MITHIN

Contents

Q.1 Write a program to find maximum occurring character from given string.

CODE:

```
package pkgfinal.exam.q.pkg1;

public class FinalExamQ1 {

    public static void main(String[] args) {
        String str="Hello,HowRU?Howyouknowme?";
       char reschar=' ',tempchar=' ';
       int rescount=0,tempcount=0;
       int pos=0;
       for(int i=0;i<str.length();i++)
       {
          pos=i; tempchar=str.charAt(i);
          tempcount=1;
          String tempstring=tempchar+"";
          while((pos=str.indexOf(tempstring,pos+1))>0)
          {
             tempcount++;
          }
          if(tempcount>rescount)
          {
             rescount=tempcount;
             reschar=tempchar;
          }
       }
       System.out.println("result char is '"+reschar);
    }
    }
```

OUTPUT:

```
Output - Final Exam Q-1 (run)
    result char is 'o
    BUILD SUCCESSFUL (total time: 0 seconds)
```

Kaiyu character vadhare repeat thai che ee apse.

Q.2 Write a program to print duplicate characters with number of occurrence in given string.

CODE:

```
package pkgfinal.exam.q2;

public class FinalExamQ2 {

    public static void main(String[] args) {
        String str="Hello,HowRU?Howyouknowme?";
        char reschar=' ',tempchar=' ';
        int rescount=0,tempcount=0;
        int pos=0;
        for(int i=0;i<str.length();i++)
        {
            pos=i; tempchar=str.charAt(i);
            tempcount=1;
            String tempstring=tempchar+"";
            while((pos=str.indexOf(tempstring,pos+1))>0)
            {
                tempcount++;
            }
            if(tempcount>rescount)
            {
                rescount=tempcount;
                reschar=tempchar;
            }
        }
        System.out.println("result char is '"+reschar+"' result count is
"+rescount);
    }
    }
```

OUTPUT:

```
Output - Final Exam Q2 (run)
    run:
    result char is 'o' result count is 5
    BUILD SUCCESSFUL (total time: 0 seconds)
```

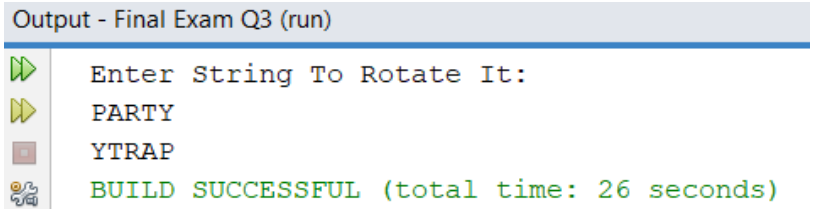
Maximum character and ketli var repeat thai che ee apse.

Q.3 Write a program to reverse the words of given string.

CODE:

```
package pkgfinal.exam.q3;

import java.util.Scanner;

public class FinalExamQ3 {

    public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
        System.out.println("Enter String To Rotate It:");
        String a = sc.next();
        a.trim();
        StringBuffer rev=new StringBuffer();
        rev.append(a);
        rev.reverse();
        System.out.println(rev);

  }
 }
```

OUTPUT:



String nu reverse thai jase, tame je bhi input apso enu.

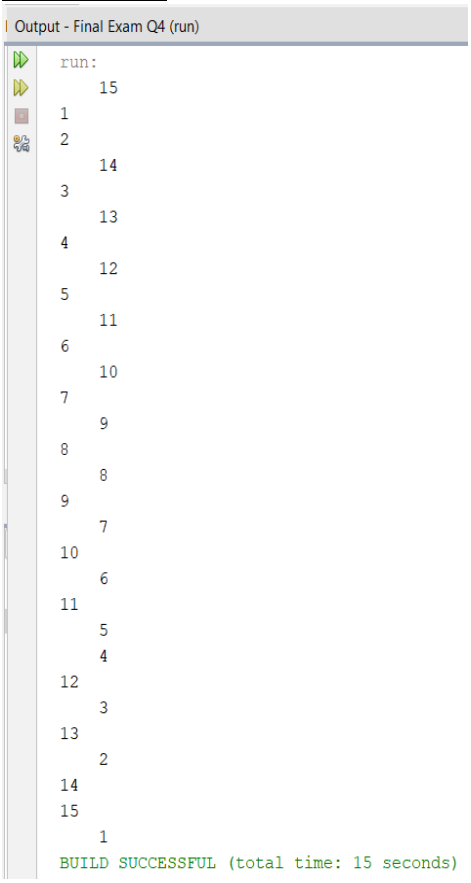
Q.4 Write a java program to print from 1-15 and 15-1 using multithreading in java.

CODE:

```
package pkgfinal.exam.q4;
class MyThread2 implements Runnable
{
   Thread T1;
   MyThread2(String str)
   {
      T1 = new Thread(this);
      T1.start();
   }
   public void run()
   {
      try
      {
         for(int i=1;i<=15;i++)
         {
            System.out.println(+ (i));
            T1.sleep(1000);
         }
      }
      catch(InterruptedException ex)
      {

      }
   }
}

public class FinalExamQ4 {

   public static void main(String[] args) {
    Thread t = Thread.currentThread();
      MyThread2 m1;
      m1 = new MyThread2("");
      try
      {
         for(int i=15;i>0;i--)
         {
            System.out.println("    " + i);
            Thread.sleep(1000);
```

```
            }
        }
        catch(InterruptedException ex)
        {

        }
    }
}
```

OUTPUT:



Q.5 Create two thread, first thread will print prime number and second thread will print perfect number between 1-100.

CODE:

Q.6 Write program to accept rollno, marks of four subjects in an array from user and throw MarksOutOfBoundsException if marks are < 0 or marks > 100. Also check ArrayIndexOutofBoundsException.

CODE:

```
package pkgfinal.exam.q6;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

class MarksOutOfBoundException extends Exception
{
   public String toString()
   {
      return "Enter marks in between 1 & 100";
   }
}
class Student1
{
   int rno;
   float marks[];
   BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
   Student1()
   {
      rno=0;
      marks=new float[4];
      for(int i=0;i<4;i++)
         marks[i]=0;
```

```
    }
    public void accept()
    {
       System.out.println("Enter roll no & marks of 4 subjects ");
       try
       {
          rno=Integer.parseInt(br.readLine());
          for(int i=0;i<4;i++)
          {
             marks[i]=Float.parseFloat(br.readLine());
             if(marks[i]<0 || marks[i]>100)
                throw new MarksOutOfBoundException();
          }
       }
       catch(MarksOutOfBoundException me)
       {
          System.out.println(me);
       }
       catch(Exception ie)
       {
          System.out.println("ARRAY INDEX OUT OF BOUND");
       }
    }

    public void display()
    {
       System.out.println("Roll no\tMarks1\tMarks2\tMarks3\tMarks4");
       System.out.print(rno+"\t");
       for(int i=0;i<4;i++)
       System.out.print(marks[i]+"\t");
    }}
public class FinalExamQ6 {

    public static void main(String[] args) {
      Student1 s1=new Student1();
       s1.accept();
       s1.display();
    }
    }
```

OUTPUT:

ROLL NUMBER AND MARKS SIDHA J MANGSE.

```
Output - Final Exam Q6 (run)
run:
Enter roll no & marks of 4 subjects
50
99
100
97
98
Roll no Marks1  Marks2  Marks3  Marks4
50      99.0    100.0   97.0    98.0    BUILD SUCCESSFUL (total time: 15 seconds)
```

Roll number and marks siddha j mangse and 100+ marks nakhso to terminate thai jase.

```
Output - Final Exam Q6 (run)
run:
Enter roll no & marks of 4 subjects
50
109
Enter marks in between 1 & 100
Roll no Marks1  Marks2  Marks3  Marks4
50      109.0   0.0     0.0     0.0     BUILD SUCCESSFUL (total time: 7 seconds)
```

For exception… Use space bar for entering data…..

```
Output - Final Exam Q6 (run)
run:
Enter roll no & marks of 4 subjects
50
67 89 99 105
ARRAY INDEX OUT OF BOUND
Roll no Marks1  Marks2  Marks3  Marks4
50      0.0     0.0     0.0     0.0     BUILD SUCCESSFUL (total time: 14 seconds)
```

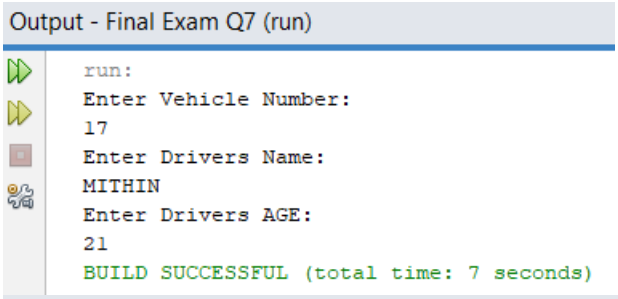
Q.7 Write a class Driver with attributes vehicle no, name & age. Initialize values through parameterized constructor. If age of driver is less than 18 then generate user-defined exception "AgeNotWithinTheRange".

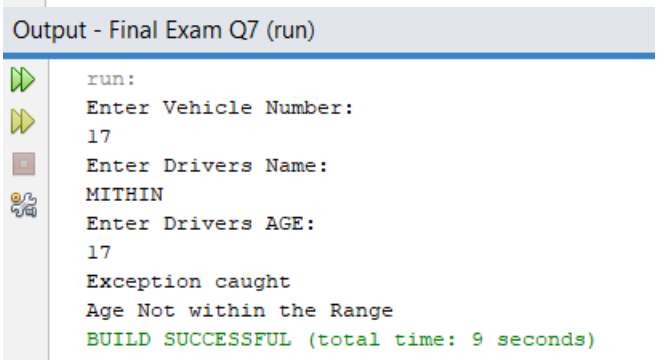
CODE:

```
package pkgfinal.exam.q7;
import java.util.Scanner;
class AgeNotWithinTheRange extends Exception
{
  AgeNotWithinTheRange(String s)
   {
      super(s);
   }
}
class vehicle
{
   void  vehicle(int age) throws AgeNotWithinTheRange
   {
         if(age<18)
            throw new AgeNotWithinTheRange("Age Not within the Range");
   }
}
public class FinalExamQ7 {

   public static void main(String[] args) {
       Scanner sc=new Scanner(System.in);
      vehicle v=new vehicle();
      try
      {
         System.out.println("Enter Vehicle Number: ");
         String num =sc.next();
         System.out.println("Enter Drivers Name: ");
         String name=sc.next();
         System.out.println("Enter Drivers AGE: ");
         int age=sc.nextInt();
         v.vehicle(age);
      }
      catch(AgeNotWithinTheRange e)
      {
         System.out.println("Exception caught");
         System.out.println(e.getMessage());
```

```
    }
  }
  }
```

OUTPUT:


Output - Final Exam Q7 (run)
run:
Enter Vehicle Number:
17
Enter Drivers Name:
MITHIN
Enter Drivers AGE:
21
BUILD SUCCESSFUL (total time: 7 seconds)


Output - Final Exam Q7 (run)
run:
Enter Vehicle Number:
17
Enter Drivers Name:
MITHIN
Enter Drivers AGE:
17
Exception caught
Age Not within the Range
BUILD SUCCESSFUL (total time: 9 seconds)

Q.8 BOOK LIBRARY ISSUE OR SUBMIT JAVA/SQL/CG.

Code a java program in such a way that cover all the transition of library for issuing and submitting books for student as well as staff. For that follow this description.

1. There is one Abstract class named **Lib** that contains 3 subjects in library like JAVA 100 books, SQL 150 books and CG 50 books. There are also two methods **submit** and **issue**.
2. There are two interfaces named **Student** and **Staff** contains one method Info. (Note: both interfaces has same name method Info)
3. There is one class named **LibTransitions** that calculate (for issue and submit), extends and implements appropriate classes and interfaces.
4. There is main class named **Library** that calls all methods according user selection for staff or Student.
5. Whenever transition occurs stock must be update for books.

CODE:

```
package pkgfinal.exam.q8;
import java.util.*;

abstract class Lib
{
   int java=100, sql=150, cg=50;

   abstract void submit();
   abstract void issue();
}

interface student
{
   void info();
}

interface staff
{
   void info();
}

class LibTransitions extends Lib
{
   Scanner sc = new Scanner(System.in);
   void submit()
   {
      System.out.println("Which Book You want to Submit? Java/SQL/CG ?");
      String s = sc.next();
```

```
    if(s.equalsIgnoreCase("java"))
    {
       if(java==100)
       {
          System.out.println("Invalid Submit");
       }
       else
       {
          java++;
          System.out.println("Book Submitted");
          System.out.println("Stock: " + java);
       }
    }

    if(s.equalsIgnoreCase("sql"))
    {
       if(sql==150)
       {
          System.out.println("Invalid Submit");
       }
       else
       {
          sql++;
          System.out.println("Book Submitted");
          System.out.println("Stock: " + sql);
       }
    }

    if(s.equalsIgnoreCase("cg"))
    {
       if(cg==50)
       {
          System.out.println("Invalid Submit");
       }
       else
       {
          cg++;
          System.out.println("Book Submitted");
          System.out.println("Stock: " + cg);
       }
    }
}
```

```
void issue()
{
   System.out.println("Which Book You want to Issue? Java/SQL/CG ?");
   String s = sc.next();

   if(s.equalsIgnoreCase("java"))
   {
      if(java==0)
      {
         System.out.println("No More Books");
      }
      else
      {
         java--;
         System.out.println("Book Issued");
         System.out.println("Stock: " + java);
      }

   }

   if(s.equalsIgnoreCase("sql"))
   {
      if(sql==0)
      {
         System.out.println("No More Books");
      }
      else
      {
         sql--;
         System.out.println("Book Issued");
         System.out.println("Stock: " + sql);
      }
   }

   if(s.equalsIgnoreCase("cg"))
   {
      if(cg==0)
      {
         System.out.println("No More Books");
      }
      else
      {
```

```
            cg--;
            System.out.println("Book Issued");
            System.out.println("Stock: " + cg);
        }
      }
   }
}

public class FinalExamQ8 {

   public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);

      LibTransitions lt = new LibTransitions();

      int ch=1;
      while(ch!=0)
      {
         System.out.println("Select Number :");
         System.out.println("0.Exit ");
         System.out.println("1.Submit ");
         System.out.println("2.Issue ");

         ch=sc.nextInt();
         switch(ch)
         {
            case 0:
               break;
            case 1:
               lt.submit();
               break;
            case 2:
               lt.issue();
               break;
            default:
               System.out.println("Please Enter Valid Number !!");
         }
      }
   }

   }
```

OUTPUT:

Output - **Final Exam Q8 (run) #2**

```
run:
Select Number :
0.Exit
1.Submit
2.Issue

1
Which Book You want to Submit? Java/SQL/CG ?
Java
Invalid Submit
Select Number :
0.Exit
1.Submit
2.Issue
```

```
run:
Select Number :
0.Exit
1.Submit
2.Issue

2
Which Book You want to Issue? Java/SQL/CG ?
Java
Book Issued
Stock: 99
Select Number :
0.Exit
1.Submit
2.Issue

2
Which Book You want to Issue? Java/SQL/CG ?
Java
Book Issued
Stock: 98
Select Number :
0.Exit
1.Submit
2.Issue

2
Which Book You want to Issue? Java/SQL/CG ?
Java
Book Issued
Stock: 97
Select Number :
0.Exit
1.Submit
2.Issue
1
Which Book You want to Submit? Java/SQL/CG ?
Java
Book Submitted
Stock: 98
Select Number :
0.Exit
1.Submit
2.Issue
```

Q.9 Write a program that will define a superclass Shape that defines dim1 and dim2 and a method Area() for 2 dimensional shapes. Define rectangle, circle and triangle subclasses with overridden method Area() to calculate area of itself.

CODE:

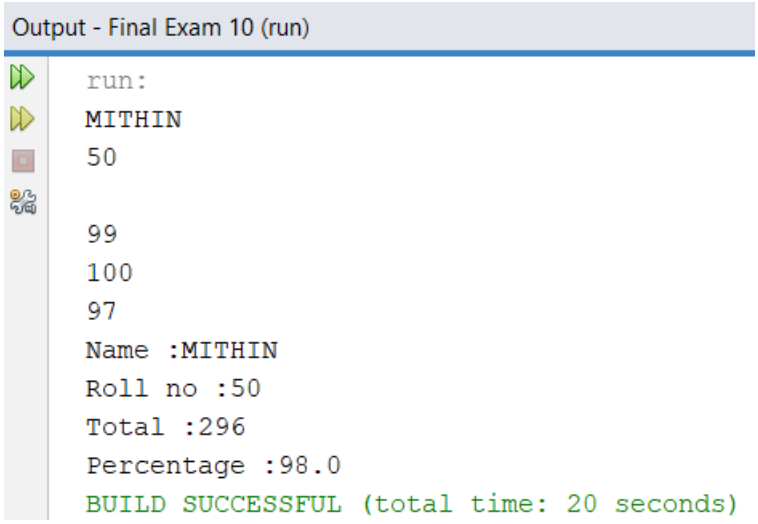
Q.10 Write a program that defines a class Student with name, rollno, marks1, marks2, marks3, total and percentage variables. Define constructor to initialize variables and a method to calculate total as well as percentage. Define a separate method displayData() to display all values.

CODE:

```
package pkgfinal.exam.pkg10;
import java.util.Scanner;
class calculate
{
   String Sname;
   int no, mark1,mark2,mark3,total;
   double percentage;
   calculate(String name,int rollno,int marks1,int marks2,int marks3)
   {
      Sname=name;
      no=rollno;
      mark1=marks1;
      mark2=marks2;
      mark3=marks3;
   }
   void cal()
   {
       total=mark1+mark2+mark3;
       percentage=(mark1+mark2+mark3)/3;
   }
   void display()
   {
       System.out.println("Name :"+Sname);
       System.out.println("Roll no :"+no);
       System.out.println("Total :"+total);
       System.out.println("Percentage :"+percentage);
   }
}
public class FinalExam10 {

   public static void main(String[] args) {
 Scanner sc=new Scanner(System.in);
       String name=sc.next();
       int rollno=sc.nextInt();
```

```
        int marks1=sc.nextInt();
        int marks2=sc.nextInt();
        int marks3=sc.nextInt();

        calculate c=new calculate(name,rollno,marks1,marks2,marks3);
        c.cal();
        c.display();

    }
    }
```

OUTPUT:



Q.11 Define a class series with overloaded methods to print Fibonacci series. This class must contain the following type of method:
a. void fibo() : It will print 10 terms of series
b. void fibo(int terms): It will print total N terms specified by terms parameter.
void fibo(int start, int end) : It will print all the terms between starting and ending numbers.
CODE:

Q.12 TRANSPORT

Code a java program for the Oil and Refinery Compnay which receives transporting of raw materials. Consider there is one abstract class named **RowMatirial** having some instance variables like Goods-value(in Rs.), Service Tax and Surcharge. There are two other subclasses like **ByRoad** and **ByShip**. There are two classes named **Railway** and **Transport** which follow the **ByRaod** class. Both this class have **calculate() and Disp()** methods. There is one class named **Ship** which follow **ByShip** and it also has **calculate() and Disp()** methods. There is a main class named **OilRef** where all these method are called according to user choice. Now you have to calculate Service Tax and surcharges according to Goods Value. For this apply following conditions :

1. **ByRoad :**
   - There is 12% Service Tax and 3% surcharge(on Service tax) on goods value for both railway and transport.
2. **By Ship:**
   - If transporting is out of country than service tax is 20% on goods value and 2% sercharges(on Service tax).
   - If transporting is within country than service tax is 10% on goods value and 2% surcharge(on Service tax).

3 If user enter goods value less than 1 than generate custom exception for it.

CODE:

```
package pkgfinal.exam.q11;

import java.util.Scanner;

abstract class RowMaterial

{

   double goods;

   double st,sr;


}

class ByRoad extends RowMaterial

{


}

class ByShip extends RowMaterial

{
```

```
}
class Railway extends ByRoad
{
   Scanner sc=new Scanner(System.in);
   void calculate()
   {
      System.out.println("Enter Goods :");
      goods = sc.nextInt();
      if(goods <1)
      {
      try
      {
         throw new MyException(" ");
      }
      catch(MyException ex)
      {
        System.out.println("Custom Exception");
      }
   }
      st = (double)(12*goods)/100;
      sr = (double)(3*st)/ 100;
   }
   void display()
   {
      System.out.println("Railway Service Tax: "+st);
```

```
        System.out.println("Railway Surcharge: "+sr);
    }
}
class Transport extends ByRoad
{
    Scanner sc=new Scanner(System.in);
    void calculate()
    {
        System.out.println("Enter Goods :");
        goods =sc.nextInt();
        if(goods <1)
        {
        try
        {
            throw new MyException("Custom Exception ");
        }
        catch(MyException ex)
        {


        }
        }

        st = (double)(12*goods)/100;
        sr = (double)(3*st)/ 100;

    }
```

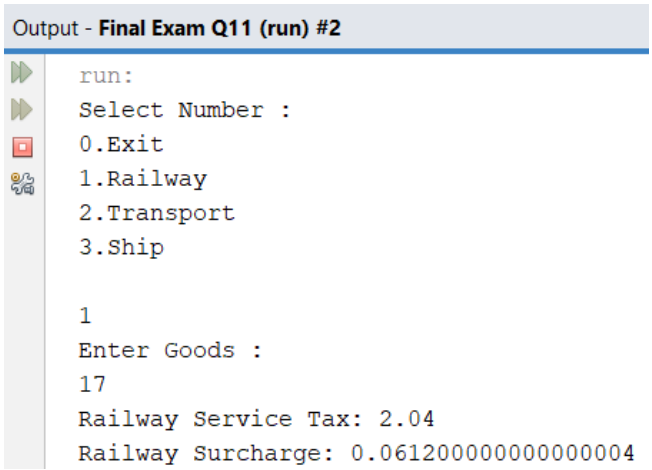
```
    void display()
    {
        System.out.println("Transport Service Tax: "+st);
        System.out.println("Transport Surcharge: "+sr);
    }
}
class Ship extends ByShip
{
    Scanner sc=new Scanner(System.in);
    int c;

    void calculate()
    {
        System.out.println("Enter Goods :");
        goods = sc.nextInt();
        if(goods < 1)
        {
        try
        {
            throw new MyException("Custom Exception ");
        }
        catch(MyException ex)
        {

        }
        }
```

```
        System.out.println("Transporting Within Country?? 0/1");
        c=sc.nextInt();
        if(c==1)
        {
            st = (double)(10*goods)/100;
            sr = (double)(2*st)/ 100;
        }
        else
        {
            st = (double)(20*goods)/100;
            sr = (double)(2*st)/ 100;
        }

    }
    void display()
    {
        System.out.println("Ship Service Tax: "+st);
        System.out.println("Ship Surcharge: "+sr);
    }
}
class MyException extends Exception
{
    public MyException(String s)
    {
        super(s);
    }
```

```
}

public class FinalExamQ11 {

   public static void main(String[] args) {
{
            Scanner sc=new Scanner(System.in);

            Railway r=new Railway();
            Transport t= new Transport();
            Ship s=new Ship();
            int ch=1;
            while(ch!=0)
            {
               System.out.println("Select Number :");
               System.out.println("0.Exit ");
               System.out.println("1.Railway ");
               System.out.println("2.Transport ");
               System.out.println("3.Ship ");
               ch=sc.nextInt();
               switch(ch)
               {
                  case 0:
                     break;
                  case 1:
                     r.calculate();
```

```
                r.display();
                break;
            case 2:
                t.calculate();
                t.display();
                break;
            case 3:
                s.calculate();
                s.display();
                break;
            default:
                System.out.println("Please Enter Valid Number !!");
        }
      }
    }
  }
}
```

OUTPUT:



2……

```
2
Enter Goods :
17
Transport Service Tax: 2.04
Transport Surcharge: 0.061200000000000004
```

3……Within and outside country…

```
3
Enter Goods :
17
Transporting Within Country?? 0/1
0
Ship Service Tax: 3.4
Ship Surcharge: 0.068
```

```
3
Enter Goods :
17
Transporting Within Country?? 0/1
1
Ship Service Tax: 1.7
Ship Surcharge: 0.034
```

Less than 1… (Condition)

```
1
Enter Goods :
0
Custom Exception
```