1.Write a program to read five random numbers and convert it to binary and octal using user defined functions.(random number : 5M binary : 5M Octal : 5M)

```
object Slip2_1
{
        def binary(num:Int)
        {
                var bstr=" ";//binary String
                var rem=0;
                println(num);
                var n1=num;
                while(n1>0)
                {
                rem=n1%2;
                n1=n1/2;
                bstr= rem+bstr;
                }
                println("Binary:"+bstr);
        }
        def octal(num:Int)
        {
                var ostr=" ";//binary String
                var rem=0;
                println();
                println(num);
                var n1=num;
                while(n1>0)
                {
                rem=n1%8;
                n1=n1/8;
                ostr= rem+ostr;
                }
                println("octal:"+ostr);
        }
        def main(args:Array[String])
        {
                val r=new scala.util.Random;
                binary(r.nextInt(15))
                octal(r.nextInt(15))
        }
}
```

2.Write a program to calculate average of all prime numbers between n1 and n2 (take n1 and n2 from user).(accept n1, n2 : 5M prime numbers : 5M average :5M)

```
object Slip2_2
{
        def main(args:Array[String])
        {
                var n1=0;
                var n2=0;
                var count=0;
                var pcount=0;
                var sum=0;
                var prime=" ";
                println("Enter two numbers:");
                n1=scala.io.StdIn.readInt();
                n2=scala.io.StdIn.readInt();
                for(i<-n1 to n2)
                {
                        count=0;
                        for(j<-1 to i )
                        {
                                if(i%j==0)
                                {
                                count=count+1;
                                }
                        }
                        if(count==2)
                        {
                                prime=prime+" "+i;
                                pcount=pcount+1;
                                sum=sum+i;
                        }
                }
                println("prime numbers:"+prime);
                println("average:"+sum/pcount);
        }
}
```

3.Create an abstract class Order (id, description). Derive two classes Purchase Order and Sales Order with details of Supplier and Customer respectively. Create object of each Purchase Order And Sales Order. Display the details of the supplier and customer.

```
abstract class Order()
{
        var orderid:Int=0
        var odescription:String=" ";
}
class PurchaseOrder( var oid:Int,val descrip:String,var sid:Int,var
sname:String,var pno:Long) extends Order()
{
        orderid=oid;
        odescription=descrip;
        def display()
        {
                println("Order Id:"+orderid);
                println("Description:"+odescription);
                println("Supplier Id:"+sid);
                println("Supplier Name:"+sname);
                println("Phone Number:"+pno);
        }
}
class SalesOrder(var oid:Int,val descrip:String,var cid:Int,var
        cname:String,var pno:Long) extends Order()
{
        orderid=oid;
        odescription=descrip;
        def display()
        {
                println("Order Id:"+orderid);
                println("Description:"+odescription);
                println("Customer Id:"+cid);
                println("Customer Name:"+cname);
                println("Phone Number:"+pno);
        }
}
object Slip3
{
        def main(args:Array[String])
        {
                var c1=new SalesOrder(1,"Two
                Laptops",200,"XYZ",233221);
                var s1=new PurchaseOrder(2,"Three
                Computers",101,"ABC",211231);
                println("Purchase Order");
                println("-------------------------------------------");
                c1.display();
                println("Sales Orders");
                println("-------------------------------------------");
                s1.display();    }         }
```

4.Write a program to calculate transpose of a matrix and check if there sultant matrix is lower triangular or not.(accept : 5 M transpose : 10M check lower triangular:10M display:5M)

```
object Slip4
{
        def main(args:Array[String])
        {
                var mat=Array.ofDim[Int](3,3);
                var rmat=Array.ofDim[Int](3,3);
                var isLower:Boolean=true;
                println("Enter Matrix");
                for(i<-0 to 2)
                {
                        for(j<-0 to 2)
                        {
                                mat(i)(j)=scala.io.StdIn.readInt();
                        }
                }
                println("Matrix is:");
                for(i<-0 to 2)
                {
                        for(j<-0 to 2)
                        {
                                print(mat(i)(j)+" ");
                        }
                        println();
                }
                for(i<-0 to 2)
                {
                        for(j<-0 to 2)
                        {
                                rmat(i)(j)=mat(j)(i);
                        }
                }
                println("Transepose of Matrix is:");
                for(i<-0 to 2)
                {
                        for(j<-0 to 2)
                        {
                                print(rmat(i)(j)+" ");
                        }
                        println();
                }
                for(i<-0 to 2)
                {
                        for(j<-0 to 2)
                        {
                                if(i<j)
                                {
                                        if(rmat(i)(j)!=0)
```

```
                                isLower=false;
                            }
                    }
            }
            if(isLower==true)
                    println("Is Lower Triangular");
            else
            println("Is not Lower Triangular");
        }
    }
```

5.Write a program to create two sets of strings and find common strings between them. Merge sets after removing common strings. Display resultant set.(create sets:10M find common elements:5M merge removing common :1

```
object Slip5
{
        def main(args:Array[String])
        {
                var str1:Set[String]=Set("Hello","good","Morning");
                var str2:Set[String]=Set("Hello","good","night");
                var str3=str1.diff(str2);
                println(str1);
                println(str2);
                println(str3);
                var str4=str2.diff(str1);
                println(str4);
                str3++=str4;
                println(str3)
        }
}
```

6.Write a program to read a character and a string from user and remove first and last occurrence of the character from the string. Display resultant string after reversing its case.

```scala
object Slip6
{
        def reverseString(ch:Char):Char=
        {
                if(ch.isLower)
                ch.toUpper;
                else
                ch.toLower;
        }
        def main(args:Array[String])
        {
                var ch=' ';
                var str=" ";
                println("Enter String:");
                str=scala.io.StdIn.readLine();
                var str1=new StringBuilder(str);
                println("Enter character:");
                ch=scala.io.StdIn.readChar();
                str1.deleteCharAt(str1.indexOf(ch.toString()));
                var str3=str1.deleteCharAt(str1.lastIndexOf(ch.toString())).toString;
                var str4=str3.map(reverseString)
                println(str4);
        }
}
```

9.Create a MAP for storing the following information about 5 students, where each Student is described with Name and percentage. Display Student information with highest percentage.

```
class Student(var rno:Int,var sname:String,var sub1:Int,var sub2:Int)
{
        var ptage:Float=(sub1+sub2)/2;
        def display()
        {
                println("Roll No:"+rno);
                println("Name:"+sname);
                println("Percentage:"+ptage);
        }
}
object Slip9
{
        def main(args:Array[String])
        {
                val s1=new Student(1,"Akshay Borse",80,70);
                val s2=new Student(2,"Sumit Amritkar",75,85);
                val s3=new Student(3,"Vishnu Khatale",77,87);
                val s4=new Student(4,"Aniket Borse",89,99);
                val s5=new Student(5,"Tushar Amrutkar",84,87);
                val m1:Map[Int,Student]=Map(1->s1,2->s2,3->s3,4->s4,5->s5);
                var max=m1(1).ptage;
                for((k,v)<-m1)
                {
                        if(m1(k).ptage>max)
                        max=m1(k).ptage;
                }
                for((k,v)<-m1)
                {
                        if(m1(k).ptage==max)
                        m1(k).display()
                }
        }
}
```

1. Write a program to create a MAP with empname and deptname. Print details of all employees working in the same department, as "Mr. Joshi".

```
class Employee(var ename:String,var dept:String)
{
        def display()
        {
                println("--------------------------");
                println("Name:"+ename);
                println("Department Name:"+dept)
        }
}
object Slip1
{
        def main(args:Array[String])
        {
                val e1=new Employee("Vishnu","finance");
                val e2=new Employee("Sumit","finance");
                val e3= new Employee("Paresh","Marketing");
                val e4 =new Employee("Tushar","Marketing");
                var e5=new Employee("Akshay","Marketing");
                var m1:Map[Int,Employee]=Map(1->e1,2->e2,3->e3,4->e4,5->e5);
                for((k,v)<-m1)
                {
                        if(v.dept.equalsIgnoreCase("marketing"))
                        v.display()
                }
        }
}
```

10.Create abstract class Shape with abstract functions volume() and display().Extend two classes Cube and Cylinder from it. Create object of Cube and Cylinder, Calculate volume of each and display it.

```
abstract class Shape
{
        def volume():Double;
        def display();
}
class Cylinder(var r:Int,var h:Int) extends Shape
{
        def volume():Double=
        {
                return 3.14*r*r*h;
        }
        def display()
        {
                println("Volume Cylinder :"+volume());
        }
}
class Cube(var s:Int) extends Shape
{
        def volume():Double=
        {
                return s*s*s;
        }
        def display()
        {
                println("Volume of cube:"+volume());
        }
}
object Slip10
{
        def main(args:Array[String])
        {
                val cyl=new Cylinder(1,1);
                cyl.display();
                val cub=new Cube(3);
                cub.display();
        }
}
```

12.Write a program for multiplication of two matrices. Find determinant ofresultant matrix.

```scala
object Slip12
{
        def main(args:Array[String])
        {
                val arr1=Array.ofDim[Int](2,2);//1st array
                val arr2=Array.ofDim[Int](2,2);//2nd array
                var rarry=Array.ofDim[Int](2,2)//resultant Array
                println("Enter Matrix1");
                for(i<-0 to 1)
                {
                        for(j<-0 to 1)
                        {
                                arr1(i)(j)=scala.io.StdIn.readInt();//read Array1 element
                        }
                }
                println("Enter Matrix2");
                for(i<-0 to 1)
                {
                        for(j<-0 to 1)
                        {
                                arr2(i)(j)=scala.io.StdIn.readInt();//read Array2 element
                        }
                }
                println("MATRIX -1");
                for(i<-0 to 1)
                {
                        for(j<-0 to 1)
                        {
                        print(arr1(i)(j)+" ");//print Array
                        Element
                        }
                println();
                }
                println("MATRIX -2");
                for(i<-0 to 1)
                {
                        for(j<-0 to 1)
                        {
                        print(arr2(i)(j)+" ");//print Array
                        Element
                        }
                println();
                }
```

```
for(i<-0 to 1)
{
        for(j<-0 to 1)
        {
                rarry(i)(j)=0;
                for(k<-0 to 1)
                rarry(i)(j)=rarry(i)(j)+arr1(i)(k)*arr2(k)(j);//multiplication
        }
}
println("RESULTANT MATRIX");
for(i<-0 to 1)
{
        for(j<-0 to 1)
        {
        print(rarry(i)(j)+" ");//print Array
        Element
        }
        println();
}
var det=(rarry(0)(0)*rarry(1)(1))-
(rarry(0)(1)*rarry(1)(0));
println("Determinant:"+det);
}
}
```

2) And **3**)=>
>db.student1.insert({name:"Abhi",course:[{coursename:"bcs"},{coursename:"bvoc"}],marks:80,age: 21,gender:"male",city:"pune"})

>db.student1.insert({name:"mukesh",course:[{coursename:"bcs"},{coursename:"bvoc"}],marks:60,age:22,gender:"male",city:"pune"})

>db.student1.insert({name:"manisha",course:[{coursename:"mcs"},{coursename:"bvoc"}],marks:90, age:22,gender:"female",city:"mumbai"})

>db.student1.insert({name:"manasi",course:[{coursename:"mcs"},{coursename:"bvoc"}],marks:92,age:22,gender:"female",city:"latur"})

>db.student1.insert({name:"apurva",course:[{coursename:"mcs"},{coursename:"bvoc"}],marks:37,age:22,gender:"female",city:"sasvad"})

>db.student1.insert({name:"arati",course:[{coursename:"mcs"},{coursename:"bvoc"}],marks:32,age: 22,gender:"female",city:"bekarai"})

4)
a) > db.student1.count({marks:{$gt:80}})

b) > db.student1.find({marks:{$lt:40}})

c) > var my=db.student.find({marks:{$gt:70}});

> while(my.hasNext()){print(tojson(my.next()));}

d)>db.student1.find({gender:"female",$or:[{city:"pune"},{city:"mumbai"},{marks:{$lt:50}}]})

Slip 2

2)
> db.product.insert({name:"robot",price:12000})

> db.product.insert({name:"toycar",price:2000})

> db.product.insert({name:"cricketset",price:9000})

> db.product.insert({name:"studymaterial",price:19000})

3)
>db.order.insert({orderno:3736,custName:"arunkumar",product:{productName:"toycar",price:20000},order_date:"12/2/2019",stetus:"processed",Totalbill:2039,invoice:{invoiceNO:67564,bill:2039,date:"17/2/2019"}})

>db.order.insert({orderno:3737,custName:"arunkumar",product:{productName:"robot",price:12000},order_date:"11/3/2019",stetus:"processed",Totalbill:12800,invoice:{invoiceNO:67574,bill:12039,date:"17/3/2019"}})

>db.order.insert({orderno:3738,custName:"arunkumar",product:{productName:"cricketset",price:9000},order
_date:"15/5/2019",stetus:"in process",Totalbill:9050})

>db.order.insert({orderno:3739,custName:"mukeshpatil",product:{productName:"studentmaterial",price:19000},order_date:"15/8/2019",stetus:"in process",Totalbill:19080})

4)
a)> db.product.find().pretty()

b) > db.order.find({Totalbill:{$lt:10000}})

c) > db.order.find({stetus:"in process"})

d) >db.order.find({custName:"arunkumar",stetus:"processed"})

slip 3

2)

> db.book.insert({BName:"shyamchiaai",cost:700,author:"sane guruji",published:2007})

> db.book.insert({BName:"Two Saints",cost:1700,author:"raguramkrishna",published:2017})

> db.book.insert({BName:"ramkrushna paramhans",cost:800,author:"raguramkrishna",published:2017})

>db.book.insert({BName:"DMS",cost:300,author:"raguramkrishna",published:2005})

3)

> db.publisher.insert({pname:"OReilly",language:"English",books:[{BName:"ramkrushna paramhans"},{BName:"Two Saints"}],city:"mumbai"})

>db.publisher.insert({pname:"vision",language:"English",books:[{BName:"DMS"}],city:"pune"})

> db.publisher.insert({pname:"OReilly",language:"marathi",books:[{BName:"shyamchi aai"}],city:"mumbai"})

4)

a)> db.publisher.find({city:"mumbai"})

b)> db.book.find({cost:{$lt:1000}})

c) > db.book.find({author:"raguramkrishna",published:2017})

d)> > db.publisher.find({pname:"OReilly",$or:[{language:"English"},{language:"marathi"}]})

slip 4

2)

```
>db.Hospital.insert({Hno:1,Hname:"AAA",Specialization:["Pediatric","Gynaec","Orthopaedic"],Peopl
e:[{Pname:"PQR",Rating:4},{Pname:"SDE",Rating:5}],Doctor:[{"Dname" : "WWW",
"Visit" : "Sunday"}]})
```

```
>db.Hospital.insert({Hno:2,Hname:"BBB",Specialization:["Gynaec","Orthopaedic"],People:[{Pname:"
POP",Rating:2},{Pname:"SDE",Rating:3}],Doctor:[{"Dname":"XXX",Visit:"Monday"}]})
```

```
>db.Hospital.insert({Hno:3,Hname:"CCC",Specialization:["Gynaec","Orthopaedic","Pediatric"],People
:[{Pname:"KLO",Rating:3},{Pname:"LPO",Rating:3}],Doctor:[{"Dname" : "XXX","Visit":"Tuesday"}]})
```

4)

a) > db.Hospital.find({Specialization:"Pediatric"})

b)>db.Hospital.find({Hname:"CCC","Doctor.Visit":"Tuesday"})

c)>db.Hospital.find({Specialization:{$not:{$size:1}},"Doctor.Dname":"XXX"})

d) > db.Hospital.find({"People.Rating":{ $gt: 3 },Hname:"AAA"})

slip 5 2) & 3)
>db.post.insert({title:"online",url:"www.abc.com",tag:["food","travel"],pname:"mukesh",pdate:new Date("2019-03-12"),like:89,user:[{name:"abhi",comment:"good",message:"do best", cdate:new Date("2020-03-12"),like:1}]})

>db.post.insert({title:"wetpet",url:"www.wetpet.com",tag:["food","travel",],pname:"Amit",pdate:new Date("2018-03-12"),like:82,user:[{name:"abhi",comment:"good",message:"do best",time:"4pm",like:1},{name:"mukesh",comment:"best",message:"success", cdate:new Date("2008-11-12"),like:2}]})

>db.post.insert({title:"wetpet",url:"www.wetpet.com",tag:["food","travel","magic"],pname:"abhijeet",pdate:newDate("2017-03-12"),like:182,user:[{name:"sagar",comment:"like",message:"dobest",time:"4pm",like:1},{name:"mukesh",comment:"best",message:"success", cdate:new Date("2019-03-12"),like:2}]})

>db.post.insert({title:"nonveg",url:"www.non.com",tag:["food","travel","chiken"],pname:"Amit",pdate:new Date("2019-07-12"),like:82,user:[{name:"manisha",comment:"good",message:"dobest",time:"4pm",like:0},{name:"manasi",comment:"best",message:"success", cdate:new Date("2018-03-12"),like:0}]})

4)
a) >db.post.find({tag:"food"})

b) >db.post.find({pname:"Amit"})

c) > db.post.find({tag:"travel",pdate:{"$lte":new Date("2018-03-11")},"user.name":"sagar","user.comment":"like"})

d) > db.post.find({$or:[{"user.cdate":{$lte:new Date("2019-08-07")}},{"user.like":0}]})

slip6

2)
```
> db.turisum.insert({name:"veenaword",rate:9,package:[{pname:"shillong",cost:10000},{pname:"gujart",cost:7000},{pname:"karnataka",cost:6000}]})

>db.turisum.insert({name:"rohit",rate:7,package:[{pname:"shillong",cost:10000},{pname:"rujan",cost:7000}]})
```

3)
```
>db.tour.insert({sourc:"john",destination:"shillong",toerisumName:"veenaword",tourisumrate:8000,expense:20000,year:2018,customer:[{cname:"mukesh",city:"pune"},{cname:"abhijeet sangita",city:"baramati"},{cname:"manisha",city:"15no"},{cname:"manasi",city:"latur"}]})

>db.tour.insert({sourc:"john",destination:"karnataka",toerisumName:"veena word",tourisumrate:80090,expense:20900,year:2017,customer:[{cname:"mukesh",city:"pune"},{cname:"abhijeetsangita",city:"baramati"},{cname:"manisha",city:"15no"},{cname:"manasi",city:"latur"}]})

>db.tour.insert({sourc:"john",destination:"rajasthan",toerisumName:"rohit",tourisumrate:6000,expense:30400,year:2019,customer:[{cname:"mukesh",city:"pune"},{cname:"abhijeet sangita",city:"baramati"},{cname:"manisha",city:"15no"},{cname:"manasi",city:"latur"}]})

>db.tour.insert({sourc:"john",destination:"taj",toerisumName:"rohit",tourisumrate:60090,expense:10400,year:2016,customer:[{cname:"mukesh",city:"pune"},{cname:"abhijeetsangita",city:"baramati"},{cname:"manisha",city:"15no"},{cname:"manasi",city:"latur"}]})
```

4)
a) >db.turisum.find({name:"veena word"}).pretty()

b) >db.turisum.find({}).sort({"rate":-1}).limit(1)

c)>db.tour.aggregate([{"$sort":{"year":1}},{"$limit":3},{$group:{_id:null,"count":{"$sum":"$expense"}}}])

d) > db.tour.find({destination:"shillong"})

slip 7

2) & **3**)

> db.scien.insert({fname:"mukesh",lname:"navse",BOD:newDate("1952-04-18"),DOD:"still alive",field:["tcs","java","c","sql"],award:[{name:"turingmachine",year:1976},{name:"robotic",year:1998},{name:"codetalent",year:1995}]})

> db.scien.insert({fname:"abhi",lname:"nalave",BOD:newDate("1972-04-18"),DOD:"still alive",field:["tcs","java","sql"],award:[{name:"codemaster",year:1976},{name:"robot",year:1998},{name:"puzzletalent",year:1995}]})

>db.scien.insert({fname:"manisha",lname:"hipparkar",BOD:new Date("1942-04-18"),DOD:new Date("2009-08-06"),field:["tcs","java"],award:[{name:"topper",year:1976},{name:"puraskar",year:1998},{name:"puzzletalent",year:1995}]})

4)
a) > db.scien.find({ lname: { $regex: /n/ } })

b) > db.scien.find({BOD:{"$gt":new Date("1950-03-11")},DOD:"still alive"})

c)>db.scien.aggregate([{$group:{_id:{year:"$award.year",Name:"$award.name"}}}])

d) > db.scien.find({"award.name":"turingmachine","award.year":{$lt:1980},field:{$size:4}})

slip8
2) & **3**)
```
>db.item.insert({itemName:"planner",tag:["wash","food","vehicle"],status:"A",height:5,width:9,inst
ack:15,warehouse:[{location:"pune",quntity:36},{location:"mumbai",quntity:67}]})

>db.item.insert({itemName:"toycar",tag:["food","vehicle"],status:"D",height:5,width:9,instack:15,wa
rehouse:[{location:"pune",quntity:36},{location:"mumbai",quntity:67}]})

> db.item.insert({itemName:"roboticcar",tag:["food","vehicle"],status:"A",height:9,width:9,instack
:5,warehouse:[{location:"pune",quntity:26},{location:"mumbai",quntity:17}]})

>db.item.insert({itemName:"bag",tag:["food","vehicle","school","travel"],status:"c",height:19,width:
39,instack:75,warehouse:[{location:"surat",quntity:26},{location:"lanavala",quntity:17}]})
```

4)
a) > db.item.find({status:"D","warehouse.quntity":{$gt:30}})

b) > db.item.find({"tag":{$size:3}})

c) >db.item.find({$or:[{status:"A"},{"warehouse.quntity":{$lt:30},height:{$gt:10}}]})

d) > db.item.find({itemName:"planner",instack:{$lt:20}})

slip 9
2) & **3**)
>db.transaction.insert({itemName:"toy",customerName:"john",paymentmode:"debitcard",payment:8000})

>db.transaction.insert({itemName:"car",customerName:"john",paymentmode:"creditcard",payment:4000})

>db.transaction.insert({itemName:"bag",customerName:"mukesh",paymentmode:"cash",payment:5000})

>db.transaction.insert({itemName:"airlineticket",customerName:"rohit",paymentmode:"cash",payment:50090})

>db.transaction.insert({itemName:"mango",customerName:"abhijeet",paymentmode:"creditcard",payment:8000})

>db.transaction.insert({itemName:"bus",customerName:"manasi",paymentmode:"debitcard",payment:7000})

4)
a) > db.transaction.find({customerName:"john"})

b) > db.transaction.find({paymentmode:"debitcard"})

c)>db.transaction.aggregate([{$match:{"paymentmode":"creditcard"}},{$group:{_id:null,"count":{"$sum":"$payment"}}}])

d) >db.transaction.aggregate([{$group:{_id:"$paymentmode","count":{"$sum":"$payment"}}}])

slip10

2)
>db.custome.insert({cname:"mukesh",modelname:"samsungj6",amount:20000})

>db.custome.insert({cname:"abhijeet",modelname:"samsungj6",amount:20060})

> db.custome.insert({cname:"manasi",modelname:"iphone7+",amount:30060})

> db.custome.insert({cname:"manisha",modelname:"iphone7+",amount:30070})

> db.custome.insert({cname:"dipak",modelname:"iphone7+",amount:30800})

3)
>db.shopping.insert({brandname:"samsung",rate:6,model:[{mname:"s40",ram:"3GB",rom:"32GB",rate:4},{mname:"j6",ram:"4GB",rom:"32GB",rate:7},{mname:"j7",ram:"6GB",rom:"64GB",rate:6}]})

>db.shopping.insert({brandname:"vivo",rate:8,model:[{mname:"Y55",ram:"3GB",rom:"32GB",rate:6},{mname:"Ys5",ram:"4GB",rom:"32GB",rate:4},{mname:"YYY",ram:"6GB",rom:"64GB",rate:6}]})

4)
a) >db.shopping.find({"model.ram":"3GB","model.rom":"32GB"})

b) > db.custome.find({modelname:"samsung j6"})

c) > db.shopping.aggregate([{"$sort":{"rate":-1}},{"$limit":1},{$group:{_id:"$brandname"}}])

d) > db.custome.find().sort({ "cname": 1 })