Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 1: “Given a string, check if the string is palindrome or not.” A string is

said to be palindrome if the reverse of the string is the same as the string.

string answers.txt

//INPUT

import java.util.Scanner;

public class Palindrome{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

boolean isPalindrome =true;

int length = str.length();

for (int i = 0; i < length / 2; i++) {

if (str.charAt(i) != str.charAt(length - 1 - i)) {

isPalindrome = false;

break;

}

}

if (isPalindrome) {

System.out.println("Palindrome");

} else {

System.out.println("Not a Palindrome");

}

}

}

//OUTPUT

Enter a string: ABCDCBA

Palindrome

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 2: Given a string, write a program to count the number of vowels,

consonants, and spaces in that string.

//INPUT

import java.util.Scanner;

public class CountVowelsConsonantsSpaces {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

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

String str=sc.nextLine();

int vowelCount = 0;

int consonantCount = 0;

int spaceCount = 0;

str = str.toLowerCase();

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

char ch = str.charAt(i);

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

vowelCount++;

} else if (ch >= 'a' && ch <= 'z') {

consonantCount++;

} else if (ch == ' ') {

spaceCount++;

}

}

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

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

System.out.println("White spaces: " + spaceCount);

}

}

//OUTPUT

enter string:-

Take u forward is Awesome

Vowels: 10

Consonants: 11

White spaces: 4

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 3: Given a String, write a program to remove vowels from a given

String.

//INPUT

import java.util.Scanner;

public class RemoveVowels{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

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

String str=sc.nextLine();

String result = str.replaceAll("[aeiouAEIOU]", "");

System.out.println("Input: " + str);

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

}

}

//OUTPUT

enter string:-

take u forward

Input: take u forward

Output: tk frwrd

/\*

Sonika negi

22152095

BCA 'E'

\*/

3

Problem Statement 4: Given a string, write a program to remove all the whitespaces from

the string.

//INPUT

import java.util.Scanner;

public class RemoveWhitespaces {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

String result = str.replaceAll("\\s", "");

System.out.println("String without whitespaces: " + result);

}

}

//OUTPUT

Enter a string: Take you forward

String without whitespaces: Takeyouforward

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 5 : Write a program to remove all characters from a string except

alphabets in a given string.

//INPUT

import java.util.Scanner;

public class RemoveNonAlphabets {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

String result = "";

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

char ch = str.charAt(i);

if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')) {

result += ch;

}

}

System.out.println("String with only alphabets: " + result);

}

}

//OUTPUT

Enter a string: take12% \*&u ^$#forward

String with only alphabets: takeuforward

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 6: Reverse a String. Write a program that reverses a given string.

Problem: Given a string, calculate the sum of numbers in a string (multiple consecutive digits

are considered one number

//INPUT

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 7: Given a string, write a program to Capitalize the first and last

character of each word of that string.

//INPUT

import java.util.Scanner;

public class CapitalizeFirstAndLastCharacter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String str = scanner.nextLine();

char[] charArray = str.toCharArray();

boolean newWord = true;

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

char ch = charArray[i];

if (Character.isLetter(ch)) {

if (newWord) {

charArray[i] = Character.toUpperCase(ch);

newWord = false;

} else if (i == charArray.length - 1 || !Character.isLetter(charArray[i + 1])) {

charArray[i] = Character.toUpperCase(ch);

newWord = true;

}

} else {

newWord = true;

}

}

String capitalizedString = new String(charArray);

System.out.println("Capitalized string: " + capitalizedString);

}

}

//OUTPUT

Enter a string: take u forward is awesome

Capitalized string: TakE U ForwarD IS AwesomE

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 8: Given two strings, check if two strings are anagrams of each other or

not.

//INPUT

import java.util.Arrays;

import java.util.Scanner;

public class AnagramCheck {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String str1 = sc.nextLine();

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

String str2 = sc.nextLine();

boolean areAnagrams = areAnagrams(str1, str2);

if (areAnagrams) {

System.out.println(str1 + " and " + str2 + " are anagrams.");

} else {

System.out.println(str1 + " and " + str2 + " are not anagrams.");

}

}

public static boolean areAnagrams(String str1, String str2) {

char[] charArray1 = str1.toCharArray();

char[] charArray2 = str2.toCharArray();

Arrays.sort(charArray1);

Arrays.sort(charArray2);

str1=new String(charArray1);

str2=new String(charArray2);

str1.equals(str2);

return Arrays.equals(charArray1, charArray2);

}

}

//OUTPUT

Enter the first string: RULES

Enter the second string: LESRT

RULES and LESRT are not anagrams.

/\*

Sonika negi

22152095

BCA 'E'

\*/

Problem Statement 9: Given a String, find the largest word in the string.

//INPUT

import java.util.Scanner;

public class LargestWordFinder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

String largestWord = "";

int largestWordLength = 0;

String currentWord = "";

int currentWordLength = 0;

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

char ch = input.charAt(i);

if (Character.isLetter(ch)) {

currentWord += ch;

currentWordLength++;

} else if (currentWordLength > 0) {

if (currentWordLength > largestWordLength) {

largestWord = currentWord;

largestWordLength = currentWordLength;

}

currentWord = "";

currentWordLength = 0;

}

}

if (currentWordLength > largestWordLength) {

largestWord = currentWord;

}

System.out.println("Largest word in the string: " + largestWord);

}

}

//OUTPUT

Enter a string: Google Doc

Largest word in the string: Google