1. Write a program in Java for dynamically changing the color of Text using

Multithreading.

PROGRAM :-

import java.awt.Color;

import javax.swing.JFrame;

import javax.swing.JLabel;

public class dynamictextcolour3 extends Thread {

    private JLabel label;

    private Color[] colors = {Color.RED, Color.GREEN, Color.BLUE};

    private int index = 0;

    public dynamictextcolour3(JLabel label) {

        this.label = label;

    }

    public void run() {

        while (true) {

            label.setForeground(colors[index]);

            index = (index + 1) % colors.length;

            try {

                Thread.sleep(1000);

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

    }

    public static void main(String[] args) {

        JFrame frame = new JFrame("Dynamic Text Color");

        JLabel label = new JLabel("Hello, world!");

        frame.add(label);

        frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        frame.setSize(300, 200);

        frame.setVisible(true);

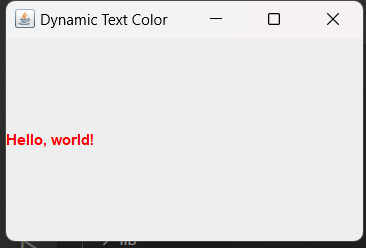
        dynamictextcolour3 dtc = new dynamictextcolour3(label);

        dtc.start();

    }

}

OUTPUT :-



2. Removing duplicate elements in java : Find/Debug the errors and get output

class duplicate

{

// Function to remove duplicate elements

// This function returns new size of modified

// array.

static int removeDuplicates(int arr[], int n)

{

// Return, if array is empty

// or contains a single element

if (n==0 || n==1)

return n;

int[] temp = new int[n];

// Start traversing elements

int j = 0;

for (int j=0; i&lt;n-1; i++)

// If current element is not equal

// to next element then store that

// current element

if (arr[i] != arr[i+1])

temp[j++] = arr[i];

// Store the last element as whether

// it is unique or repeated, it hasn&#39;t

// stored previously

temp[j++] = arr[n-1];

// Modify original array

for (int i=0; i&lt;j; i++)

arr[i] = temp[i];

return j;

}

public static void main (String[] args)

{

it arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};

int n = arr.length;

n = removeDuplicates(arr);

// Print updated array

for (int i=0; i&lt;n; i++)

System.out.print(arr[i]+&quot; &quot;);

}

}

PROGRAM :-

public class removeduplicate3 {

        static int removeDuplicates(int arr[], int n) {

            if (n == 0 || n == 1) {

                return n;

            }

            int[] temp = new int[n];

            int j = 0;

            for (int i = 0; i < n - 1; i++) {

                if (arr[i] != arr[i + 1]) {

                    temp[j++] = arr[i];

                }

            }

            temp[j++] = arr[n - 1];

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

                arr[i] = temp[i];

            }

            return j;

        }

        public static void main(String[] args) {

            int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};

            int n = arr.length;

            n = removeDuplicates(arr, n);

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

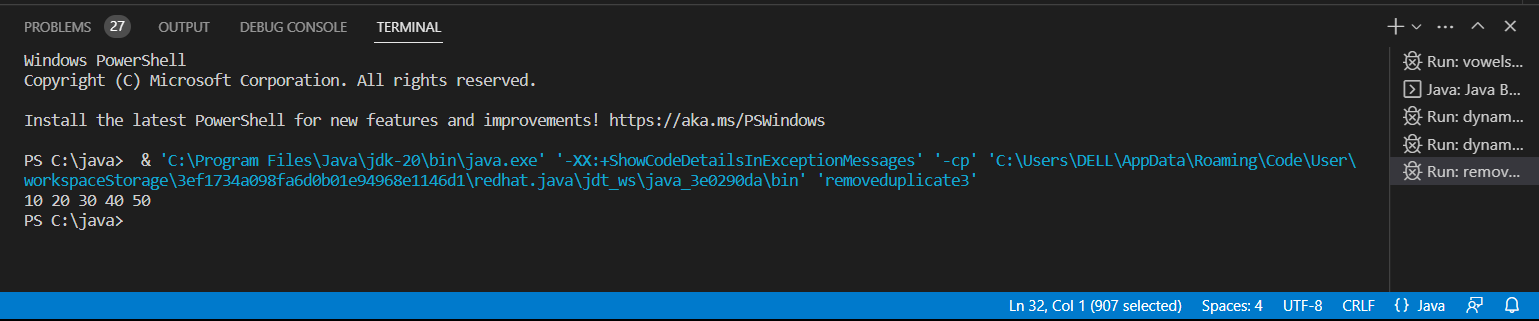
                System.out.print(arr[i] + " ");

            }

        }

    }

OUTPUT :-



3. Write a program to reverse a word using loop? (Not to use inbuilt functions)

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET

PROGRAM :-

import java.util.\*;

public class reverseword3 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a word to reverse: ");

        String word = scanner.nextLine();

        String reversedWord = "";

        for (int i = word.length() - 1; i >= 0; i--) {

            reversedWord += word.charAt(i);

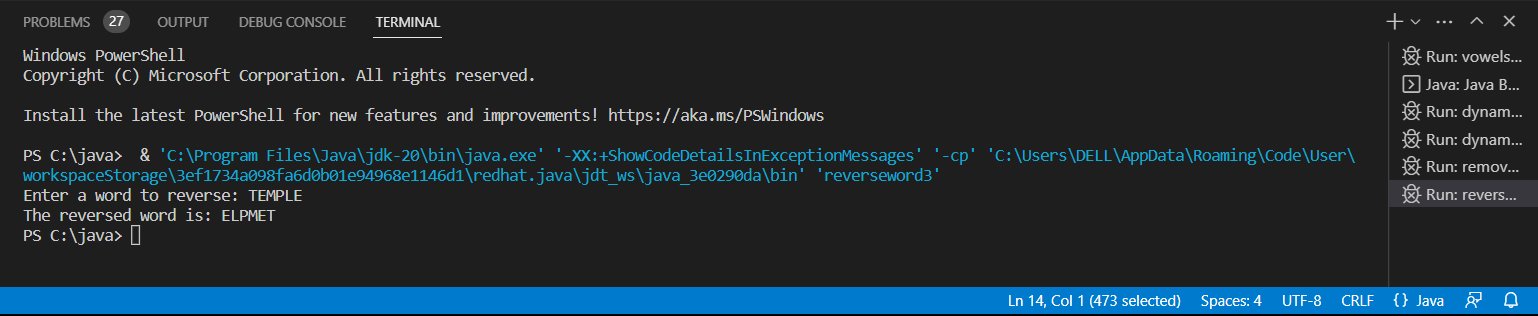
        }

        System.out.println("The reversed word is: " + reversedWord);

    }

}

OUTPUT :-



4. Write a program to print the number of vowels in the given statement?

Sample Input:

Saveetha School of Engineering

Sample Output:

Number o vowels = 12

PROGRAM :-

import java.util.\*;

public class vowelsvount3 {

    public static void main(String[] args) {

        Scanner v = new Scanner(System.in);

        System.out.println("Enter the String : ");

        String ch = v.nextLine();

        int count = 0;

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

        {

            char x = ch.charAt(i);

            if (x == 'a' || x == 'e' || x == 'i' || x == 'o' || x == 'u' || x == 'A' || x == 'E' || x == 'I' || x == 'O' || x == 'U')

            {

                count++;

            }

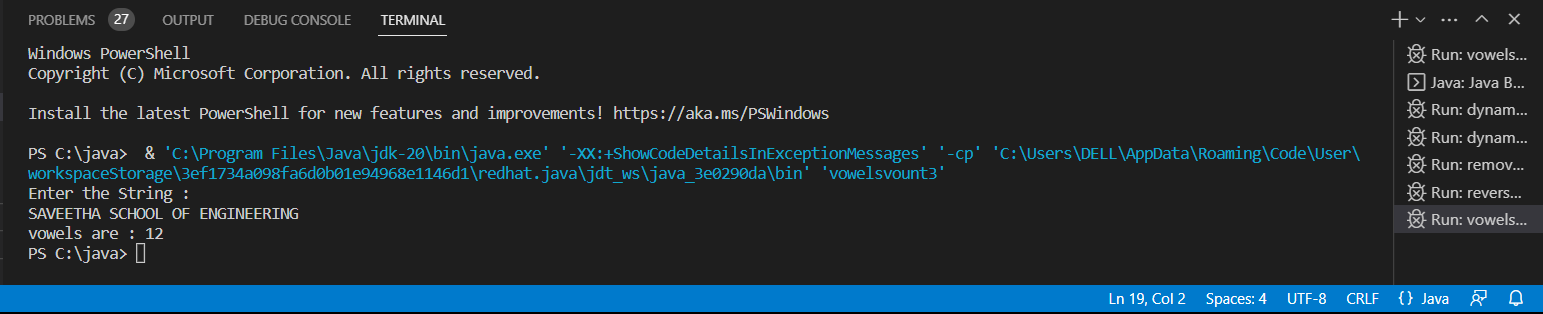
        }

        System.out.println("vowels are : " + count);

    }

}

OUTPUT :-



5. Write a program to print consonants and vowels separately in the given word

Sample Input:

Given Word: Engineering

Sample Output:

Consonants: n g n r n g

Vowels: e i e ei

PROGRAM :-

import java.util.\*;

public class vowelsconsonents3 {

    public static void main(String[] args) {

        Scanner v = new Scanner(System.in);

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

        String x = v.nextLine();

        String vowels = "";

        String consonants = "";

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

            char ch = x.charAt(i);

            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

                vowels += ch;

            } else if (Character.isLetter(ch)) {

                consonants += ch;

            }

        }

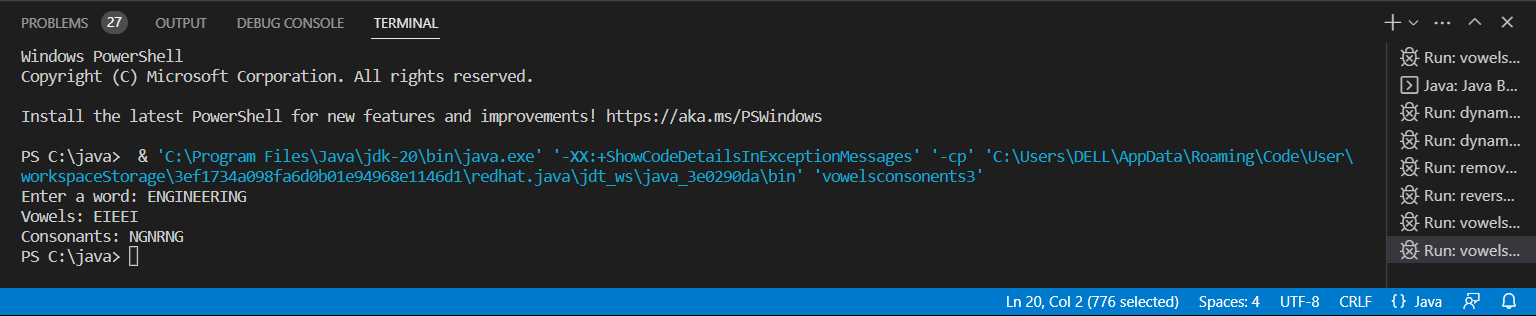
        System.out.println("Vowels: " + vowels);

        System.out.println("Consonants: " + consonants);

    }

}

OUTPUT :-



6. Write a program that finds whether a given character is present in a string or not. In case

it is present it prints the index at which it is present. Do not use built-in find functions to

search the character.

Sample Input:

Enter the string: I am a programmer

Enter the character to be searched: p

Sample Output:

P is found in string at index: 8

Note: Check for non available Character in the given statement as Hidden Test case.

PROGRAM :-

import java.util.\*;

public class searchcharacter3 {

    public static void main(String[] args) {

        Scanner v = new Scanner (System.in);

        System.out.println("Enter the String : ");

        int n,i;

        String s1 = v.nextLine();

        n = s1.length();

        System.out.println("Enter the character : ");

        char s2 = v.next().charAt(0);

        System.out.println(s2 + "is found in string : ");

        for (i=0 ; i<n ; i++)

        {

            if (s1.charAt(i) == s2)

            {

                System.out.println(i+ " ");

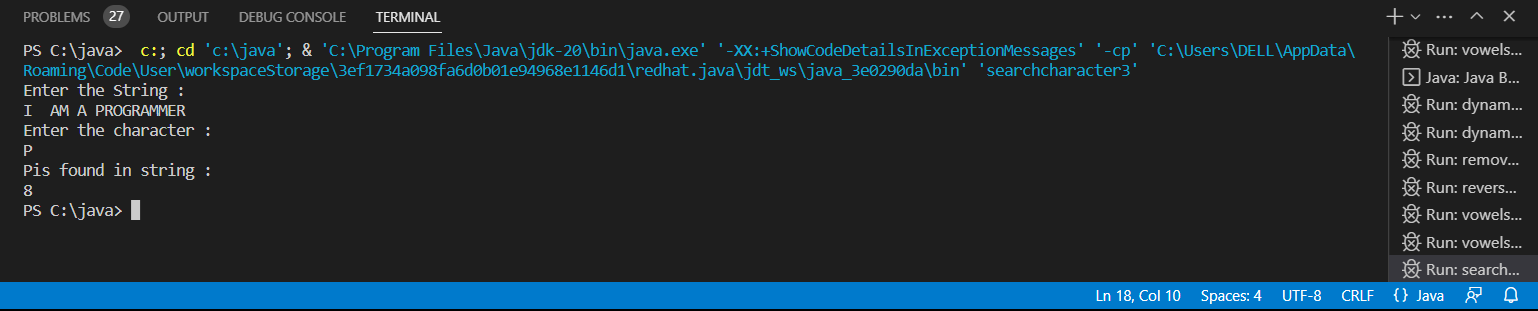
            }

        }

    }

}

OUTPUT :-



7. Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:

Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION

2. MATRICULATION

3. MANIPULATION

PROGRAM :-

import java.util.\*;

public class reversealphabeticalorder3 {

    public static void main(String[] args) {

        Scanner v = new Scanner(System.in);

        System.out.print("Enter the word: ");

        String word = v.nextLine().toUpperCase();

        Character[] letters = new Character[word.length()];

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

            letters[i] = word.charAt(i);

        }

        Arrays.sort(letters, Collections.reverseOrder());

        System.out.print("Alphabetical Order: ");

        for (char letter : letters) {

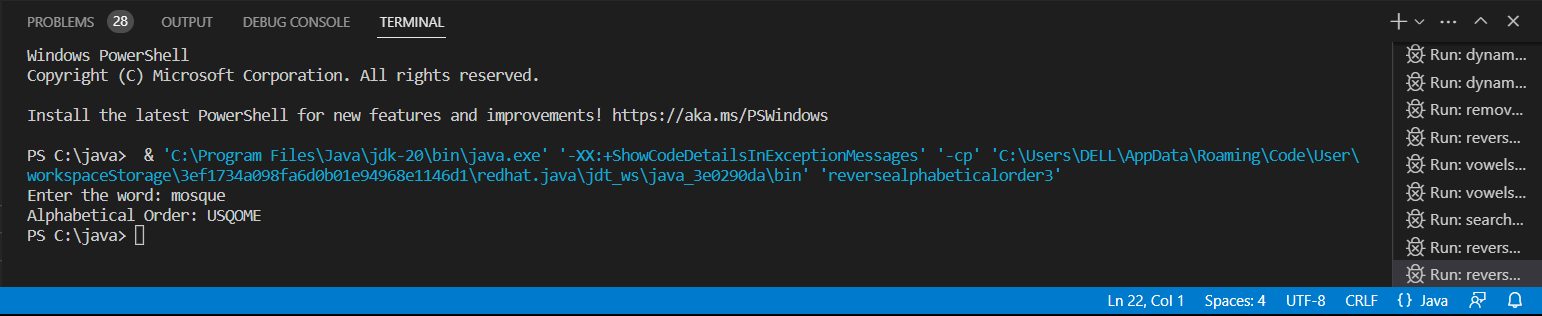
            System.out.print(letter);

        }

    }

}

OUTPUT :-



8. Write a program that accepts a string from user and displays the same string after

removing vowels from it.

Sample Input &amp; Output:

Enter a string: we can play the game

The string without vowels is: w cn ply thgm

PROGRAM :-

import java.util.\*;

public class removingvowels3 {

    public static void main(String[]args)

    {

        Scanner v = new Scanner(System.in);

        System.out.println("enter the string : ");

        String s1,s2;

        s1 = v.nextLine();

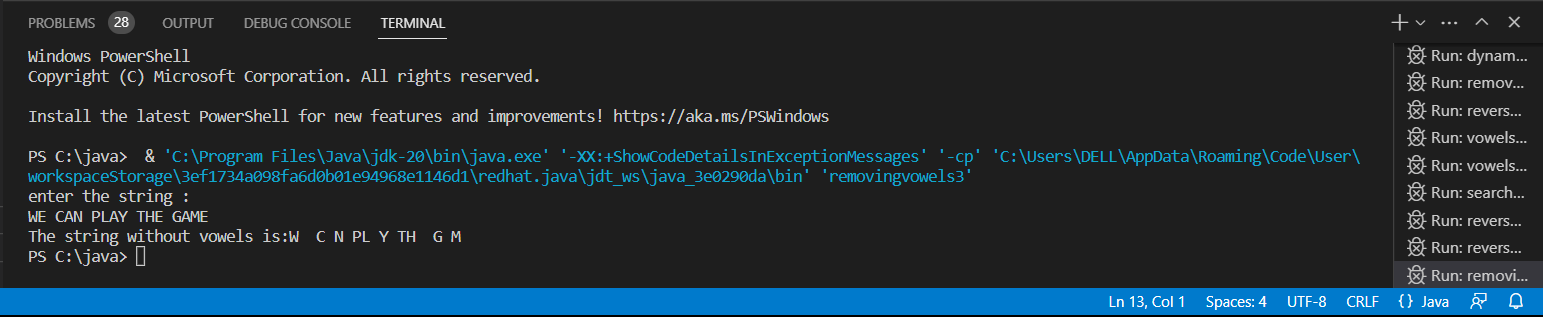
        s2 = s1.replaceAll("[aeiouAEIOU]", " ");

        System.out.println("The string without vowels is:"+s2);

    }

}

OUTPUT :-



9. Write a program to print the special characters separately and print number of Special

characters in the line?

PROGRAM :-

import java.util.\*;

public class specialcharacters3

{

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        System.out.print("Enter a line of text: ");

        String line = s.nextLine();

        int count = 0;

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

            char ch = line.charAt(i);

            if (!Character.isLetterOrDigit(ch) && !Character.isWhitespace(ch)) {

                System.out.println(ch);

                count++;

            }

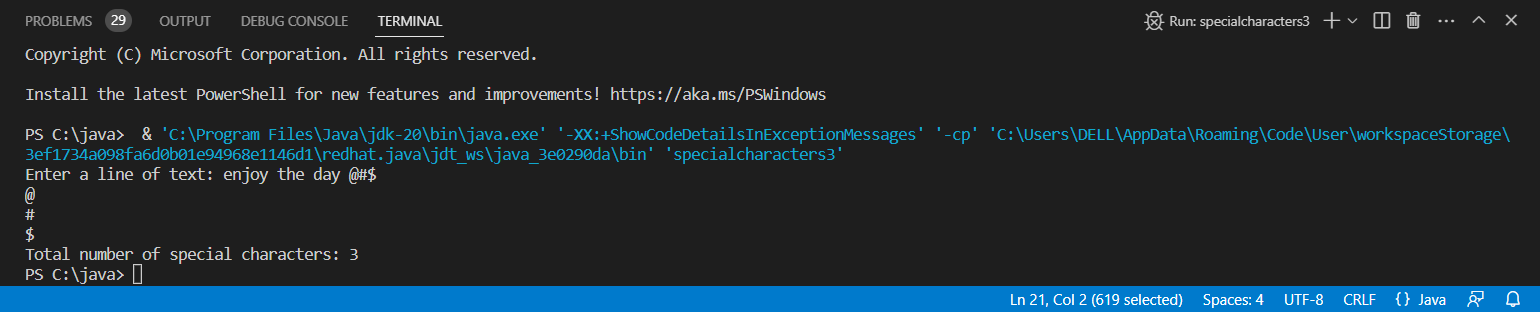
        }

        System.out.println("Total number of special characters: " + count);

    }

}

OUTPUT :-



10. Write a program that would sort a list of names in alphabetical order Ascending or

Descending, choice get from the user?

Sample Input:

Banana

Carrot

Radish

Apple

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

PROGRAM :-

import java.util.\*;

public class ascendescordersorting {

    public static void main(String[] args) {

        Scanner v = new Scanner(System.in);

        System.out.print("Enter the number of names you want to sort: ");

        int n = v.nextInt();

        String[] names = new String[n];

        System.out.println("Enter the names: ");

        for (int i = 0; i < n; i++) {

            names[i] = v.next();

        }

        System.out.print(" Order (A/D) : ");

        char choice = v.next().charAt(0);

        if (choice == 'A') {

            Arrays.sort(names);

            System.out.println("Sorted names in ascending order:");

            for (String name : names) {

                System.out.println(name);

            }

        } else if (choice == 'D') {

            Arrays.sort(names, Collections.reverseOrder());

            System.out.println("Sorted names in descending order:");

            for (String name : names) {

                System.out.println(name);

            }

        } else {

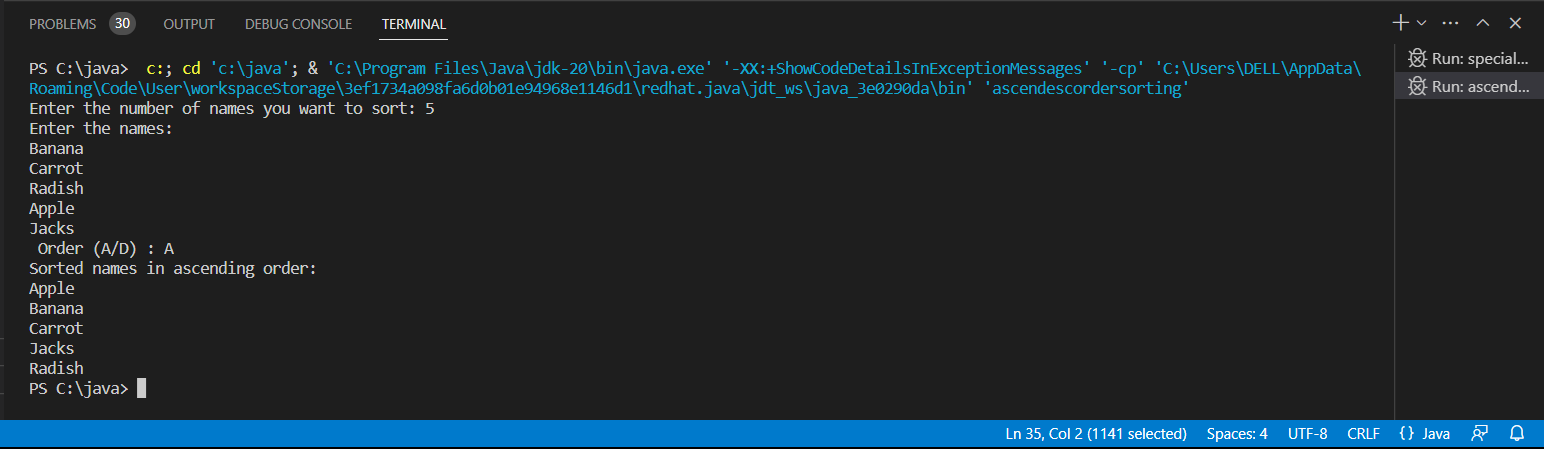
            System.out.println("Invalid choice. Please enter 'A' or 'D'.");

        }

    }

}

OUTPUT :-



11. Write a program to convent the given string to integer?

Sample Input:

String: 1234

Sample Output:

Out put String: 1234

PROGRAM :-

import java.util.\*;

public class sameoutput3 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

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

        String str = input.next();

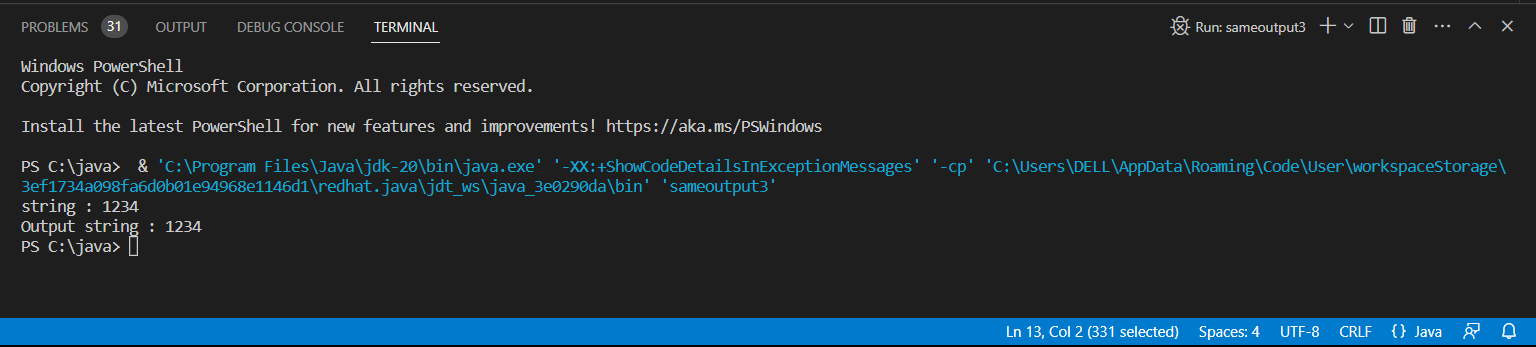
        int num = Integer.parseInt(str);

        System.out.println("Output string : " + num);

    }

}

OUTPUT :-



12. Write a program to check the entered user name is valid or not. Get both the inputs from the user.

PROGRAM :-

import java.util.\*;

public class validorinvlaid3 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

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

        String username = input.nextLine();

        boolean isValid = true;

        if (username.length() < 6 || username.length() > 20) {

            isValid = false;

        }

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

            char ch = username.charAt(i);

            if (!Character.isLetterOrDigit(ch) && ch != '@') {

                isValid = false;

                break;

            }

        }

        if (isValid) {

            System.out.println("The username is valid.");

        } else {

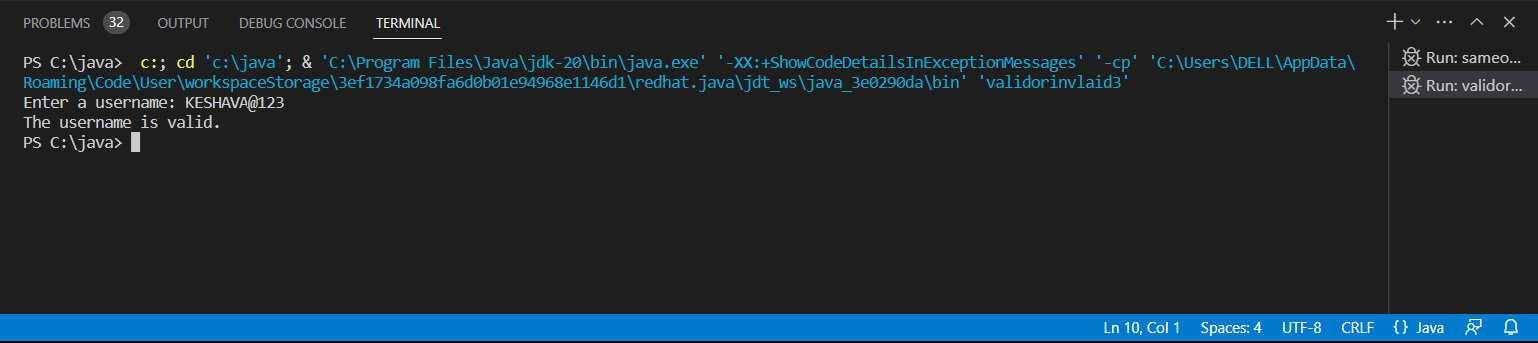
            System.out.println("The username is invalid.");

        }

    }

}

OUTPUT :-



13. The Fibonacci numbers, commonly denoted F(n) form a sequence, called the

Fibonacci sequence, such that each number is the sum of the two preceding ones,

starting from 0 and 1. That is,

F(0) = 0, F(1) = 1

F(n) = F(n - 1) + F(n - 2), for n &gt; 1.

Given n, calculate F(n).

Example 1:

Input: n = 2

Output: 1

Explanation: F(2) = F(1) + F(0) = 1 + 0 = 1.

Example 2:

Input: n = 3

Output: 2

Explanation: F(3) = F(2) + F(1) = 1 + 1 = 2.

Example 3:

Input: n = 4

Output: 3

Explanation: F(4) = F(3) + F(2) = 2 + 1 = 3.

Constraints:

0 &lt;= n &lt;= 30

class Solution {

public:

int fib(int n) {

}

}

PROGRAM :-

import java.util.\*;

public class fibanocissi3 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

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

        int n = input.nextInt();

        int fib1 = 0, fib2 = 1, fibN = 0;

        if (n == 0) {

            fibN = 0;

        } else if (n == 1) {

            fibN = 1;

        } else {

            for (int i = 2; i <= n; i++) {

                fibN = fib1 + fib2;

                fib1 = fib2;

                fib2 = fibN;

            }

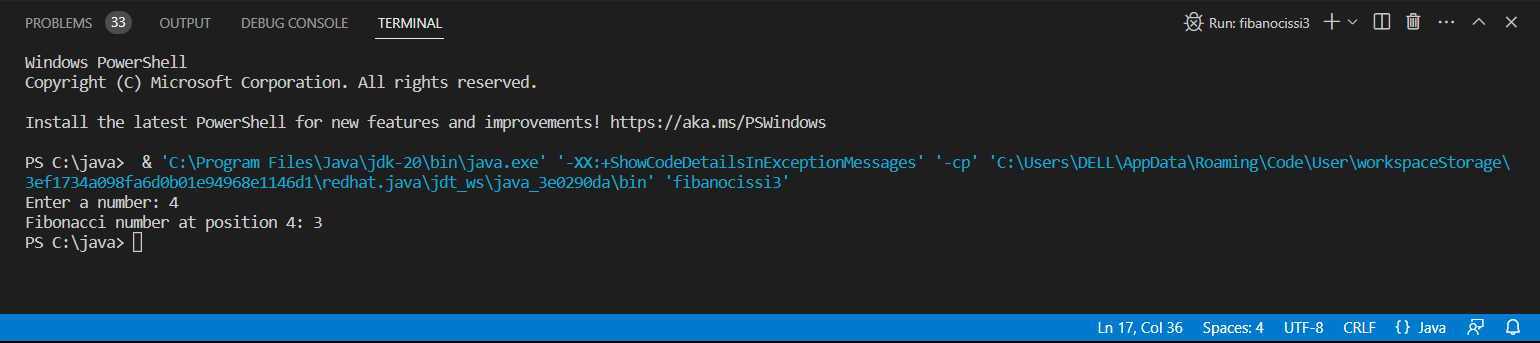
        }

        System.out.println("Fibonacci number at position " + n + ": " + fibN);

    }

}

OUTPUT :-



14. Differentiate Multiprocessing and Multithreading. Display Multiplication table for 5

and 10 using various stages of life cycle of the thread by generating a suitable code in

Java.

PROGRAM :-

    public class multiplicationtable3 implements Runnable {

        private int number;

        public multiplicationtable3(int number) {

            this.number = number;

        }

        @Override

        public void run() {

            System.out.println("Thread " + Thread.currentThread().getId() + " is running");

            for (int i = 1; i <= 10; i++) {

                System.out.println(number + " x " + i + " = " + (number \* i));

                try {

                    Thread.sleep(100);

                } catch (InterruptedException e) {

                    e.printStackTrace();

                }

            }

            System.out.println("Thread " + Thread.currentThread().getId() + " has finished executing");

        }

        public static void main(String[] args) {

            multiplicationtable3 table5 = new multiplicationtable3(5);

            multiplicationtable3 table10 = new multiplicationtable3(10);

            Thread thread1 = new Thread(table5);

            Thread thread2 = new Thread(table10);

            thread1.start();

            thread2.start();

            try {

                thread1.join();

                thread2.join();

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

            System.out.println("Main thread has finished executing");

        }

    }

OUTPUT :-



15. An ugly number is a positive integer whose prime factors are limited to 2, 3, and 5.

Given an integer n, return true if n is an ugly number.

Example 1:

Input: n = 6

Output: true

Explanation: 6 = 2 × 3

Example 2:

Input: n = 1

Output: true

Explanation: 1 has no prime factors, therefore all of its prime factors are

limited to 2, 3, and 5.

Example 3:

Input: n = 14

Output: false

Explanation: 14 is not ugly since it includes the prime factor 7.

Constraints:

-231 &lt;= n &lt;= 231 - 1

class Solution {

public:

bool isUgly(int n) {

}

}

PROGRAM:-

public class ugly3 {

        public static boolean isUgly(int n) {

            if (n <= 0) {

                return false;

            }

            while (n % 2 == 0) {

                n /= 2;

            }

            while (n % 3 == 0) {

                n /= 3;

            }

            while (n % 5 == 0) {

                n /= 5;

            }

            return n == 1;

        }

        public static void main(String[] args) {

            int num1 = 6;

            int num2 = 14;

            System.out.println(num1 + " is an ugly number: " + isUgly(num1));

            System.out.println(num2 + " is an ugly number: " + isUgly(num2));

        }

    }

OUTPUT :-

