**// STRING BASED TASK – 8 //**

Q1. Write a program to check if the letter ‘e’ is present in

the word ‘Umbrella.’.

Enter Any String: Umbrella

e is Present in the word :YES

// SOURCE CODE

import java.util.Scanner;

public class CheckLetter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Any String: ");

String input = scanner.nextLine();

if (input.contains("e")) {

System.out.println("e is Present in the word :YES");

} else {

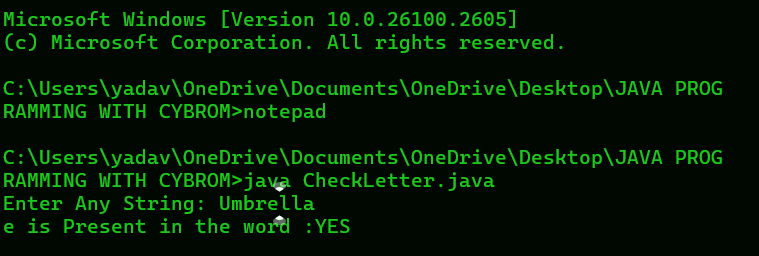
System.out.println("e is not Present in the word :NO");

}

}

}

// OUTPUT:



Q2. Write a program to check if the word orange is

present in the &quot;This is orange juice&quot;.

Enter Any String: This is orange juice

orange is Present in the Sentence :YES

// SOURCE CODE

import java.util.Scanner;

public class CheckWord {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Any String: ");

String input = scanner.nextLine();

if (input.contains("orange")) {

System.out.println("orange is Present in the Sentence :YES");

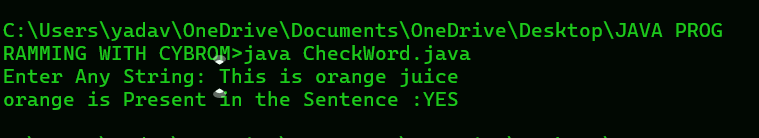
} else {

System.out.println("orange is not Present in the Sentence :NO");

}

}

}



Q3. Write a program to find the first and the last occurrence

public class FindOccurrences {

public static void main(String[] args) {

String str = "Hello, World";

int firstO = str.indexOf('o');

int lastO = str.lastIndexOf('o');

int firstComma = str.indexOf(',');

int lastComma = str.lastIndexOf(',');

System.out.println("First Occurrence of 'o': " + firstO);

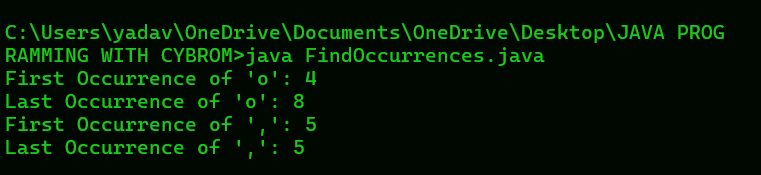
System.out.println("Last Occurrence of 'o': " + lastO);

System.out.println("First Occurrence of ',': " + firstComma);

System.out.println("Last Occurrence of ',': " + lastComma);

}

}



Q4. Write a program that takes your full name as input and

displays the abbreviations of the first and middle names

except the last name which is displayed as it is.

For example, if your name is Robert Brett Roser, then the

output should be R.B.Roser.

Example:

Input:

Enter Full Name : Sanjay Kumar Jain

Result: S.K. Jain

import java.util.Scanner;

public class NameAbbreviation {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String fullName = sc.nextLine();

String[] nameParts = fullName.split(" ");

String abbreviatedName = "";

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

if (i < nameParts.length - 1) {

abbreviatedName += nameParts[i].charAt(0) + ".";

} else {

abbreviatedName += nameParts[i];

}

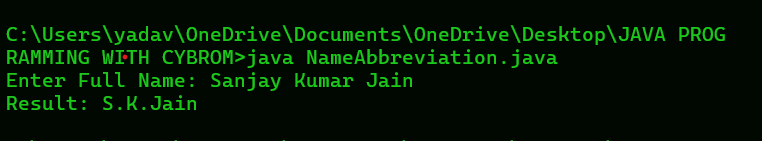
}

System.out.println("Result: " + abbreviatedName);

sc.close();

}

}



Q5. Write a program to find the number of vowels,

consonents, digits and white space characters in a string.

Example:

Input : str = A1 B@ d

Output : Vowels: 2

Consonant: 4

Digit: 1

Special Character: 6

public class CharacterCounts {

public static void main(String[] args) {

String str = "A1 B@ d adc";

int vowels = 0, consonants = 0, digits = 0, spaces = 0, specialChars = 0;

for (char ch : str.toLowerCase().toCharArray()) {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowels++;

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

consonants++;

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

digits++;

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

spaces++;

} else {

specialChars++;

}

}

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

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

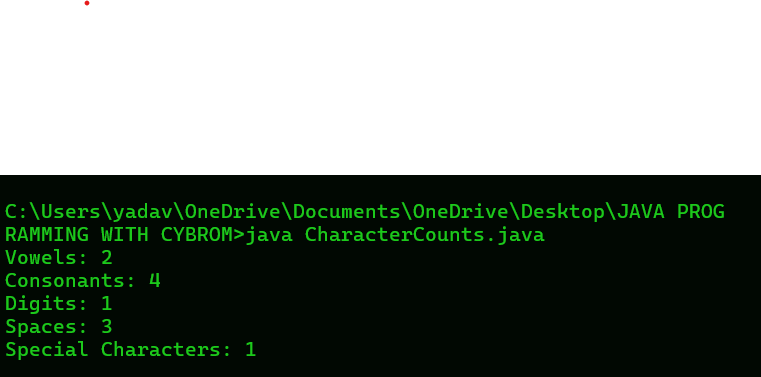
System.out.println("Digits: " + digits);

System.out.println("Spaces: " + spaces);

System.out.println("Special Characters: " + specialChars);

}

}



Q6. Write a program to delete all consonents from the

string &quot;Hello, have a good day&quot;.

Example:

Input : Hello

Output: eo

Example:

Input: Hello, have a good day

Output: eo, ae a oo a

import java.util.Scanner;

public class RemoveConsonants {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

scanner.close();

String vowelsOnly = removeConsonants(inputString);

System.out.println("Output: " + vowelsOnly);

}

public static String removeConsonants(String str) {

String vowels = "a e i o u A E I O U ,";

StringBuilder result = new StringBuilder();

for (char c : str.toCharArray()) {

if (vowels.indexOf(c) != -1) {

result.append(c);

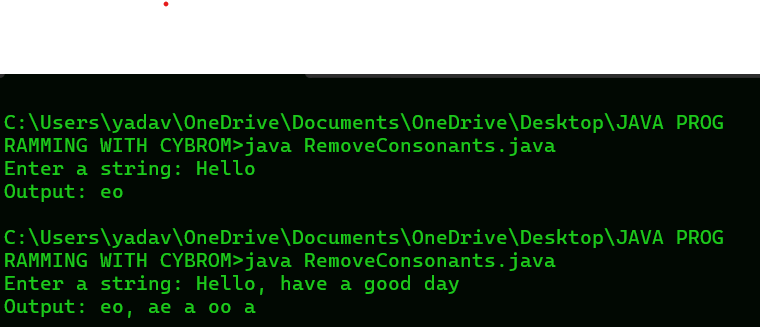
}

}

return result.toString();

}

}



Q8. Check whether two Strings are anagram of each other

Given two strings. The task is to check whether the given

strings are  anagrams  of each other or not. An anagram of a

string is another string that contains the same characters,

only the order of characters can be different. For example,

“abcd” and “dabc” are an anagram of each other.

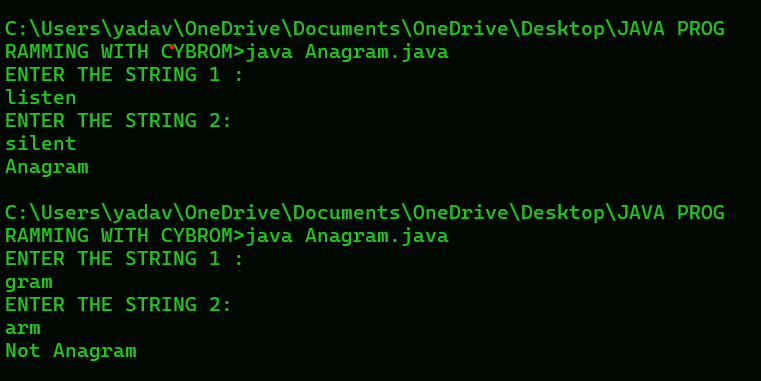
Input: str1 = “listen”  str2 = “silent”

Output: “Anagram”

Explanation: All characters of “listen” and “silent” are the same.

Input: str1 = “gram”  str2 = “arm”

Output: “Not Anagram”



Q9. Find first non-repeating character of given String

Given a string S consisting of lowercase Latin Letters, the task is to find the first

non-repeating character in S.

Examples:

Input: “geeksforgeeks”

Output: f

Explanation: As ‘f’ is first character in the string which does

not repeat.

Examples:

Input: “algorithm”

Output: a

Explanation: As ‘a’ is first character in the string which does

import java.util.Scanner;

public class FirstNonRepeatingCharacter {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

sc.close();

char firstNonRepeatingChar = findFirstNonRepeatingChar(str);

if (firstNonRepeatingChar != '\0') {

System.out.println("First non-repeating character: " + firstNonRepeatingChar);

} else {

System.out.println("No non-repeating character found.");

}

}

public static char findFirstNonRepeatingChar(String str) {

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

char c = str.charAt(i);

if (str.indexOf(c) == str.lastIndexOf(c)) {

return c;

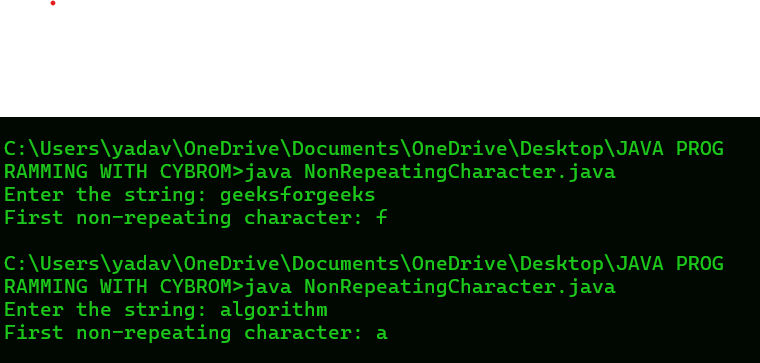
}

}

return '\0';

}

}



Q10. Java program to check whether a string

is a Palindrome without using pre-defined

method?

A string is said to be a palindrome if it is the same if we start reading it

from left to right or right to left. In this article, we will learn how to

check if a string is a palindrome in Java.

So let us consider a string “str”, now the task is just to find out with its

reverse string is the same as it is.

Example of Palindrome:

Input: str = “naman”

Output: Yes

Input: str = “ram”

Output: No

import java.util.Scanner;

public class PalindromeCheck {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str = sc.nextLine();

sc.close();

if (isPalindrome(str)) {

System.out.println("YES");

} else {

System.out.println("NO");

}

}

public static boolean isPalindrome(String str) {

int left = 0;

int right = str.length() - 1;

while (left < right) {

if (str.charAt(left) != str.charAt(right)) {

return false;

}

left++;

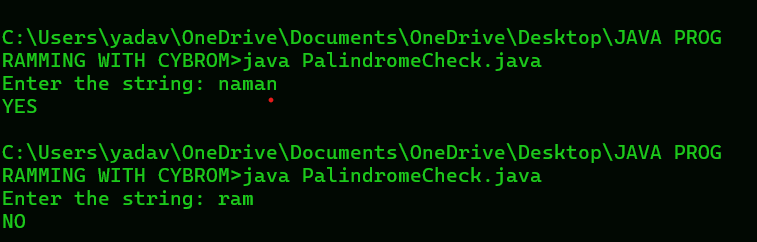
right--;

}

return true;

}

}



Q11. Remove duplicates from a given string

Given a string S which may contain lowercase and uppercase characters. The

task is to remove all duplicate characters from the string and find the resultant

string.

Note: The order of remaining characters in the output should be the same as in

the original string.

Example:

Input: Str = HappyNewYear

Output: HapyNewYr

Explanation: After removing duplicate characters such as p, e, a, we

have string as “HapyNewYr”.

public class RemoveDuplicates {

public static void main(String[] args) {

String str = "HappyNewYear";

String result = removeDuplicates(str);

System.out.println("Result: " + result);

}

public static String removeDuplicates(String str) {

StringBuilder result = new StringBuilder();

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

if (str.indexOf(str.charAt(i)) == i) {

result.append(str.charAt(i));

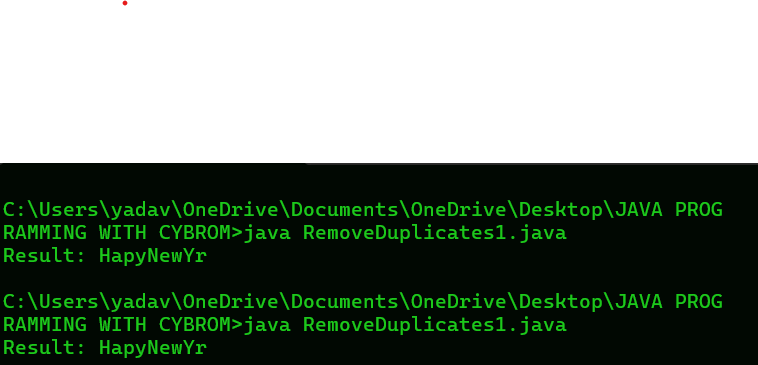
}

}

return result.toString();

}

}



Q12. How to remove all white spaces from

a String in Java?

Given a string with white spaces, the task is to remove all white

spaces from a string using without built-in methods.

Examples:

Input: str = &quot; A Computer Science Portal&quot;

Output: AComputerSciencePortal

import java.util.Scanner;

public class RemoveWhitespace {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

scanner.close();

String result = removeWhitespace(str);

System.out.println("Result: " + result);

}

public static String removeWhitespace(String str) {

StringBuilder result = new StringBuilder();

for (char c : str.toCharArray()) {

if (!Character.isWhitespace(c)) {

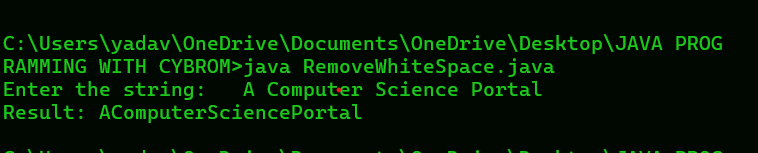
result.append(c);

}

}

return result.toString();

}

}

Q13. Remove all the Digits from String in Java

Given  alphanumeric string  str, the task is to write a Java program to remove all

the digit from this string and prints the modified string.

Examples:

Input: str = “12Java”

Output: Java

import java.util.Scanner;

public class RemoveDigits {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

scanner.close();

String result = removeDigits(str);

System.out.println("Result: " + result);

}

public static String removeDigits(String str) {

StringBuilder result = new StringBuilder();

for (char c : str.toCharArray()) {

if (!Character.isDigit(c)) {

result.append(c);

}

}

return result.toString();

}

}

