**ASSIGNEMNET-5**

**J51: Frequency Counter**

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.

EXAMPLE

INPUT:

Can you write a whole paragraph without the letter a? Your sentences will sound wrong.

Everyone will notice something weird. You will use uncommon words.

will

OUTPUT:

3

Explanation: The count of will is 3 in the text.

**CODE:**

import java.util.\*;

public class J51\_3003{

public static void main(String[] args) {

/\*String str=new String("Can you write a whole paragraph without the letter a? Your sentences will sound wrong.\n" +

"Everyone will notice something weird. You will use uncommon words.");

String substr=new String("you");\*/

System.out.println("R.Prabhakara Arjun\n2022503003\n");

Scanner input=new Scanner(System.in);

System.out.print("ENTER PARAGRAPH:");

String str= input.nextLine();

System.out.print("\nENTER KEY:");

String substr= input.nextLine();

int count=0;

String check="";

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

if(check.toLowerCase().contains(substr.toLowerCase())){

count++;

check="";

}

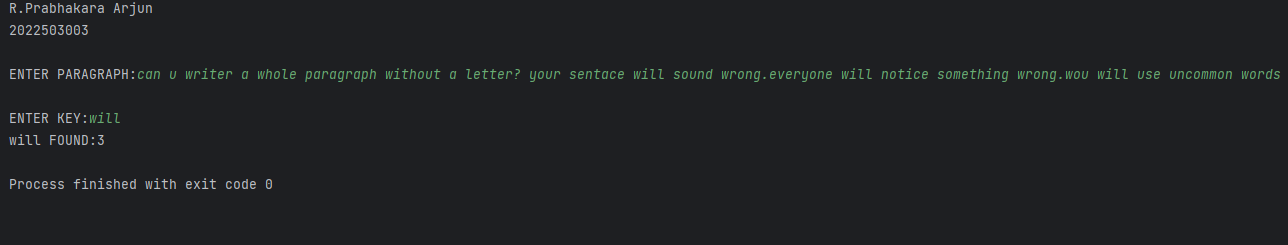
check=check+str.charAt(i);

}

check="";

System.out.println(substr+" FOUND:"+count);

}

**}**

**J52: Palindrome**

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.

The format of the output is <input-string> <True/False>

EXAMPLES:

INPUT: Nitin

OUTPUT: Nitin True

INPUT: Surya

OUTPUT: Surya False

**CODE:**

import java.util.\*;

public class J52\_3003 {

public static String reverse(String s){

String rev="";

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

rev=c+rev;

}

return rev;

}

public static void main(String[] args){

/\*String our\_str="abiba";

String rev=reverse(our\_str);

System.out.println(our\_str+"--->"+rev);\*/

System.out.println("R.Prabhakara Arjun\n2022503003\n");

Scanner input=new Scanner(System.in);

System.out.print("ENTER STR TO CHECK PALINDROME:");

String our\_str= input.nextLine();

String rev=reverse(our\_str);

if(rev.equals(our\_str)){

System.out.println(our\_str+" is a palindrome\n"+our\_str+":"+true);

}

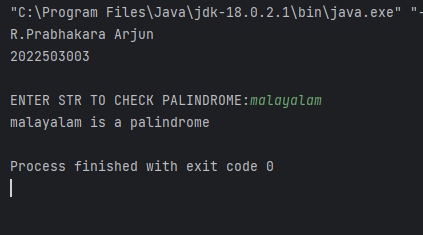
else{

System.out.println(our\_str+" is NOT a palindrome\n"+our\_str+":"+false);

}

}

}



**J53. It’s Good**

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'.

NOTE: In this question, all input strings for evaluation will definitely contain 'not' and

'bad' as substrings, such that 'bad' comes after 'not'.

EXAMPLES:

Write a program as per the following specification: The input to the program is a string.

The string may contain substrings 'not' and 'bad'. There are either 0 or 1 occurrences of

'not' and 'bad'. If 'bad' comes after 'not', then the program outputs a string such that the

whole 'not...bad' substring in the input is replaced by 'good'. Otherwise, it prints the

original string itself.

NOTE: In this question, the input strings for evaluation may or may not contain the

substrings 'not' and 'bad' as substrings. Even if the input contains both, it is not

guaranteed that 'bad' comes after 'not'.

EXAMPLES:

INPUT: The song is good.

OUTPUT: The song is good.

INPUT: Food is bad? not at all.

OUTPUT: Food is bad? not at all.

INPUT: The lyrics are not

that bad!

OUTPUT: The lyrics are

good!

**CODE:**

import java.util.\*;

public class J53\_3003 {

public static String helper(String s){

if(s.indexOf("not")<s.indexOf("bad")){

String modified="";

modified=s.substring(0,s.indexOf("not"))+"good"+s.substring(s.indexOf("bad")+3,s.length());

return modified;

}

else{

return s;

}

}

public static void main(String[] args){

System.out.println("R.Prabhakara Arjun\n2022503003\n");

Scanner input=new Scanner(System.in);

System.out.print("ENTER STR TO CHECK EX\_3:");

String our\_str= input.nextLine();

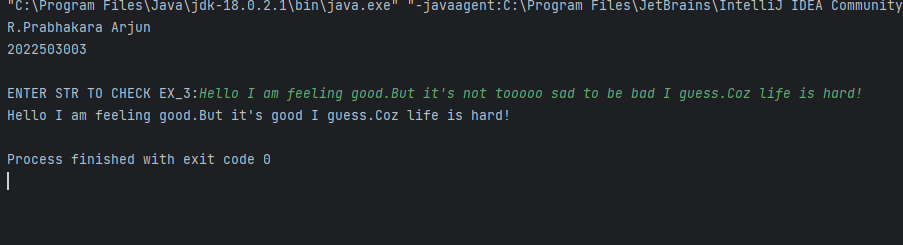
/\*String str="Food is not audfba bad!!!.";\*/

String new\_str=helper(our\_str);

System.out.println(new\_str);

}

}



**J54. Character Count.**

Write a program to print the frequency of characters in a string in the given

format.

EXAMPLES:

Input: www.google.com

Output: w:3, .:2, g:2, o:3, l:1, e:1, c:1, m:1

Input: abbac

INPUT: Food is not bad.

OUTPUT: Food is good.

INPUT: The lyrics are not that bad!

OUTPUT: The lyrics are good!

Output: a:2, b:2, c:1

(Please note that the order of characters in the output does not matter as long as the

corresponding counts are correct).

**CODE:**

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class J54\_3003 {

public static HashMap<Character,Integer> frequency(String s){

HashMap<Character,Integer> hasher=new HashMap<>();

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

if(hasher.containsKey(c)){

hasher.put(c,hasher.get(c)+1);

}

else{

hasher.put(c,1);

}

}

return hasher;

}

public static void display(HashMap<Character,Integer> f){

System.out.print("The key value pairs are:");

for(Map.Entry<Character,Integer> entry:f.entrySet()){

System.out.print(entry.getKey()+":"+entry.getValue()+" ");

}

}

public static void main(String[] args){

System.out.println("R.Prabhakara Arjun\n2022503003\n");

System.out.println("FREQUENCY CHECKER:\n------------------\n");

//String our\_str="hello abi!";

Scanner input=new Scanner(System.in);

System.out.print("ENTER STR TO CHECK FREQUENCY:");

String our\_str= input.nextLine();

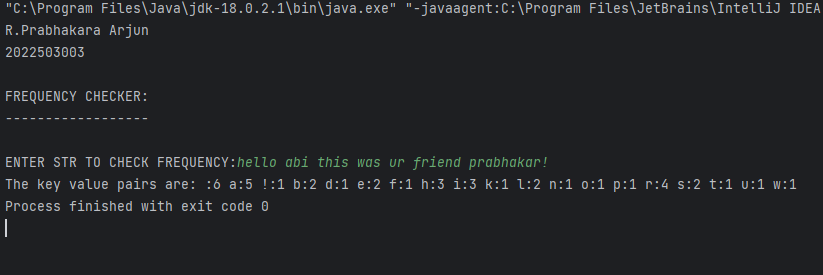
HashMap<Character,Integer> freq=frequency(our\_str);

display(freq);

//System.out.println(freq);

}

}



**J55. Pangram**

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.

If the input string is a Pangram, the output should be: Yes, the string is a pangram.

If the string is not a Pangram, it should report the missing letters, in lowercase, in

ORDER. See the Examples below.

EXAMPLES:

INPUT: The brown fox jump over the lazy dog

OUTPUT: No, the string is NOT a pangram. Missing letter(s) is(are) c, i, k, q, s.

INPUT: The quick brown fox jumps over the lazy dog

OUTPUT: Yes, the string is a pangram.

Hint: boolean[] alphabet = new boolean[26];

**CODE:**

import java.util.\*;

public class J55\_3003 {

public static Boolean pangram(String s){

Boolean[] check=new Boolean[26];

//System.out.println(Arrays.toString(check));

int count=0;

for(char c:s.toLowerCase().toCharArray()) {

if(check[(int)c-97]==null){

check[(int)c-97]=true;

count++;

}

}

//System.out.println(count);

//System.out.println(Arrays.toString(check));

if(count!=26){

System.out.print("The missing letters is(are):");

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

if(check[i]==null){

System.out.print(" "+(char)(i+97));

}

}

System.out.println();

}

return (count==26)?true:false;

}

public static void main(String[] args){

System.out.println("R.Prabhakara Arjun\n2022503003\n");

Scanner input=new Scanner(System.in);

System.out.print("ENTER STR TO CHECK PANGRAM:");

String our\_string= input.nextLine();

//String our\_string="efghijklmonpqrstuvwxyz";

Boolean flag=pangram(our\_string);

if(flag){

System.out.println(our\_string+"is a panagaram");

}

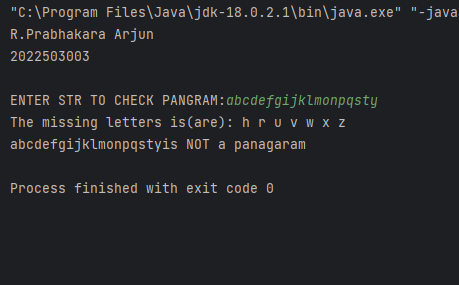
else{

System.out.println(our\_string+"is NOT a panagaram");

}

}

}



**J56. Complex Number**

Write a program to create a ComplexNumber class with the following features:

Two private double fields to represent the real and imaginary parts.

A constructor to initialize the complex number.

Getter methods for the real and imaginary parts.

Methods to add, subtract, multiply, and divide complex numbers.

An override of the toString() method to represent the complex number in the form

"a + bi" OR "a -bi" OR "-a -bi" OR "-a +bi".

An override of the equals() method to compare two complex numbers.

write a main method to demonstrate the usage of this class.

Hint:

public ComplexNumber add(ComplexNumber other) {

double newReal = this.real + other.real;

double newImaginary = this.imaginary + other.imaginary;

return new ComplexNumber(newReal, newImaginary);

}

@Override

public boolean equals(Object obj) { ..}

@Override

public String toString() { ...}

Example:

Input:

ComplexNumber c1 = new ComplexNumber(3, 4);

ComplexNumber c2 = new ComplexNumber(1, -2);

System.out.println("c1: " + c1);

System.out.println("c2: " + c2);

ComplexNumber sum = c1.add(c2);

ComplexNumber difference = c1.subtract(c2);

ComplexNumber product = c1.multiply(c2);

ComplexNumber quotient = c1.divide(c2);

boolean isEqual = c1.equals(c2);

System.out.println("c1 equals c2: " + isEqual);

Output:

c1: 3.0 + 4.0i

c2: 1.0 + -2.0i

Sum: 4.0 + 2.0i

Difference: 2.0 + 6.0i

Product: 11.0 + -2.0i

Quotient: -0.5 + 1.5i

c1 equals c2: false

**CODE:**

class complex{

private double real,imaginary;

complex(){

this.real=0;

this.imaginary=0;

}

double getReal(){

return this.real;

}

double getImaginary(){

return this.imaginary;

}

void setReal(double real){

this.real=real;

}

void setImaginary(double imaginary){

this.imaginary=imaginary;

}

void display(){

System.out.println("The complex no:"+getReal()+"+"+getImaginary()+"i");

}

public String toString(){

if(imaginary>=0){

return real+"+"+imaginary+"i";

}

else{

return real+"-"+Math.abs(imaginary)+"i";

}

}

complex add(complex b){

complex c=new complex();

c.setReal(this.getReal()+b.getReal());

c.setImaginary(this.getImaginary()+b.getImaginary());

return c;

}

complex sub(complex b){

complex c=new complex();

c.setReal(this.getReal()-b.getReal());

c.setImaginary(this.getImaginary()-b.getImaginary());

return c;

}

complex multiplication(complex b){

complex c=new complex();

c.setReal(this.getReal()\*b.getReal()-this.getImaginary()\*b.getImaginary());

c.setImaginary(this.getReal()\*b.getImaginary()+b.getReal()\*this.getImaginary());

return c;

}

Boolean equals(complex c){

return this.getReal()==c.getReal() && this.getImaginary()==c.getImaginary();

}

complex divide(complex b){

complex c=new complex();

c.setReal((this.getReal()\*b.getReal()+this.getImaginary()\*b.getImaginary())/(b.getReal()\*b.getReal()+b.getImaginary()\*b.getImaginary()));

c.setImaginary((this.getReal()\*b.getImaginary()-b.getReal()\*this.getImaginary())/(b.getReal()\*b.getReal()+b.getImaginary()\*b.getImaginary()));

return c;

}

}

public class J56\_3003 {

public static void main(String[] args){

complex a=new complex();

complex b=new complex();

System.out.println("The complex no:"+a.getReal()+"+"+a.getImaginary()+"i");

a.setReal(10);a.setImaginary(15);

b.setReal(10);b.setImaginary(15);

a.display();

System.out.println("The complex no a:"+a);

System.out.println("The complex no b:"+b);

complex sum=a.add(b);

System.out.println("complex no addition(a+b):"+sum);

complex sub=a.sub(b);

System.out.println("complex no difference(a-b):"+sub);

complex mul=a.multiplication(b);

System.out.println("Complex no multiplication(a\*b):"+mul);

complex quo=a.divide(b);

System.out.println("Complex no quotient(a/b):"+quo);

Boolean isEqual=a.equals(b);

System.out.println("a.equals(b):"+isEqual);

}

}

