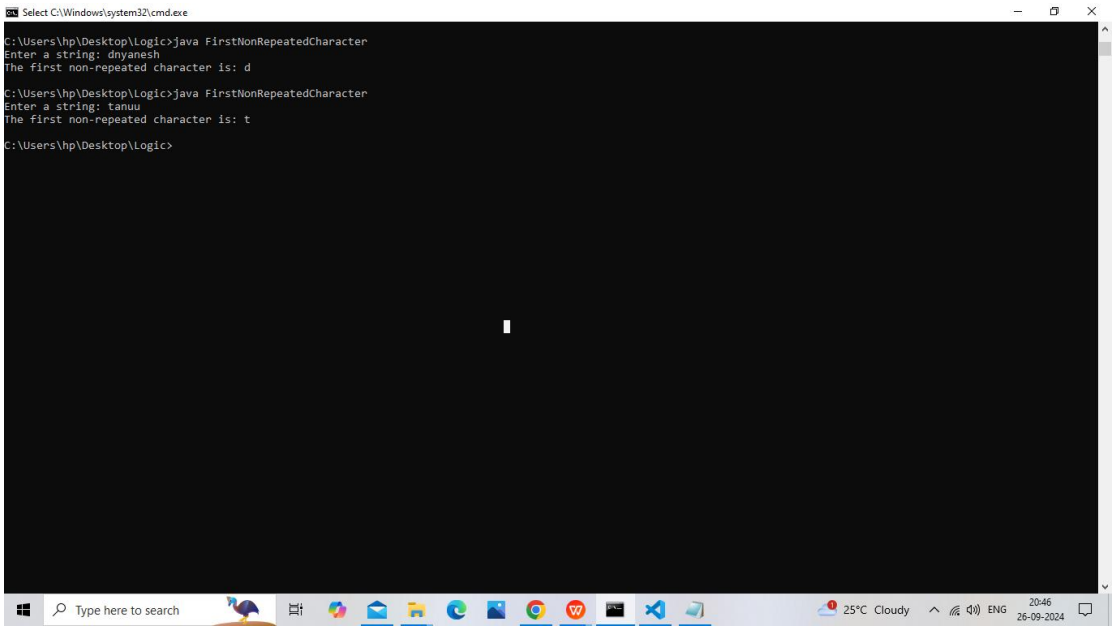
```
        for (int i = 0; i < 256; i++) {
            if (charCount[i] > 1) {
                repeatedChars.add((char) i);
            }
        }

        return repeatedChars;
    }

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


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


        ArrayList<Character> repeatedChars = findRepeatedCharacters(input);


        System.out.println("Repeated characters: " + repeatedChars);
    }
}
```



Question - 8

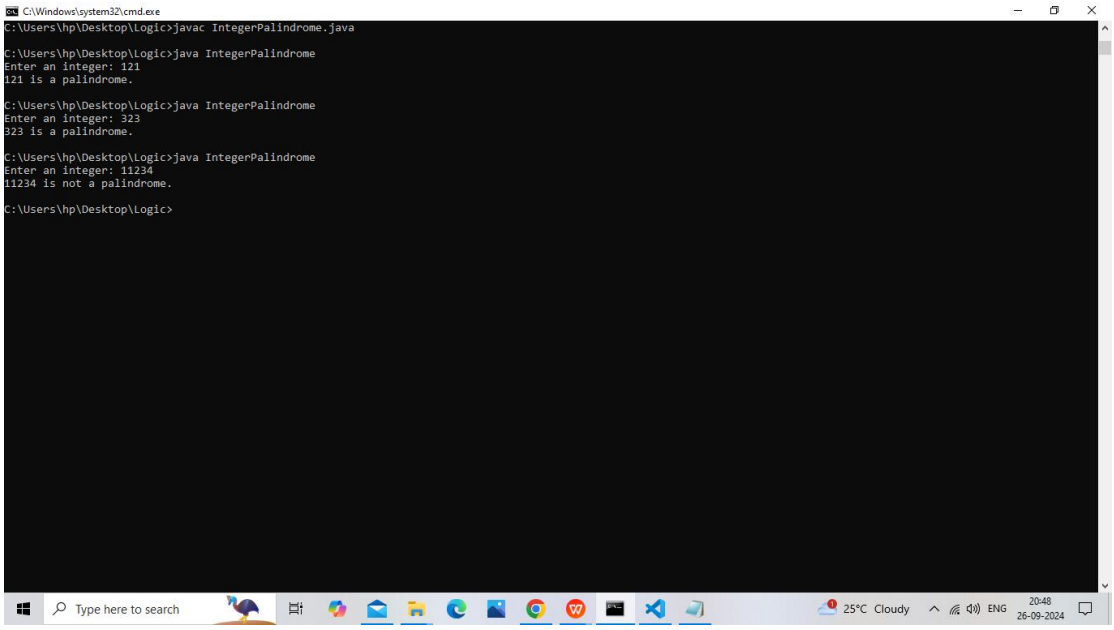
```
import java.util.LinkedHashMap;
import java.util.Map;
import java.util.Scanner;

public class FirstNonRepeatedCharacter {


    public static Character findFirstNonRepeatedCharacter(String input) {
```

```
        Map<Character, Integer> charCountMap = new LinkedHashMap<>();


        for (char c : input.toCharArray()) {
            charCountMap.put(c, charCountMap.getOrDefault(c, 0) + 1);
        }


        for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
            if (entry.getValue() == 1) {
                return entry.getKey();
            }
        }

        return null;
    }

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


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


        Character result = findFirstNonRepeatedCharacter(input);


        if (result != null) {
            System.out.println("The first non-repeated character is: " + result);
        } else {
            System.out.println("No non-repeated character found.");
        }
    }
}
```



Question - 9

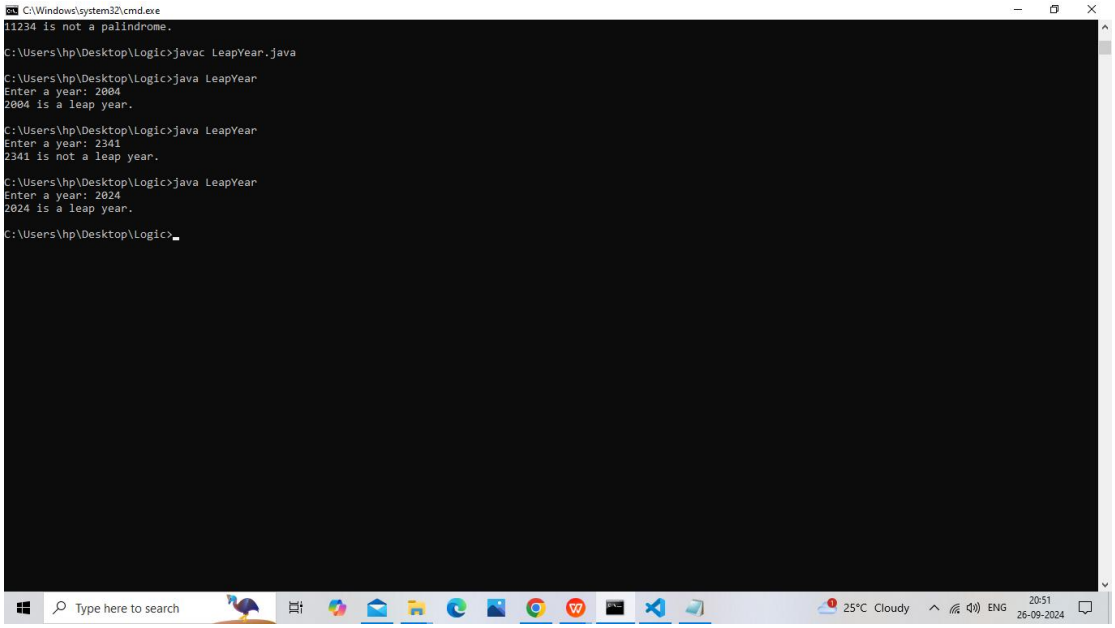
```
import java.util.Scanner;

public class IntegerPalindrome {


   public static boolean isPalindrome(int num) {

      if (num < 0) {
         return false;
      }

      int originalNum = num;
      int reversedNum = 0;


      while (num != 0) {
         int lastDigit = num % 10;
         reversedNum = reversedNum * 10 + lastDigit;
         num /= 10;
      }


      return originalNum == reversedNum;
   }

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


      System.out.print("Enter an integer: ");
      int input = scanner.nextInt();


      boolean result = isPalindrome(input);


      if (result) {
         System.out.println(input + " is a palindrome.");
      } else {
         System.out.println(input + " is not a palindrome.");
      }
   }
}
```



Question - 10

```
import java.util.Scanner;

public class LeapYear {


   public static boolean isLeapYear(int year) {

      if (year % 4 == 0) {
         if (year % 100 == 0) {
            return year % 400 == 0;
         }
         return true;
      }
      return false;
   }

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


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


      boolean result = isLeapYear(year);


      if (result) {
         System.out.println(year + " is a leap year.");
      } else {
         System.out.println(year + " is not a leap year.");
      }
   }
```

}


C:\Windows\system32\cmd.exe
11234 is not a palindrome.
C:\Users\hp\Desktop\Logic>javac LeapYear.java
C:\Users\hp\Desktop\Logic>java LeapYear
Enter a year: 2004
2004 is a leap year.
C:\Users\hp\Desktop\Logic>java LeapYear
Enter a year: 2341
2341 is not a leap year.
C:\Users\hp\Desktop\Logic>java LeapYear
Enter a year: 2024
2024 is a leap year.
C:\Users\hp\Desktop\Logic>
Type here to search
25°C Cloudy
ENG
20:51
26-09-2024