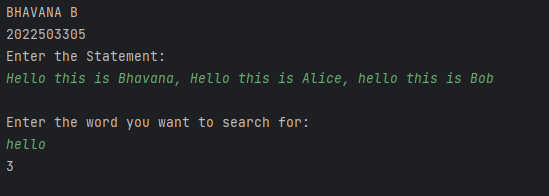
**EX-1. WRITE A PROGRAM THAT FIRST READS A PIECE OF TEXT ENTERED BY A USER ON ONE LINE, AND THEN READS A KEY ON THE SECOND LINE. THE PROGRAM DISPLAYS THE FREQUENCY WITH WHICH THE KEY HAS OCCURRED IN THE PIECE OF TEXT.**

**PROGRAM**

import java.util.\*;  
public class Assignment5\_2\_3305 {  
 public static void main(String[] args){  
 String statement;  
 int count=0;  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("BHAVANA B\n2022503305");  
 System.*out*.println("Enter the Statement: ");  
 statement = scan.nextLine();  
 System.*out*.println("\nEnter the word you want to search for: ");  
 String word = scan.nextLine();  
 String[] statement1 = statement.split(" ");  
 for(int i = 0; i<statement1.length;i++){  
 if(statement1[i].equalsIgnoreCase(word)){  
 count = count +1;  
 }  
 }  
 System.*out*.println(count);  
 }  
}

**OUTPUT**

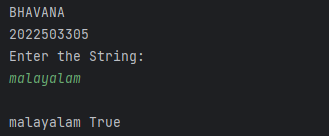
****

**EX- 2. WRITE A PROGRAM THAT ACCEPTS A STRING FROM THE USER AND PRINTS WHETHER IT IS A PALINDROME OR NOT. IGNORE THE CASE OF THE CHARACTERS.**

**PROGRAM**

import java.util.\*;  
public class Assignment5\_1\_3305{  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 String statement;  
 String statement1 = "";  
 System.*out*.println("BHAVANA\n2022503305");  
 System.*out*.println("Enter the String: ");  
 statement = scan.nextLine();  
 char character;  
 for (int i = 0; i<statement.length(); i++) {  
 character = statement.charAt(i);  
 statement1 = character + statement1;  
 }  
 if (statement. equalsIgnoreCase(statement1)){  
 System.*out*.println("\n" +statement + " True");  
 }  
 else{  
 System.*out*.println(statement + " False");  
 }  
 }  
}

**OUTPUT**

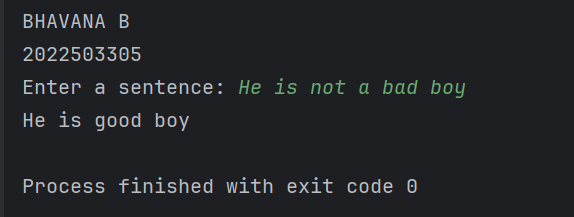


**EX-3. WRITE A PROGRAM AS PER THE FOLLOWING SPECIFICATION: THE INPUT TO THE PROGRAM IS A STRING. THE STRING CONTAINS SUBSTRINGS 'NOT' AND 'BAD' SUCH THAT 'BAD' COMES AFTER 'NOT'. THERE ARE ONLY SINGLE OCCURRENCES OF 'NOT' AND 'BAD'. THE PROGRAM OUTPUTS A STRING SUCH THAT THE WHOLE 'NOT...BAD' SUBSTRING IN THE INPUT IS REPLACED BY 'GOOD'.**

**PROGRAM**

import java.util.Scanner;  
  
public class Assignment5\_3\_3305 {  
 public static void main(String[] args) {  
 System.*out*.println("BHAANA B\n2022503305");  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print("Enter a sentence: ");  
 String text = sc.nextLine();  
 System.*out*.println(*replaceNotBad*(text));  
 sc.close();  
 }  
 public static String replaceNotBad(String input) {  
 int not = input.indexOf("not");  
 int bad = input.indexOf("bad", not);  
  
 if (not != -1 && bad != -1 && bad > not) {  
 return input.substring(0, not) + "good" + input.substring(bad + 3);  
 }  
 return input;  
 }  
}

**OUTPUT**

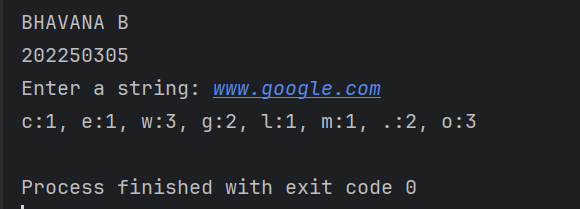
****

**EX-4. WRITE A PROGRAM TO PRINT THE FREQUENCY OF CHARACTERS IN A STRING IN THE GIVEN FORMAT.**

**PROGRAM**

import java.util.HashMap;  
import java.util.Scanner;  
  
public class Assignment5\_4\_3305 {  
  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n202250305");  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Enter a string: ");  
 String input = scanner.nextLine();  
  
 HashMap<Character, Integer> map = *countFrequency*(input);  
  
 *printFrequency*(map);  
  
 scanner.close();  
 }  
 private static HashMap<Character, Integer> countFrequency(String str) {  
 HashMap<Character, Integer> map = new HashMap<>();  
  
 for (char c : str.toCharArray()) {  
 if (!Character.*isWhitespace*(c)) {  
 if (map.containsKey(c))  
 map.put(c, map.get(c) + 1);  
 else  
 map.put(c, 1);  
 }  
 }  
 return map;  
 }  
 private static void printFrequency(HashMap<Character, Integer> map) {  
 StringBuilder sb = new StringBuilder();  
  
 for (HashMap.Entry<Character, Integer> entry : map.entrySet()) {  
 sb.append(entry.getKey()).append(":").append(entry.getValue()).append(", ");  
 }  
  
 if (sb.length() > 0) {  
 sb.setLength(sb.length() - 2);  
 }  
  
 System.*out*.println(sb.toString());  
 }  
}

**OUTPUT**

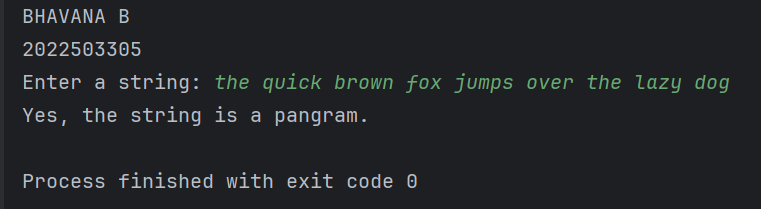
****

**EX-5. WRITE A PROGRAM TO CHECK WHETHER AN INPUT STRING IS A PANGRAM OR NOT. PANGRAMS ARE WORDS OR SENTENCES CONTAINING EVERY LETTER OF THE ALPHABET AT LEAST ONCE. IGNORE THE CASE OF THE CHARACTERS.**

**PROGRAM**

import java.util.Scanner;  
  
public class Assignment5\_5\_3305 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("BHAVANA B\n2022503305");  
 System.*out*.print("Enter a string: ");  
 String input = sc.nextLine().toLowerCase();  
  
 boolean[] alphabet = new boolean[26];  
  
 for (char ch : input.toCharArray()) {  
 if (ch >= 'a' && ch <= 'z') {  
 alphabet[ch - 'a'] = true;  
 }  
 }  
  
 StringBuilder missingLetters = new StringBuilder();  
 for (int i = 0; i < 26; i++) {  
 if (!alphabet[i]) {  
 missingLetters.append((char) ('a' + i)).append(", ");  
 }  
 }  
  
 if (missingLetters.length() == 0) {  
 System.*out*.println("Yes, the string is a pangram.");  
 } else {  
 missingLetters.setLength(missingLetters.length() - 2);  
 System.*out*.println("No, the string is NOT a pangram. Missing letter(s) is(are) " + missingLetters + ".");  
 }  
  
 sc.close();  
 }  
}

**OUTPUT**

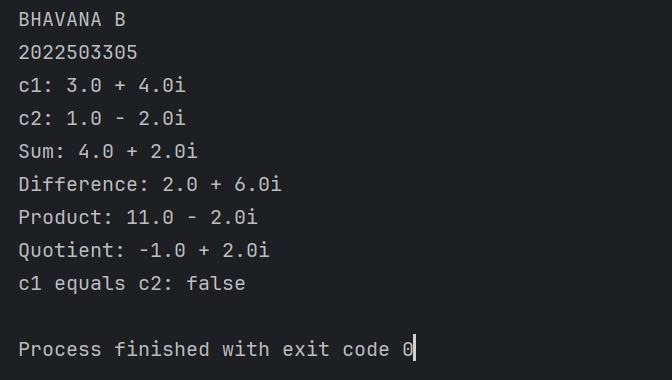
****

**EX-5. COMPLEX NUMBER**

**PROGRAM**

public class Assignment5\_6\_3305 {  
 private double real;  
 private double imaginary;  
  
 public Assignment5\_6\_3305(double real, double imaginary) {  
 this.real = real;  
 this.imaginary = imaginary;  
 }  
  
 public double getReal() { return real; }  
  
 public double getImaginary() { return imaginary; }  
  
 public Assignment5\_6\_3305 add(Assignment5\_6\_3305 other) {  
 double newReal = this.real + other.real;  
 double newImaginary = this.imaginary + other.imaginary;  
 return new Assignment5\_6\_3305(newReal, newImaginary);  
 }  
 public Assignment5\_6\_3305 subtract(Assignment5\_6\_3305 other) {  
 double newReal = this.real - other.real;  
 double newImaginary = this.imaginary - other.imaginary;  
 return new Assignment5\_6\_3305(newReal, newImaginary);  
 }  
 public Assignment5\_6\_3305 multiply(Assignment5\_6\_3305 other) {  
 double newReal = this.real \* other.real - this.imaginary \* other.imaginary;  
 double newImaginary = this.real \* other.imaginary + this.imaginary \* other.real;  
 return new Assignment5\_6\_3305(newReal, newImaginary);  
 }  
 public Assignment5\_6\_3305 divide(Assignment5\_6\_3305 other) {  
 double denominator = other.real \* other.real + other.imaginary \* other.imaginary;  
 double newReal = (this.real \* other.real + this.imaginary \* other.imaginary) / denominator;  
 double newImaginary = (this.imaginary \* other.real - this.real \* other.imaginary) / denominator;  
 return new Assignment5\_6\_3305(newReal, newImaginary);  
 }  
  
 @Override  
 public boolean equals(Object obj) {  
 if (this == obj)  
 return true;  
 if (obj == null || getClass() != obj.getClass())  
 return false;  
  
 Assignment5\_6\_3305 that = (Assignment5\_6\_3305) obj;  
  
 return (Double.*compare*(that.real, real) == 0 && Double.*compare*(that.imaginary, imaginary) == 0);  
 }  
  
 @Override  
 public String toString() {  
 if (imaginary >= 0) {  
 return real + " + " + imaginary + "i";  
 } else {  
 return real + " - " + (-imaginary) + "i";  
 }  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("BHAVANA B\n2022503305");  
 Assignment5\_6\_3305 c1 = new Assignment5\_6\_3305(3, 4);  
 Assignment5\_6\_3305 c2 = new Assignment5\_6\_3305(1, -2);  
  
 System.*out*.println("c1: " + c1);  
 System.*out*.println("c2: " + c2);  
  
 Assignment5\_6\_3305 sum = c1.add(c2);  
 Assignment5\_6\_3305 difference = c1.subtract(c2);  
 Assignment5\_6\_3305 product = c1.multiply(c2);  
 Assignment5\_6\_3305 quotient = c1.divide(c2);  
 boolean isEqual = c1.equals(c2);  
  
 System.*out*.println("Sum: " + sum);  
 System.*out*.println("Difference: " + difference);  
 System.*out*.println("Product: " + product);  
 System.*out*.println("Quotient: " + quotient);  
 System.*out*.println("c1 equals c2: " + isEqual);  
 }  
}

**OUTPUT**

****