

Government Engineering College Sec-28 Gandhinagar

Sem: - IV

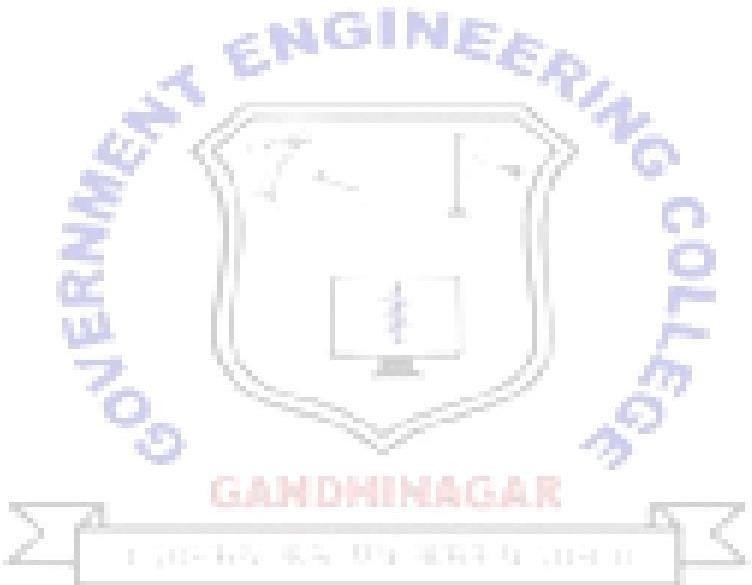
Subject: - Object Oriented Programming -I Subject Code: - 3140705

**Government Engineering College**

**Sec-28 Gandhinagar**

**Certificate**

**This is to certify that**



*Mr./Ms. Of class*

………. *Division ………, Enrollment No Has*

*Satisfactorily completed his/her term work in*

*…………………………………. Subject for the term ending in*

*……………2022.*

*Date: -*

Contents

* **Submission Instructions** 5
* **Institute Vision/Mission** 6

Vision 6

Mission 6

* **Computer Engineering Department Vision/Mission** 7

Vision 7

Mission 7

* **Program Educational Outcome (PEO)** 8
* **PSO** 8
* **POs** 9
* **Assignment Index** 12
* **Practical Index** 13
* **Assignment 1** 15
* **Assignment 2** 18
* **Assignment 3** 33
* **Assignment 4** 38
* **Practical 1** 43
* **Practical 2** 44
* **Practical 3** 45
* **Practical 4** 46
* **Practical 5** 48
* **Practical 6** 50
* **Practical 7** 52
* **Practical 8** 53
* **Practical 9** 55
* **Practical 10** 57
* **Practical 11** 59
* **Practical 12** 62
* **Practical 13** 63
* **Practical 14** 66
* **Practical 15** 68
* **Practical 16** 70
* **Practical 17** 72
* **Practical 18** 74
* **Practical 19** 76
* **Practical 20** 79
* **Practical 21** 83
* **Practical 22** 84
* **Practical 23** 85
* **Practical 24** 87
* **Practical 25** 89

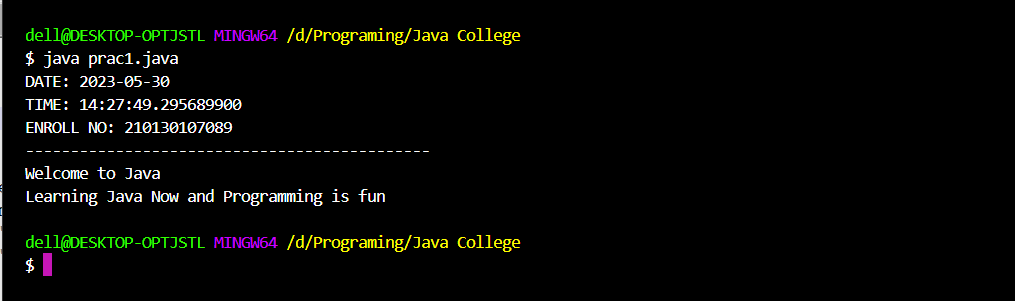
**Practical-01:** **Write a Program that displays Welcome to Java, Learning Java Now and**

**Programming is fun.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
  
  
public class Practical1 {  
 public static void main(String []args){  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
  
 System.*out*.println("Welcome to Java");  
 System.*out*.println("Learning Java Now and Programming is fun");  
 }  
}

**Output:**



**Practical-02:** **Write a program that solves the following equation and displays the value**

**x and y: 1) 3.4x+50.2y=44.5 2) 2.1x+.55y=5.9 (Assume Cramer’s rule to solve**

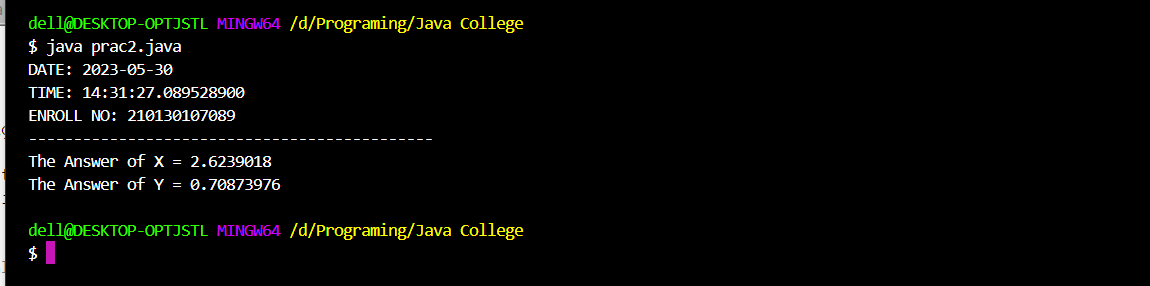
**equation ax+by=e x=ed-bf/ad-bc,**

**cx+dy=f y=af-ec/ad-bc )**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class Practical2 {  
 public static void main(String[] args) {  
  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 float a=3.4f;  
 float b=50.2f;  
 float e=44.5f;  
 float c=2.1f;  
 float d=0.55f;  
 float f=5.9f;  
  
 float x=((e\*d)-(b\*f))/((a\*d)-(b\*c));  
 float y=((a\*f)-(e\*c))/((a\*d)-(b\*c));  
  
 System.*out*.println("The Answer of X = " +x);  
 System.*out*.println("The Answer of Y = " +y);  
 }  
}

**Output:**



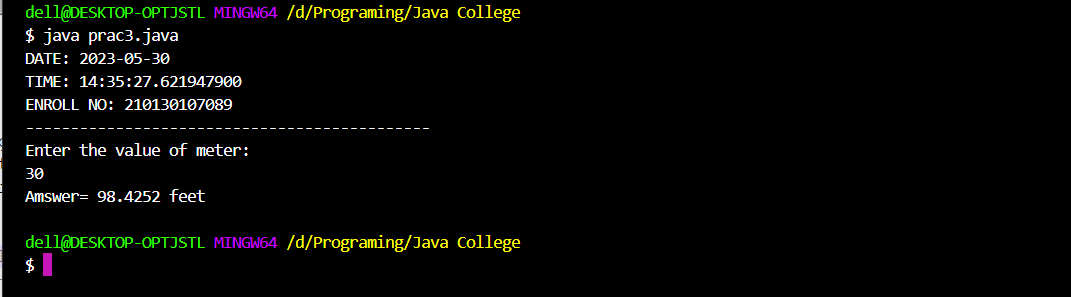
**Practical-03:** **Write a program that reads a number in meters, converts it to feet, and**

**Displays the result.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.\*;  
  
public class Practical3 {  
 public static void main(String []args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.println("Enter the value of meter:");  
 int a=sc.nextInt();  
  
 float result=a\*3.28084f;  
 System.*out*.println("Amswer= "+result+ " feet");  
 }  
 }

**Output:**



**Practical-04:** **Body Mass Index (BMI) is a measure of health on weight. It can be**

**calculated by taking your weight in kilograms and dividing by the square**

**of your height in meters. Write a program that prompts the user to enter a**

**weight in pounds and height in inches and displays the BMI.**

**Note:-** 1 pound=0.45359237 Kg and 1 inch=0.0254 meters.

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
public class Practical4 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.println("BODY WEIGHT INDEX (BMI) CALCULATE PROGRAM.............");  
  
 System.*out*.println("Enter weight in pound:");  
 int w=sc.nextInt();  
 System.*out*.println("Enter height in inch:");  
 float h=sc.nextFloat();  
  
 float Answer=(w\*0.45359237f)/((h\*0.0254f)\*(h\*0.0254f));  
  
 System.*out*.println("BMI="+Answer);  
 }  
  
}

**Output:**



**Practical-05: Write a program that prompts the user to enter three integers and display**

**the integers in decreasing order.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical5 {  
  
  
 Scanner sc=new Scanner(System.*in*);  
 void Bubblesort() {  
 int arr[];  
 arr =new int[3];  
*// int n=arr.length;* for (int i = 0; i < arr.length; i++) {  
 System.*out*.println("Enter the value of " +i + " integer");  
 arr[i]=sc.nextInt();  
 }  
  
 for(int i=0;i<(arr.length)-1;i++){  
 for(int j=0;j<(arr.length)-i-1;j++){  
 if (arr[j]< arr[j+1]){  
 int temp=arr[j];  
 arr[j]=arr[j+1];  
 arr[j+1]=temp;  
 }  
 }  
 }  
 System.*out*.println("Answer.............");  
 for (int i=0;i<arr.length;++i){  
 System.*out*.print(arr[i]);  
 System.*out*.print(" ");  
 }  
  
 }  
  
  
  
 public static void main (String[]args){  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Practical5 obj=new Practical5();  
 obj.Bubblesort();  
*// obj.Sorted\_arry();* }  
 }

**Output:**



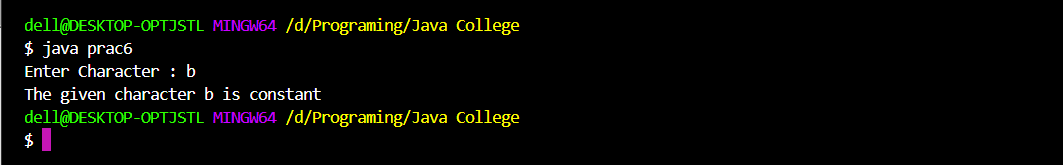
**Practical-06:** **Write a program that prompts the user to enter a letter and check**

**whether a letter is a vowel or constant.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
  
public class Practical6 {  
  
  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Enter the character:");  
 char ch = sc.next().charAt(0);  
  
  
 if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {  
 System.*out*.println("Vowel");  
 }  
 else {  
 System.*out*.println("Constant");  
 }  
 }  
}

**Output:**



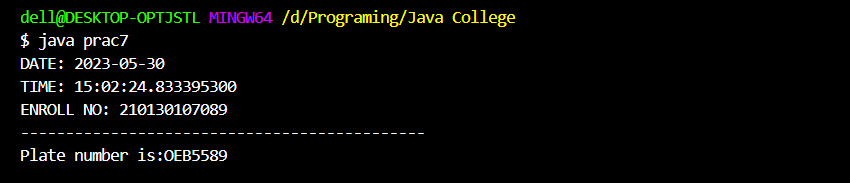
**Practical-07:** **Assume a vehicle plate number consists of three uppercase letters followed**

**by four digits. Write a program to generate a plate number.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class Practical7 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 int alpha1 = 'A' + (int)(Math.*random*() \* ('Z' - 'A'));  
 int alpha2 = 'A' + (int)(Math.*random*() \* ('Z' - 'A'));  
 int alpha3 = 'A' + (int)(Math.*random*() \* ('Z' - 'A'));  
 int digit1 = (int)(Math.*random*() \* 10);  
 int digit2 = (int)(Math.*random*() \* 10);  
 int digit3 = (int)(Math.*random*() \* 10);  
 int digit4 = (int)(Math.*random*() \* 10);  
 System.*out*.println("Plate number is:" + (char)(alpha1) + ((char)(alpha2)) +  
 ((char)(alpha3)) + digit1 + digit2 + digit3 + digit4);  
 }  
}

**Output:**



**Practical-08:** **Write a program that reads an integer and displays all its smallest factors**

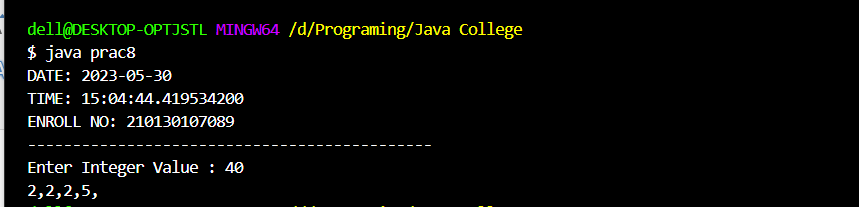
**in increasing order.**

**For Example** if input number is 120, the output should be as follows:2,2,2,3,5.

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical8 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 int div=2;  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter Integer Value : ");  
 int number = input.nextInt();  
 while(number > 1)  
 {  
 if(number%div==0)  
 {  
 System.*out*.print(div+",");  
 number=number/div;  
 }  
 else  
 {  
 div++;  
 }  
 }  
 }  
}

**Output:**



**Practical-09:** **Write a method with following method header.**

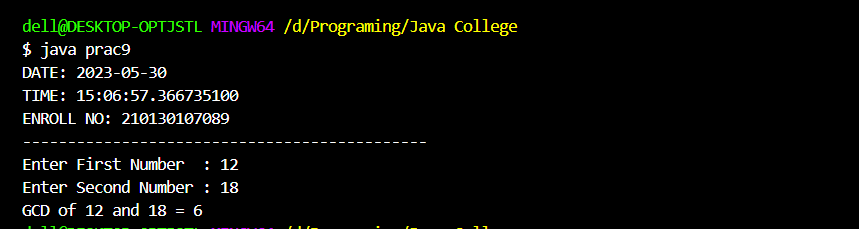
public static int gcd(int num1, int num2) **Write a program that prompts**

**the user to enter two integers and compute the gcd of two integers.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical9 {  
 public static int gcd(int num1, int num2) {  
 while (num1 != num2) {  
 if (num1 > num2) {  
 num1 = num1 - num2;  
 } else {  
 num2 = num2 - num1;  
 }  
 }  
 return num1;  
 }  
 public static void main(String[]args){  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter First Number : ");  
 int number1 = input.nextInt();  
 System.*out*.print("Enter Second Number : ");  
 int number2 = input.nextInt();  
 System.*out*.print("GCD of " + number1 + " and " + number2 + " = " + *gcd*(number1, number2));  
  
 }  
 }

**Output:**



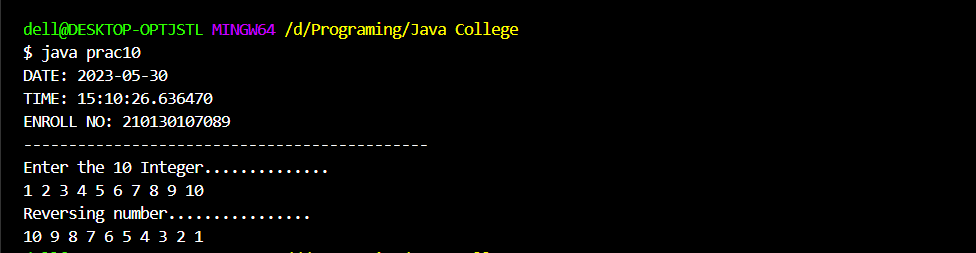
**Practical-10:** **Write a test program that prompts the user to enter ten numbers, invoke a**

**method to reverse the numbers, display the numbers.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
public class Practical10 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner sc=new Scanner(System.*in*);  
 int a[]=new int[10];  
 System.*out*.println("Enter the 10 Integer..............");  
 for (int i=0;i<a.length;i++){  
 a[i]=sc.nextInt();  
 }  
  
 for (int i=0;i<(a.length)-1;i++){  
 for (int j=0;j<(a.length)-i-1;j++){  
 int temp;  
 temp=a[j];  
 a[j]=a[j+1];  
 a[j+1]=temp;  
 }  
 }  
 System.*out*.println("Reversing number................");  
 for (int i=0;i<(a.length);++i){  
 System.*out*.print(a[i]);  
 System.*out*.print(" ");  
 }  
 }  
}

**Output:**



**Practical-11:** **Write a program that generate 6\*6 two-dimensional matrix, filled with 0’s**

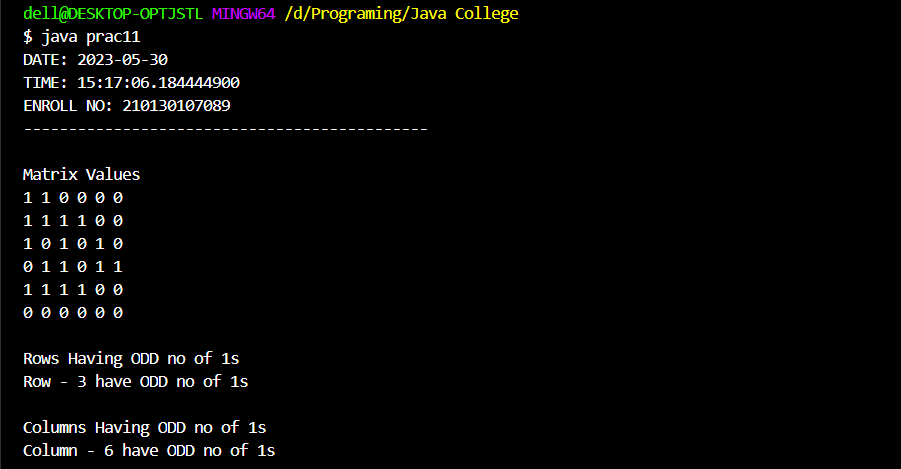
**and 1’s , display the matrix, check every raw and column have an odd**

**number’s of 1’s.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical11 {  
 public static int[][] create\_fill\_matrix()  
 {  
 int [][]matrix = new int[6][6];  
 for(int i=0;i<6;i++)  
 {  
 for(int j=0;j<6;j++)  
 {  
 matrix[i][j]=(int)((Math.*random*()\*5)%2);  
 }  
 }  
 return matrix;  
 }  
 public static void displayMatrix(int [][]matrix)  
 {  
 System.*out*.print("\nMatrix Values \n");  
 for(int i=0;i<6;i++)  
 {  
 for(int j=0;j<6;j++)  
 {  
 System.*out*.print(matrix[i][j]+ " ");  
 }  
 System.*out*.println();  
 }  
 }  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
   
 int my\_matrix[][];  
 int i,j,cnt;  
 my\_matrix=*create\_fill\_matrix*();  
 *displayMatrix*(my\_matrix);  
 System.*out*.println("\nRows Having ODD no of 1s");  
 for(i=0;i<6;i++)  
 {  
 cnt=0;  
 for(j=0;j<6;j++)  
 {  
 if(my\_matrix[i][j]==1)  
 {  
 cnt++;  
 }  
 }  
 if(cnt%2!=0)  
 {  
 System.*out*.println("Row - "+(i+1)+" have ODD no of 1s");  
 }  
 }  
 System.*out*.println("\nColumns Having ODD no of 1s");  
 for(i=0;i<6;i++)  
 {  
 cnt=0;  
 for(j=0;j<6;j++)  
 {  
 if(my\_matrix[j][i]==1)  
 {  
 cnt++;  
 }  
 }  
 if(cnt%2!=0)  
 {  
 System.*out*.println("Column - "+(i+1)+" have ODD no of 1s");  
 }  
 }  
 }  
}

**Output:**



**Practical-12:** **Write a program that creates a Random object with seed 1000 and**

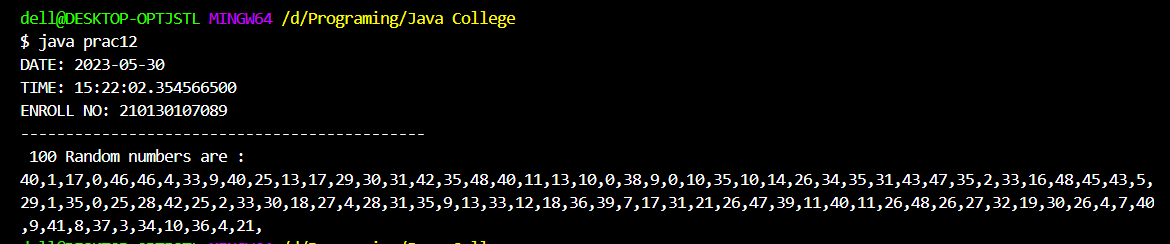
**displays the first 100 random integers between 1 and 49 using the**

**NextInt (49) method.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class Practical12 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 java.util.Random randomNo = new java.util.Random(1000);  
  
 *// displays first 100 random integers between 0 to 49* System.*out*.println(" 100 Random numbers are : ");  
 for (int i = 0; i < 100; i++) {  
 System.*out*.println(randomNo.nextInt(49) + " ");  
  
 }  
  
 }  
 }

**Output:**



**Practical-13:** **Write a program for calculator to accept an expression as a string in**

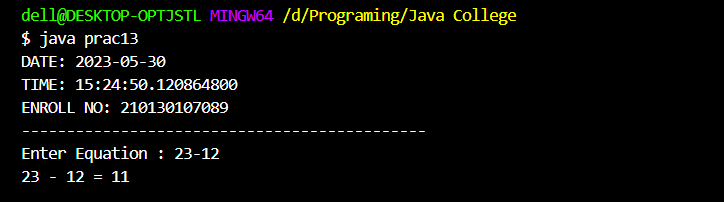
**which the operands and operator are separated by zero or more spaces.**

**For ex:** 3+4 and 3 + 4 are acceptable expressions.

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical13 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter Equation : ");  
 String string = input.nextLine();  
 String a = string.replaceAll(" ","");  
  
 if (a.length() < 3) {  
 System.*out*.println(  
 "Minimum 2 Opearator and 1 Opearand Required");  
 System.*exit*(0);  
 }  
 int result = 0;  
 int i = 0;  
 while(a.charAt(i)!='+' && a.charAt(i)!='-' && a.charAt(i)!='\*' && a.charAt(i)!='/')  
 {  
 i++;  
 }  
 switch (a.charAt(i)) {  
 case '+' :  
 result = Integer.*parseInt*(a.substring(0,i))+Integer.*parseInt*(a.substring(i+1,a.length()));  
 break;  
 case '-' :  
 result = Integer.*parseInt*(a.substring(0,i))-Integer.*parseInt*(a.substring(i+1,a.length()));  
 break;  
 case '\*' :  
 result = Integer.*parseInt*(a.substring(0,i))\*Integer.*parseInt*(a.substring(i+1,a.length()));  
 break;  
 case '/' :  
 result = Integer.*parseInt*(a.substring(0,i))/Integer.*parseInt*(a.substring(i+1,a.length()));  
 break;  
 }  
 System.*out*.println(a.substring(0,i) + ' ' + a.charAt(i) + ' ' + a.substring(i+1,a.length())  
 + " = " + result);  
 }  
}

**Output:**



**Practical-14:** **Write a program that creates an Array List and adds a Loan object , a**

**Date object , a string, and a Circle object to the list, and use a loop to**

**display all elements in the list by invoking the object’s to String() method.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.ArrayList;  
import java.util.Date;  
  
public class Practical14 {  
 public static void main(String[] args) {  
  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 ArrayList<Object> arr\_list = new ArrayList<Object>();  
 arr\_list.add(new Loan(10000));  
 arr\_list.add(new Date());  
 arr\_list.add(new String("GTU Practical"));  
 arr\_list.add(new Circle1(3.45));  
 for (int i = 0; i < arr\_list.size(); i++)  
 {  
 System.*out*.println((arr\_list.get(i)).toString());  
 }  
 }  
}  
class Circle1  
{  
 double radius;  
 Circle1(double r)  
 {  
 this.radius=r;  
 }  
 public String toString()  
 {  
 return "Circle with Radius "+this.radius;  
 }  
}  
class Loan  
{  
 double amount;  
 Loan(double amt)  
 {  
 this.amount=amt;  
 }  
 public String toString()  
 {  
 return "Loan with Amount "+this.amount;  
 }  
}

**Output:**



**Practical-15:** **Write the bin2Dec (string binary String) method to convert a binary string**

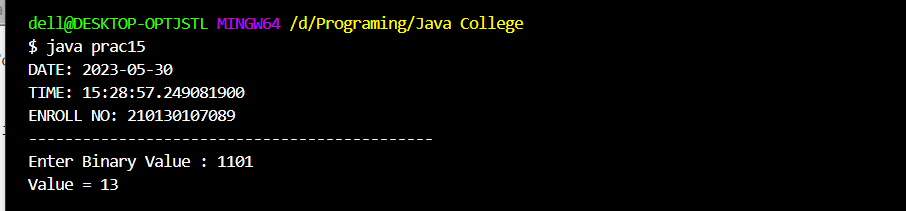
**into a decimal number. Implement the bin2Dec method to throw a**

**NumberFormatException if the string is not a binary string.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical15 {  
 public static int bin2Dec(String binaryString) throws NumberFormatException  
 {  
 int decimal = 0;  
 int strLength=binaryString.length();  
 for (int i = 0; i < strLength; i++)  
 {  
 if (binaryString.charAt(i) < '0' || binaryString.charAt(i) > '1')  
 {  
 throw new NumberFormatException("The Input String is not Binary");  
 }  
  
 decimal += (binaryString.charAt(i)-'0') \* Math.*pow*(2, strLength-1-i);  
 }  
 return decimal;  
 }  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.print("Enter Binary Value : ");  
 String str = input.nextLine();  
 try  
 {  
 System.*out*.println("Value = " + *bin2Dec*(str));  
 }  
 catch(NumberFormatException e)  
 {  
 System.*out*.println(e);  
 }  
 }  
}

**Output:**



**Practical-16:** **Write a program that prompts the user to enter a decimal number and**

**displays the number in a fraction.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
class Fraction{  
 private int real;  
 private int imaginary;  
  
 Fraction(int r,int img){  
 real=r;  
 imaginary=img;  
 }  
 public long gcm(long a, long b) {  
 return b == 0 ? a : gcm(b, a % b);  
 }  
 public String toString() {  
 long gcm = gcm(real, imaginary);  
 return real/gcm+"/"+imaginary/gcm;  
 }  
}  
public class Practical16 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.print("Enter a Decimal Number: ");  
 String decimal=sc.nextLine();

int indexOfDecimal = decimal.indexOf(".");  
 int len=decimal.substring(indexOfDecimal).length();  
 int imag\_part=(int) Math.*pow*(10,len-1);  
 int real\_part=(int)(Double.*parseDouble*(decimal)\*imag\_part);

Fraction fr=new Fraction(real\_part,imag\_part);  
 System.*out*.println("The Fraction Number is "+fr);

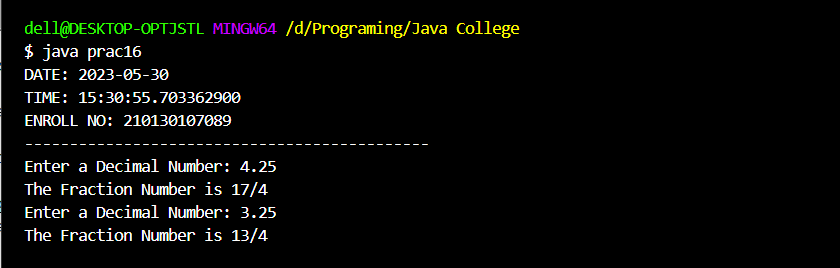
System.*out*.print("Enter a Decimal Number: ");  
 decimal=sc.nextLine();  
 indexOfDecimal = decimal.indexOf(".");

len=decimal.substring(indexOfDecimal).length();  
 imag\_part=(int) Math.*pow*(10,len-1);

real\_part=(int)(Double.*parseDouble*(decimal)\*imag\_part);  
 fr=new Fraction(real\_part,imag\_part);

System.*out*.println("The Fraction Number is "+fr);  
 }  
}

**Output:**



**Practical-17:** **Write a program that displays a tic-tac-toe board. A cell may be X, O, or**

**empty. What to display at each cell is randomly decided. The X and O are**

**images in the files X.gif and O.gif.**

**Code:**

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

public class prac17 extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Tic-Tac-Toe by kt");

GridPane gridPane = new GridPane();

for (int i = 0; i < 3; i++) {

for (int j = 0; j < 3; j++) {

int n = (int) (Math.random() \* 3);

if (n == 0)

gridPane.add(createX(), i, j);

else if (n == 1)

gridPane.add(createO(), i, j);

else

continue;

}

}

Scene primaryScene = new Scene(gridPane, 300, 300);

primaryStage.setScene(primaryScene);

primaryStage.show();

}

public VBox createX() {

Image imageX = new Image("File:F:/x.jpeg");

ImageView imageViewX = new ImageView(imageX);

VBox xBox = setProp(imageViewX);

return xBox;

}

public VBox createO() {

Image imageO = new Image("File:F:/o.jpeg");

ImageView imageViewO = new ImageView(imageO);

VBox oBox = setProp(imageViewO);

return oBox;

}

public VBox setProp(ImageView iv) {

iv.setFitHeight(50);

iv.setFitWidth(50);

iv.setPreserveRatio(true);

VBox vBox = new VBox();

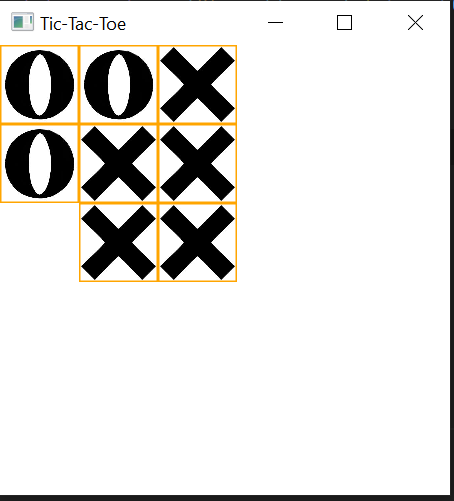
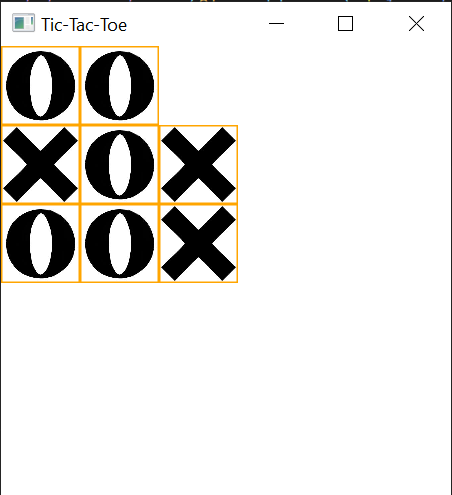
vBox.getChildren().add(iv);

vBox.setStyle("-fx-border-color: orange");

return vBox;

}

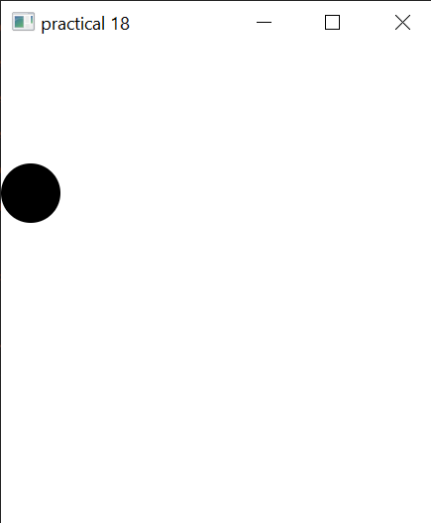
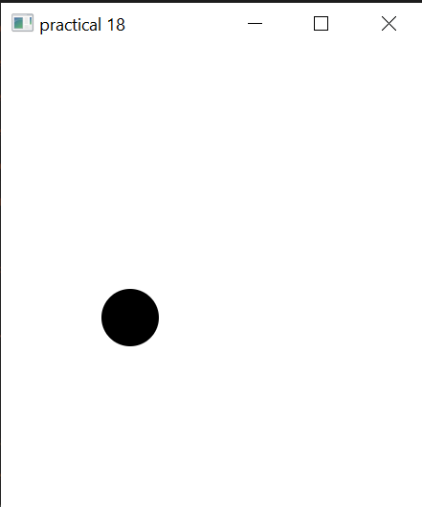
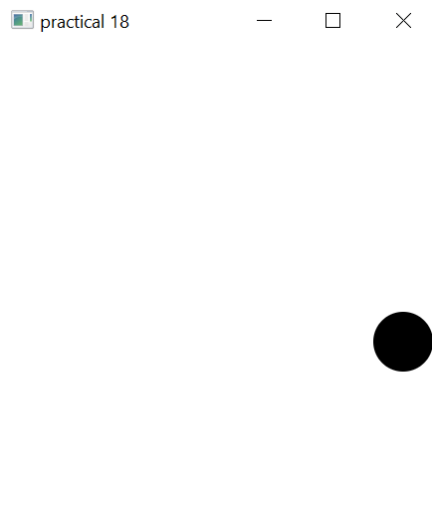
**Output:**



**Practical-18:** **Write a program that moves a circle up, down, left or right using arrow keys.**

**Code:**

import javafx.application.Application;  
import javafx.scene.Scene;  
import javafx.scene.shape.Circle;  
import javafx.scene.layout.Pane;  
import javafx.geometry.Insets;  
import javafx.stage.Stage;  
  
public class prac18 extends Application {  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 Pane pane = new Pane();  
 pane.setPadding(new Insets(20, 20, 20, 20));  
 *// Create a circle* Circle circle = new Circle(20, 20, 20);  
 pane.getChildren().add(circle);  
  
 *// Create and register the handle* pane.setOnKeyPressed(e -> {  
 switch (e.getCode()) {  
 case *UP* : circle.setCenterY(circle.getCenterY() >  
 circle.getRadius() ? circle.getCenterY() - 10 :  
 circle.getCenterY()); break;  
 case *DOWN* : circle.setCenterY(circle.getCenterY() <  
 pane.getHeight() - circle.getRadius() ?  
 circle.getCenterY() + 10 : circle.getCenterY());  
 break;  
 case *LEFT* : circle.setCenterX(circle.getCenterX() >  
 circle.getRadius() ? circle.getCenterX() - 10 :  
 circle.getCenterX()); break;  
 case *RIGHT* : circle.setCenterX(circle.getCenterX() <  
 pane.getWidth() - circle.getRadius() ?  
 circle.getCenterX() + 10: circle.getCenterX());  
 }  
 });  
  
 *// Create a scene and place it in the stage* Scene scene = new Scene(pane, 200, 200);  
 primaryStage.setTitle("practical 18"); *// Set the stage title* primaryStage.setScene(scene); *// Place the scene in the stage* primaryStage.show(); *// Display the stage* pane.requestFocus();  
 }  
}

**Output:**

**Practical-19:** **Write a program that displays the colour of a circle as red when the mouse**

**button is pressed and as blue when the mouse button is released.**

**Code:**

import javafx.application.Application;  
import javafx.scene.Scene;  
import javafx.scene.layout.StackPane;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Circle;  
import javafx.stage.Stage;  
  
public class prac19 extends Application {  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 double width = 400;  
 double height = 400;  
 Circle c = new Circle(width / 2, height / 2, Math.*min*(width, height) / 10, Color.*WHITE*);  
  
 c.setStroke(Color.*BLACK*);  
  
 StackPane pane = new StackPane(c);  
  
 primaryStage.setScene(new Scene(pane, width, height));  
 pane.setOnMousePressed(e -> c.setFill(Color.*RED*));  
 pane.setOnMouseReleased(e -> c.setFill(Color.*BLUE*));  
 primaryStage.setTitle("Click circle..");  
 primaryStage.show();  
 }  
 }

**Output:**

i) Output at mouse button pressed ii) Output at mouse button released

A screenshot of a computer

Description automatically generatedGraphical user interface, application

Description automatically generated

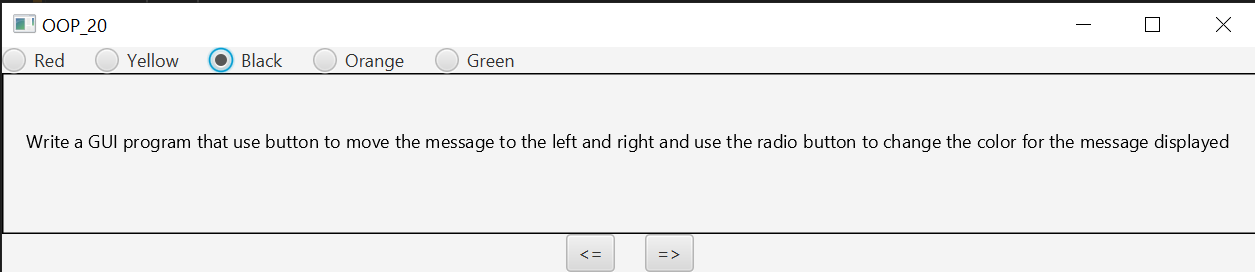
**Practical-20:** **Write a GUI program that use button to move the message to the left and**

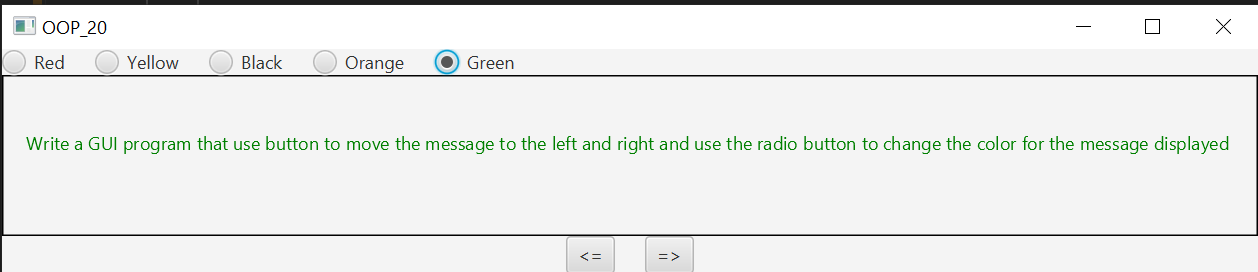
**right and use the radio button to change the colour for the message displayed.**

**Code:**

import javafx.application.Application;  
import javafx.stage.Stage;  
import javafx.scene.Scene;  
import javafx.geometry.Pos;  
import javafx.scene.control.Button;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.Pane;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.text.Text;  
import javafx.scene.control.RadioButton;  
import javafx.scene.control.ToggleGroup;  
import javafx.scene.paint.Color;  
  
public class prac20 extends Application {  
 protected Text text = new Text(50, 50, "•\tWrite a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed");  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 HBox paneForButtons = new HBox(20);  
 Button btLeft = new Button("<=");  
 Button btRight = new Button("=>");  
 paneForButtons.getChildren().addAll(btLeft, btRight);  
 paneForButtons.setAlignment(Pos.*CENTER*);  
 BorderPane pane = new BorderPane();  
 pane.setBottom(paneForButtons);  
  
 HBox paneForRadioButtons = new HBox(20);  
 RadioButton rbRed = new RadioButton("Red");  
 RadioButton rbYellow = new RadioButton("Yellow");  
 RadioButton rbBlack = new RadioButton("Black");  
 RadioButton rbOrange = new RadioButton("Orange");  
 RadioButton rbGreen = new RadioButton("Green");  
 paneForRadioButtons.getChildren().addAll(rbRed, rbYellow,  
 rbBlack, rbOrange, rbGreen);  
  
 ToggleGroup group = new ToggleGroup();  
 rbRed.setToggleGroup(group);  
 rbYellow.setToggleGroup(group);  
 rbBlack.setToggleGroup(group);  
 rbOrange.setToggleGroup(group);  
 rbGreen.setToggleGroup(group);  
  
 Pane paneForText = new Pane();  
 paneForText.setStyle("-fx-border-color: black");  
 paneForText.getChildren().add(text);  
 pane.setCenter(paneForText);  
 pane.setTop(paneForRadioButtons);  
  
 btLeft.setOnAction(e -> text.setX(text.getX() - 10));  
 btRight.setOnAction(e -> text.setX(text.getX() + 10));  
  
 rbRed.setOnAction(e -> {  
 if (rbRed.isSelected()) {  
 text.setFill(Color.*RED*);  
 }  
 });  
  
 rbYellow.setOnAction(e -> {  
 if (rbYellow.isSelected()) {  
 text.setFill(Color.*YELLOW*);  
 }  
 });  
  
 rbBlack.setOnAction(e -> {  
 if (rbBlack.isSelected()) {  
 text.setFill(Color.*BLACK*);  
 }  
 });  
  
 rbOrange.setOnAction(e -> {  
 if (rbOrange.isSelected()) {  
 text.setFill(Color.*ORANGE*);  
 }  
 });  
  
 rbGreen.setOnAction(e -> {  
 if (rbGreen.isSelected()) {  
 text.setFill(Color.*GREEN*);  
 }  
 });  
  
 Scene scene = new Scene(pane, 450, 150);  
 primaryStage.setTitle("OOP\_20");  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
}

**Output:**





**Practical-21: Write a program to create a file name 123.txt, if it does not exist. Append a**

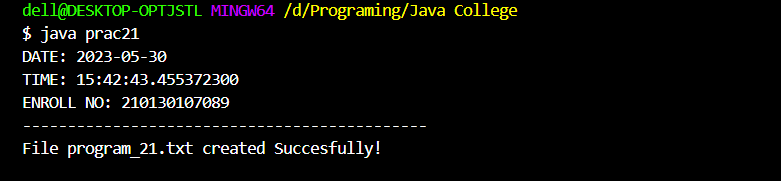
**new data to it if it already exist. write 150 integers created randomly into**

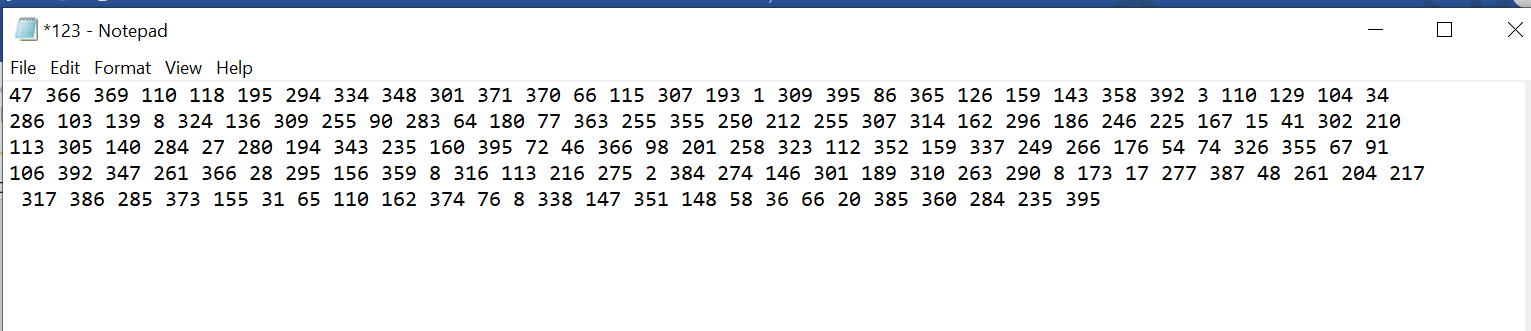
**the file using Text I/O. Integers are separated by space.**

**Code:**

import java.io.\*;  
import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class Practical21 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 File file = new File("123.txt");  
 if (file.exists()) {  
 System.*out*.println("File already exists");  
 System.*exit*(0);  
 }  
  
 try (PrintWriter output = new PrintWriter(file);) {  
 *// writing 150 integers created randomly to file* for (int i = 0; i < 150; i++) {  
 output.print(((int) (Math.*random*() \* 400) + 1) + " ");  
 }  
 System.*out*.println("File program\_21.txt created Succesfully!");  
 } catch (FileNotFoundException e) {  
 System.*out*.println("Cannot create the file");  
 e.printStackTrace();  
 }  
 }  
 }

**Output:**





**Practical-22: Write a recursive method that returns the smallest integer in an array. Write**

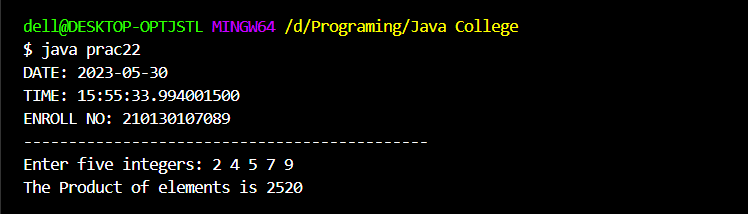
**a test program that prompts the user to enter an integer and display its product.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Scanner;  
  
public class Practical22 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Scanner input = new Scanner(System.*in*);  
 int product=1;  
 System.*out*.print("Enter five integers: ");  
 int[] list = new int[5];  
 for (int i = 0; i < list.length; i++)  
 {  
 list[i] = input.nextInt();  
 product \*= list[i];  
 }  
 System.*out*.println("The Product of elements is " + product);  
 }  
}

}

**Output:**



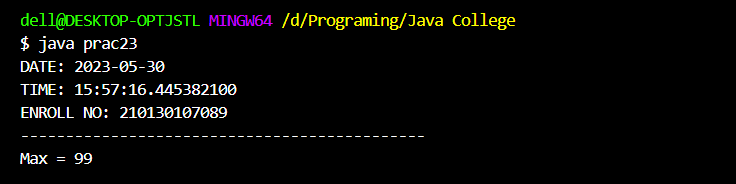
**Practical-23: Write a generic method that returns the minimum elements in a two**

**dimensional array.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
  
public class Practical23 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 Integer[][] list = new Integer[10][10];  
 int value = 0;  
 for (int i = 0; i < list.length; i++)  
 {  
 for (int j = 0; j < list[i].length; j++)  
 {  
 list[i][j] = value++;  
 }  
 }  
 System.*out*.println("Max = " + *max*(list));  
 }  
  
 public static <E extends Comparable<E>> E max(E[][] list)  
 {  
 E max = list[0][0];  
 for (E[] elements : list)  
 {  
 for (E element : elements)  
 {  
 if (element.compareTo(max) > 0)  
 {  
 max = element;  
 }  
 }  
 }  
 return max;  
 }  
}

**Output:**



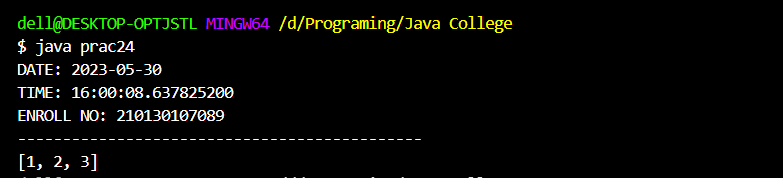
**Practical-24: Define MYPriorityQueue class that extends Priority Queue to implement the**

**Cloneable interface and implement the clone() method to clone a priority queue.**

**Code:**

import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.PriorityQueue;  
public class Practical24 {  
 public static void main(String[] args) {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 MyPriorityQueue<String> queue = new MyPriorityQueue<>();  
 queue.offer("1");  
 queue.offer("2");  
 queue.offer("3");  
  
 MyPriorityQueue<String> queue1 = null;  
 try  
 {  
 queue1 = (MyPriorityQueue<String>)(queue.clone());  
 }  
 catch (CloneNotSupportedException e)  
 {  
 e.printStackTrace();  
 }  
 System.*out*.print(queue1);  
 }  
  
 static class MyPriorityQueue<E> extends PriorityQueue<E> implements Cloneable  
 {  
 @Override  
 public Object clone() throws CloneNotSupportedException  
 {  
 MyPriorityQueue<E> clone = new MyPriorityQueue<>();  
 this.forEach(clone::offer);  
 return clone;  
 }  
  
 }  
}

**Output:**



**Practical-25: Write a program that reads words from a text file and displays all the**

**nonduplicate words in descending order. The text file is passed as a**

**command-line argument.**

**Code:**

import java.io.\*;  
import java.security.InvalidParameterException;  
import java.time.LocalDate;  
import java.time.LocalTime;  
import java.util.Arrays;  
import java.util.HashSet;  
import java.util.TreeSet;  
import java.util.Collections;  
import java.util.Iterator;  
  
public class Practical25 {  
 public static void main(String[] args) throws FileNotFoundException {  
 LocalDate myObj = LocalDate.*now*();  
 LocalTime myObj1 = LocalTime.*now*();  
 System.*out*.println("DATE: "+myObj);  
 System.*out*.println("TIME: "+myObj1);  
 System.*out*.println("ENROLL NO: 210130107089");  
 System.*out*.println("---------------------------------------------");  
  
 if (args.length != 1)  
 throw new InvalidParameterException("Usage: fullFilePathName");  
  
 File file = new File(args[0]);  
 if (!file.isFile())  
 throw new FileNotFoundException(file + " is not a file.");  
 try (BufferedReader in = new BufferedReader(new InputStreamReader(new FileInputStream(file)), 10000)) {  
  
 String inputS;  
 StringBuilder sb = new StringBuilder(10000);  
 while ((inputS = in.readLine()) != null)  
 sb.append(inputS);  
  
 String[] words = sb.toString().split("\\s+");  
 *// non duplicate words* TreeSet<String> ndWords = new TreeSet<>(Arrays.*asList*(words));  
  
 for (String s : ndWords)  
 System.*out*.println(s);  
 } catch (IOException e) {  
 e.printStackTrace();  
 System.*exit*(0);  
 }  
 }  
}

**output:**

