CSA0980 -- PROGRAMMING IN JAVA FOR IDL TECHNOLOGY :-

M . NAGA KESHAVA

REG NO :-192125019

1. Write a program to given an integer n, return true if it is a power of three. Otherwise, return false.

Input =27
Output= true
Explanation: 27=3 3

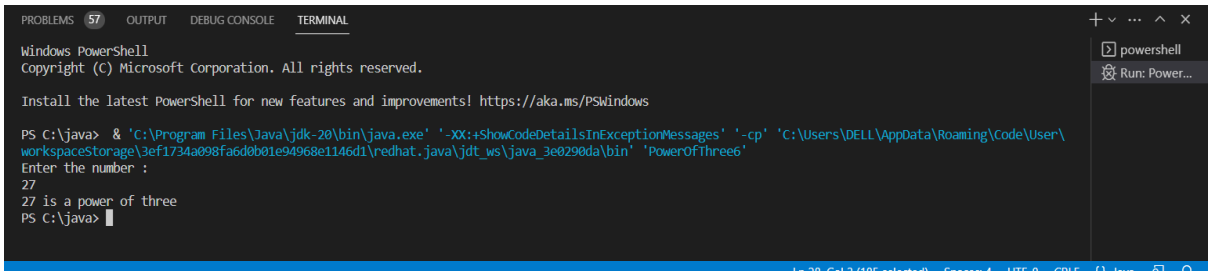
Test cases:
1. 12
2. abc@45
3. 1827
4. -100
5. 0

PROGRAM :-

```
import java.util.*;
public class PowerOfThree6
{
     public static boolean isPowerOfThree6(int n)
        {
            if (n <= 0)
            {
                return false;
            }
            while (n % 3 == 0)
            {
                n /= 3;
            }
            return n == 1;
        }
        public static void main(String[] args)
        {
            Scanner v = new Scanner (System.in);
            System.out.println("Enter the number : ");
            int n = v.nextInt();
            if (isPowerOfThree6(n))
            {
                System.out.println(n + " is a power of three");
            } else {
                System.out.println(n + " is not a power of three");
```

```
            }
        }
}
```

OUTPUT :-



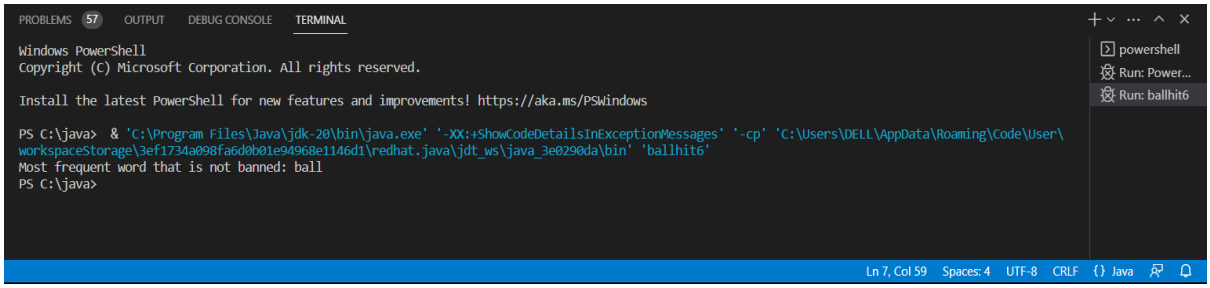
2. Write a program to given a string paragraph and a string array of the banned words banned, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique.
Input Paragraph=”Ram hit a ball, the hit ball flew far after it was hit”,
Banned = [hit]
Output=”Ball”

PROGRAM :-

```
import java.util.*;
public class ballhit6 {
    public static String mostCommonWord(String paragraph, String[] banned)
    {
        String cleanParagraph = paragraph.replaceAll("[^a-zA-Z0-9 ]",
"").toLowerCase();
        Map<String, Integer> wordCounts = new HashMap<>();
        String[] words = cleanParagraph.split("\\s+");
        for (String word : words) {
            if (!Arrays.asList(banned).contains(word)) {
                wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);
            }
        }
        String mostCommonWord = null;
        int maxCount = 0;
        for (String word : wordCounts.keySet()) {
            int count = wordCounts.get(word);
            if (count > maxCount) {
                mostCommonWord = word;
                maxCount = count;
            }
        }
        return mostCommonWord;
    }
    public static void main(String[] args) {
```

```
        String paragraph = "Ram hit a ball, the hit ball flew far after it was
hit";
        String[] banned = {"hit"};
        String mostCommonWord = mostCommonWord(paragraph, banned);
        System.out.println("Most frequent word that is not banned: " +
mostCommonWord);
    }
}
```

OUTPUT :-



3. Write a program to given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right.

Input: arr = [1, 0, 2, 3, 0, 4, 5, 0]

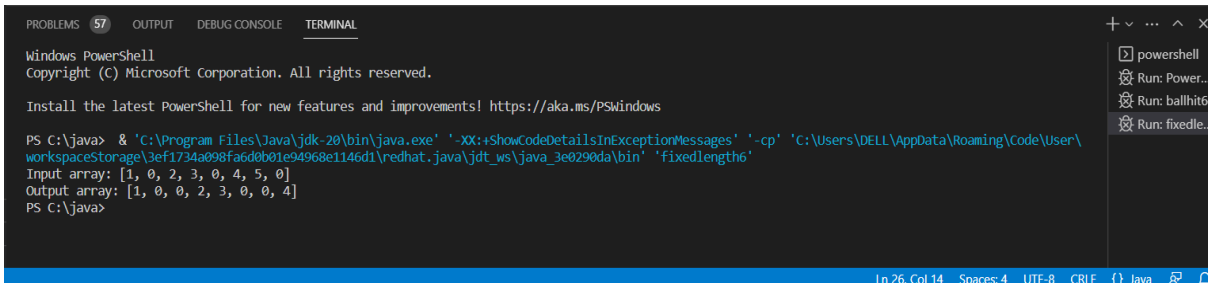
Output: [1, 0, 0, 2, 3, 0, 0, 4]

PROGRAM :-

```
import java.util.*;
public class fixedlength6
{
    public static void duplicateZeros(int[] arr)
    {
        int zeros = 0;
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            if (arr[i] == 0)
            {
                zeros++;
            }
        }
        for (int i = n - 1, j = n + zeros - 1; i >= 0; i--, j--)
        {
            if (arr[i] == 0)
            {
                if (j < n) {
                    arr[j] = 0;
                }
                j--;
            }
            if (j < n)
```

```
            {
                arr[j] = arr[i];
            }
        }
    }
    public static void main(String[] args)
    {
        int[] arr = {1, 0, 2, 3, 0, 4, 5, 0};
        System.out.println("Input array: " + Arrays.toString(arr));
        duplicateZeros(arr);
        System.out.println("Output array: " + Arrays.toString(arr));
    }
}
```

OUTPUT :-



Explanation: After calling your function, the input array is modified to [1, 0, 0, 2, 3, 0, 0, 4]

4. Write a program to given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

Input nums = [3, 0, 1]

Output: 2

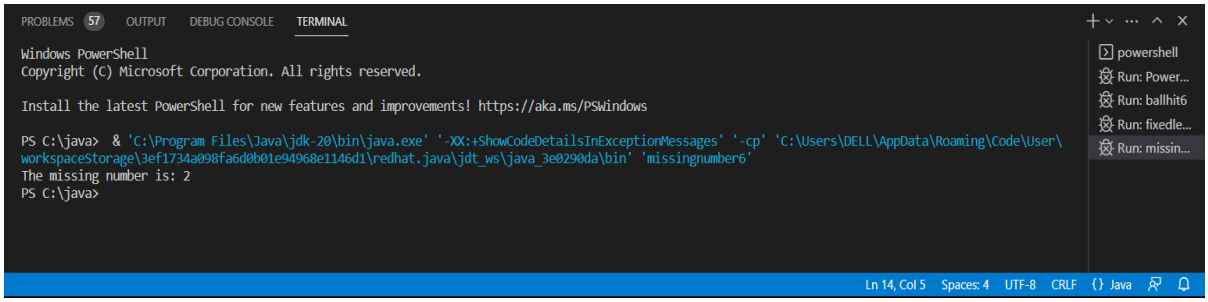
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0, 3]. 2 is the missing number in the range since it does not appear in nums.

PROGRAM :-

```
public class missingnumber6
{
        public int missingNumber(int[] nums)
        {
            int n = nums.length;
            int sum = (n * (n + 1)) / 2;
            int actualSum = 0;
            for (int i = 0; i < n; i++)
            {
                actualSum += nums[i];
            }
            return sum - actualSum;
        }

        public static void main(String[] args) {
```

```
            int[] nums = {3, 0, 1};
            missingnumber6 mn = new missingnumber6();
            int missingNum = mn.missingNumber(nums);
            System.out.println("The missing number is: " + missingNum);
        }
    }
```

OUTPUT :-

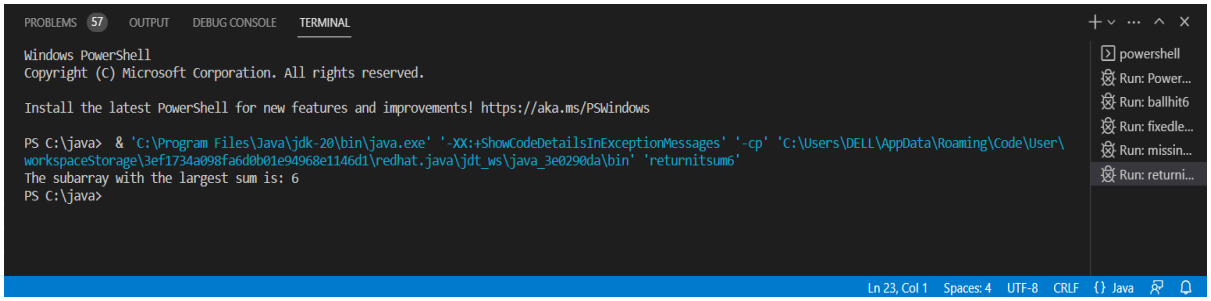


5. Write a program to given an integer array nums, find the subarray with the largest sum, and return its sum.
Input nums = [-2, 1,-3, 4,-1, 2, 1,-5, 4]
Output: 6
Explanation: The subarray [4,-1, 2, 1] has the largest sum 6.

PROGRAM :-

```
public class returnitsum6
{
        public static int maxSubArray(int[] nums)
        {
            int maxSum = nums[0];
            int currentSum = nums[0];
            for (int i = 1; i < nums.length; i++)
            {

                currentSum = Math.max(nums[i], currentSum + nums[i]);
                maxSum = Math.max(maxSum, currentSum);
            }
            return maxSum;
        }

        public static void main(String[] args)
        {
            int[] nums = {-2, 1, -3, 4, -1, 2, 1, -5, 4};
            int largestSum = maxSubArray(nums);
            System.out.println("The subarray with the largest sum is: " +
largestSum);
        }
}
```

OUTPUT :-



6. Write a program to print the multiplication table of number m up to n.

Sample Input:

M = 4

N = 5

Sample Output:

1x4=4

2x4=8

3x4=12

4x4=16

5x4=20

Test cases:

M = 6, N = -3

M = -3, N = 5

M = 4, N = 0

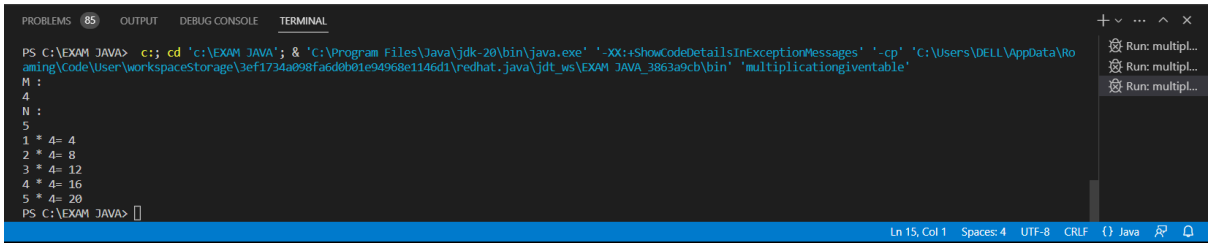
M = 0, N = 0

M = -5, N = -5

PROGRAM :-

```
import java.util.*;
public class multiplicationgiventable {
    public static void main(String[] args) {
        Scanner v = new Scanner(System.in);
        System.out.println("M : ");
        int a = v.nextInt();
        System.out.println("N : ");
        int b = v.nextInt();
        for (int i=1;i<=b;i++)
        {
            System.out.println(i + " * "+ a + "= " + i*a);
        }
    }
}
```

OUTPUT :-



7. Write Java programs to implement multiple threads and apply join method for thread and thread has to be started after 500ms using sleep ().

PROGRAM :-

```
public class sleepthread6
{
        public static void main(String[] args)
        {
            Thread t1 = new Thread(new Runnable()
            {
                public void run() {
                    try {
                        System.out.println("Thread 1 started");
                        Thread.sleep(2000);
                        System.out.println("Thread 1 ended");
                    } catch (InterruptedException e)
                    {
                        e.printStackTrace();
                    }
                }
            });
            Thread t2 = new Thread(new Runnable()
            {
                public void run() {
                    try {
                        System.out.println("Thread 2 started");
                        Thread.sleep(3000);
                        System.out.println("Thread 2 ended");
                    } catch (InterruptedException e)
                    {
                        e.printStackTrace();
                    }
                }
            });
            try {
                Thread.sleep(500);
                t1.start();
                t2.start();
            } catch (InterruptedException e)
            {
                e.printStackTrace();
            }
```

```
            try {
                t1.join();
                t2.join();
            } catch (InterruptedException e)
            {
                e.printStackTrace();
            }
            System.out.println("Threads have completed");
        }
    }
```

OUTPUT :-

8. Generate a Java code that implements java selection and iteration statements. Use do while loop to process a menu selection. When a menu is selected, it should display the syntax of the selected statements.

PROGRAM :-

```
import java.util.*;
public class menuselectedjava6 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("Menu:");
            System.out.println("1. if statement");
            System.out.println("2. switch statement");
            System.out.println("3. for loop");
            System.out.println("4. while loop");
            System.out.println("5. do-while loop");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Syntax of if statement:");
                    System.out.println("if (condition) {");
```

```
                    System.out.println("    // code to be executed if
condition is true");
                    System.out.println("}");
                    break;
                case 2:
                    System.out.println("Syntax of switch statement:");
                    System.out.println("switch (expression) {");
                    System.out.println("    case value1:");
                    System.out.println("        // code to be executed if
expression matches value1");
                    System.out.println("        break;");
                    System.out.println("    case value2:");
                    System.out.println("        // code to be executed if
expression matches value2");
                    System.out.println("        break;");
                    System.out.println("    // ...");
                    System.out.println("    default:");
                    System.out.println("        // code to be executed if none
of the cases match");
                    System.out.println("}");
                    break;
                case 3:
                    System.out.println("Syntax of for loop:");
                    System.out.println("for (initialization; condition;
update) {");
                    System.out.println("    // code to be executed");
                    System.out.println("}");
                    break;
                case 4:
                    System.out.println("Syntax of while loop:");
                    System.out.println("while (condition) {");
                    System.out.println("    // code to be executed");
                    System.out.println("}");
                    break;
                case 5:
                    System.out.println("Syntax of do-while loop:");
                    System.out.println("do {");
                    System.out.println("    // code to be executed");
                    System.out.println("} while (condition);");
                    break;
                case 0:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice, try again.");
                    break;
            }
        } while (choice != 0);
```

```
        scanner.close();
    }
}
```

OUTPUT :-

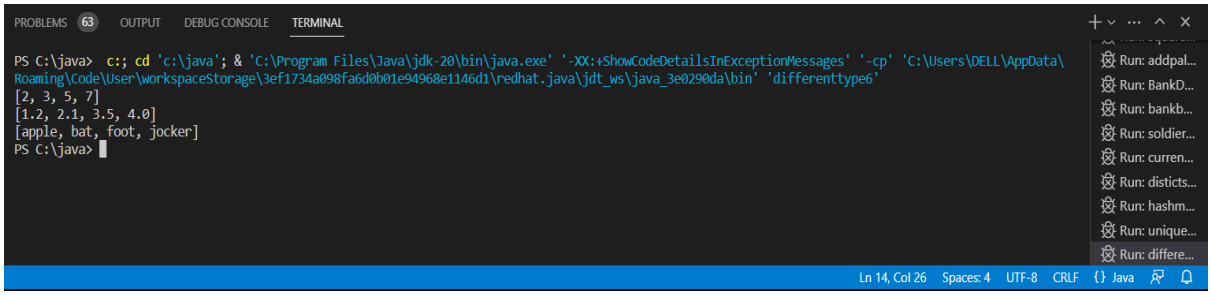
9. Create a simple generics class with type parameters for sorting values of different types.

PROGRAM :-

```
import java.util.Arrays;

class Sorter<T extends Comparable<T>> {
    private T[] arr;

    public Sorter(T[] arr) {
        this.arr = arr;
    }

    public void sort() {
        Arrays.sort(arr);
    }

    public void print() {
        System.out.println(Arrays.toString(arr));
    }
}

public class differenttype6 {
        public static void main(String[] args) {
            Integer[] intArr = { 5, 2, 7, 3 };
            Double[] doubleArr = { 1.2, 3.5, 2.1, 4.0 };
            String[] strArr = { "foot", "bat", "apple", "jocker" };

            Sorter<Integer> intSorter = new Sorter<>(intArr);
            intSorter.sort();
            intSorter.print();
```

```
            Sorter<Double> doubleSorter = new Sorter<>(doubleArr);
            doubleSorter.sort();
            doubleSorter.print();

            Sorter<String> strSorter = new Sorter<>(strArr);
            strSorter.sort();
            strSorter.print();
        }
    }
```

OUTPUT :-

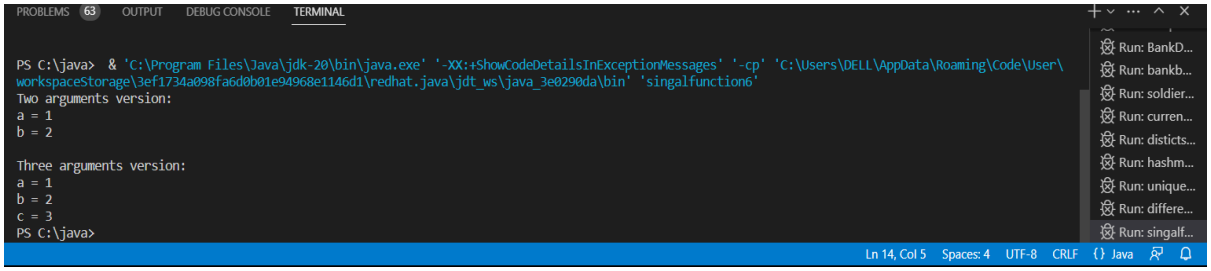


10. Create a class name ‘overload’. Write a program to assign the values for two values by different number of arguments using a single function.

PROGRAM :-

```
public class singalfunction6 {
        public void assignValues(int a, int b) {
            System.out.println("Two arguments version:");
            System.out.println("a = " + a);
            System.out.println("b = " + b);
        }

        public void assignValues(int a, int b, int c) {
            System.out.println("Three arguments version:");
            System.out.println("a = " + a);
            System.out.println("b = " + b);
            System.out.println("c = " + c);
        }

        public static void main(String[] args) {
            singalfunction6 obj = new singalfunction6();
            obj.assignValues(1, 2);
            System.out.println();
            obj.assignValues(1, 2, 3);
        }
}
```

OUTPUT :-



11. Write a Java Program to count the number of words in a string using Hash Map.

PROGRAM :-

```
import java.util.HashMap;
import java.util.Map;
public class wordastring6 {

    public static void main(String[] args) {
        String str = "This is a sample string with sample words";
        String[] words = str.split(" ");

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

        for (String word : words) {
            if (wordCount.containsKey(word)) {
                int count = wordCount.get(word);
                wordCount.put(word, count + 1);
            } else {
                wordCount.put(word, 1);
            }
        }

        System.out.println("Word count:");
        for (String word : wordCount.keySet()) {
            System.out.println(word + ": " + wordCount.get(word));
        }
    }
}
```

OUTPUT :-

12. Write a Java Program to read an email and password from excel sheet by retrieving the cell using getRow() and getCell() method.

PROGRAM :-

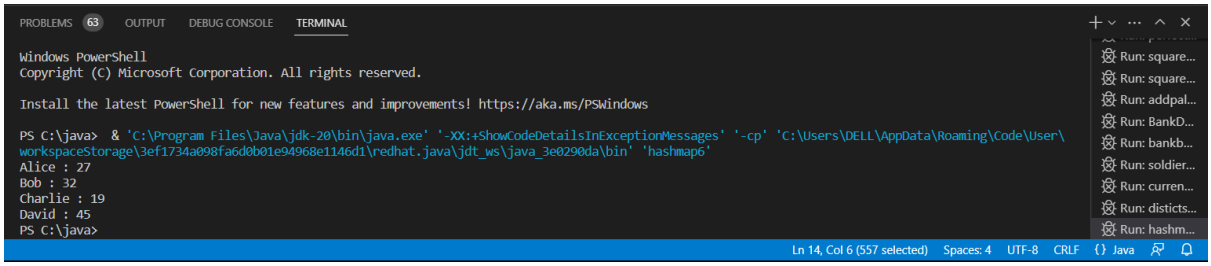
```java
import java.io.FileInputStream;
import java.io.IOException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
public class excelsheet6 {

    public static void main(String[] args) {
        try {
            FileInputStream file = new
FileInputStream("path/to/excel/file.xlsx");
            Workbook workbook = new XSSFWorkbook(file);
            Sheet sheet = workbook.getSheetAt(0); // Accessing the first
worksheet
            Row row = sheet.getRow(0); // Accessing the first row
            Cell emailCell = row.getCell(0); // Accessing the first cell
            Cell passwordCell = row.getCell(1); // Accessing the second cell
            String email = emailCell.getStringCellValue();
            String password = passwordCell.getStringCellValue();
            System.out.println("Email: " + email);
            System.out.println("Password: " + password);
            workbook.close();
            file.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

13. Write a Java program to sorts the given value using Hash Map.

PROGRAM :-

```java
import java.util.*;
public class hashmap6 {
          public static void main(String[] args) {
            HashMap<String, Integer> map = new HashMap<>();
            map.put("Alice", 27);
            map.put("Bob", 32);
            map.put("Charlie", 19);
            map.put("David", 45);
            TreeMap<String, Integer> sortedMap = new TreeMap<>(map);
            for (Map.Entry<String, Integer> entry : sortedMap.entrySet()) {
                System.out.println(entry.getKey() + " : " + entry.getValue());
            }
        }
    }
```

OUTPUT :-

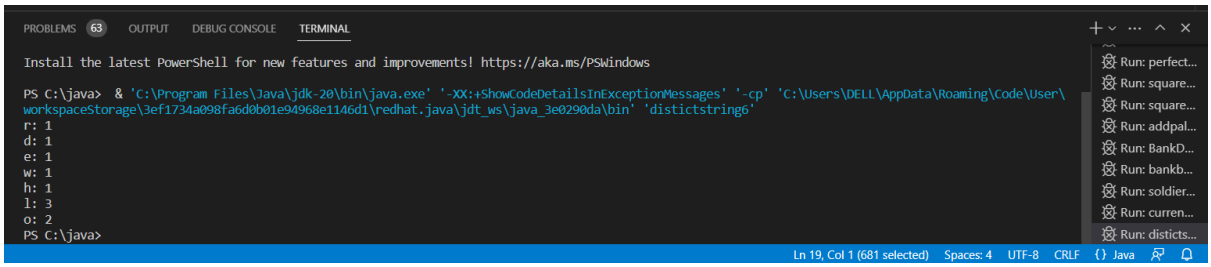


14. Write a Java program to find distinct characters and their count in a string.

PROGRAM :-

```
import java.util.*;
public class distictstring6 {
     public static void main(String[]args)
     {
         String input = "Hello, World!";
        Map<Character, Integer> characterCount = new HashMap<>();
        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);
            if (Character.isLetter(ch)) {
                ch = Character.toLowerCase(ch);
                characterCount.put(ch, characterCount.getOrDefault(ch, 0) +
1);
            }
        }
        for (Map.Entry<Character, Integer> entry : characterCount.entrySet())
{
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

OUTPUT :-



15. Write a program to print all the unique characters in a String. For instance, if the input string is “abcb”, the output will be the characters ‘a’ and ‘c’ as they are unique. The character ‘b’ repeats twice and so it will not be printed.

PROGRAM :-

```
import java.util.HashSet;
public class uniquecharacter6 {
    public static void main(String[] args) {
```

```java
        String input = "abcb";
        HashSet<Character> uniqueChars = new HashSet<Character>();

        for (int i = 0; i < input.length(); i++) {
            char c = input.charAt(i);
            if (input.indexOf(c) == input.lastIndexOf(c)) {
                uniqueChars.add(c);
            }
        }

        System.out.println("Unique characters in the string \"" + input + "\"
are:");
        for (char c : uniqueChars) {
            System.out.print(c + " ");
        }
    }
}
```

OUTPUT :-