**SCALA PRACTICALS 2020-2021**

**SLIP SOLUTIONS**

****

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **SLIP** | **PAGE NUMBER** |
| **1** | **SLIP 1** | **2** |
| **2** | **SLIP 2** | **5** |
| **3** | **SLIP 3** | **7** |
| **4** | **SLIP 4** | **10** |
| **5** | **SLIP 5** | **13** |
| **6** | **SLIP 6** | **15** |
| **7** | **SLIP 7** | **16** |
| **8** | **SLIP 8** | **17** |
| **9** | **SLIP 9** | **19** |
| **10** | **SLIP 10** | **22** |
| **11** | **SLIP 11** | **24** |
| **12** | **SLIP 12** | **29** |
| **13** | **SLIP 13** | **30** |
| **14** | **SLIP 14** | **32** |
| **15** | **SLIP 15** | **33** |
| **17** | **SLIP 17** | **37** |
| **18** | **SLIP 18** | **39** |
| **19** | **SLIP 19** | **40** |
| **20** | **SLIP 20** | **41** |

**SLIP 1**

1. Write a program for multiplication of two matrices. Find determinant of resultant matrix. (Accept matrix: 5M Multiplication : 10M Determinant : 10M Display : 5M)

**CODE :**

object Slip1

{

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);

}

}

**OUTPUT :**

Enter Matrix1

1

2

3

4

Enter Matrix2

5

6

3

1

MATRIX -1

1 2

3 4

MATRIX -2

5 6

3 1

RESULTANT MATRIX

11 8

27 22

Determinant:26

**SLIP 2**

1. Write a program to read two strings. Find the occurrence of second string in the first string. Reverse the case of each occurrence in the string and display resultant string.(Read : 5M find Occurance:10M Reverse : 10M Display: 5M)

**CODE :**

object Slip2

{

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();

var str4=str1.map(reverseString)

println(str4);

}

}

**OUTPUT :**

My name is Milo

Enter character:

M

ArrayBuffer(m, Y, , N, A, M, E, , I, S, , m, I, L, O)

**SLIP 3**

1. **A.** 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)

**B.** 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)

**A : CODE :**

object Slip3

{

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))

}

}

**A : OUTPUT :**

8

Binary:1000

10

Octal:12

**B : CODE :**

object Slip3B

{

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);

}

}

**B : OUTPUT :**

Enter two numbers:

34

56

prime numbers: 37 41 43 47 53

average:44

**SLIP 4**

1. Create an abstract class Order (id, description). Derive two classes PurchaseOrder and SalesOrder with details of Supplier and Customer. Create object of each PurchaseOrder And SalesOrder. Display the details of all supplier and customers.

(abstract class : 5M derivation : 20 M display : 10M)

**CODE :**

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 Slip4

{

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();

}

}

**OUTPUT :**

Purchase Order

--------------------------------------------

Order Id:1

Description:Two Laptops

Customer Id:200

Customer Name:XYZ

Phone Number:233221

Sales Orders

--------------------------------------------

Order Id:2

Description:Three Computers

Supplier Id:101

Supplier Name:ABC

Phone Number:211231

**SLIP 5**

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

(create MAP : 10M logic: 10M print :5M)

**CODE :**

class Employee(var eno:String,var ename:String,var dept:String)

{

def display()

{

println("---------------------------");

println("Eno:"+eno);

println("Name:"+ename);

println("Department Name:"+dept)

}

}

object Slip5

{

def main(args:Array[String])

{

val e1=new Employee("101","Vishnu","finance");

val e2=new Employee("104","Sumit","finance");

val e3= new Employee("102","Paresh","Marketing");

val e4 =new Employee("134","Tushar","Marketing");

var e5=new Employee("157","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()

}

}

}

**OUTPUT :**

---------------------------

Eno:157

Name:Akshay

Department Name:Marketing

---------------------------

Eno:102

Name:Paresh

Department Name:Marketing

---------------------------

Eno:134

Name:Tushar

Department Name:Marketing

**SLIP 6**

1. Write a program to merge two sets of integers and calculate sum of all integers in the merged set. Also display largest and smallest element from merged set. (Merge : 10M sum :5M largest :5M smallest :5 M display : 5M)

**CODE :**

import scala.collection.mutable.Set

object Slip6

{

def main(args:Array[String])

{

var s1=Set(1,2,3,4,5,6);

var s2=Set(4,5,6,7,8);

s1++=s2;

println(s1);

println("Sum:"+s1.sum);

println("Maximum:"+s1.max);

println("Minimum:"+s1.min);}

}

**OUTPUT :**

HashSet(1, 2, 3, 4, 5, 6, 7, 8)

Sum:36

Maximum:8

Minimum:1

**SLIP 7**

1. Create lists using five different methods and display each of them.(List style, java style, fill, range, tabulate methods)

(each method : 5 M display : 1M each)

**CODE :**

object Slip7

{

def main (args:Array[String])

{

val num=1::2::3::Nil //Lisp Style

val num1=List(11,22,33); //Java Style

val num2=List.range(10,50) //Range Style

val num3 = List.tabulate(5)(n=>n\*n) //Tabulate Style

println(num)

println(num1)

println(num2)

println(num3)

}

}

**OUTPUT :**

List(1, 2, 3)

List(11, 22, 33)

List(10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49)

List(0, 1, 4, 9, 16)

**SLIP 8**

1. Write a program to calculate sum of all perfect numbers between 1 and 100. Display perfect numbers also.

(find Perfect number : 10 M sum : 10M display numbers and sum: 10M)

**CODE :**

object Slip8

{

def main(args:Array[String])

{

var sum=0;

var psum=0;

var perfect=" ";

for(i<-1 to 100)

{

for(j<-1 to i-1)

{

if(i%j==0)

{

sum=sum+j;

}

}

if(sum==i)

{

psum=psum+i;//sum of perfect number;

perfect=perfect+" "+i;

}

sum=0;

}

println("Perfect Numbers:"+perfect);

println("Sum of Perfect Number:"+psum);

}

}

**OUTPUT :**

Perfect Numbers: 6 28

Sum of Perfect Number:34

**SLIP 9**

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

**CODE :**

object Slip9

{

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("Transpose 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");

}

}

**OUTPUT :**

Enter Matrix

10

4

6

2

13

45

3

7

8

Matrix is:

10 4 6

2 13 45

3 7 8

Transpose of Matrix is:

10 2 3

4 13 7

6 45 8

Is not Lower Triangular

**SLIP 10**

1. 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.

(read : 10 M remove occurance:10M reverse : 5M display :5M)

**CODE :**

object Slip10

{

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);

}

}

**OUTPUT :**

Enter String:

Scala is a good language.

Enter character:

l

sCAA IS A GOOD ANGUAGE.

**SLIP 11**

1. Design an abstract class Employee with computeSal() as an abstract function. Create two subclasses Worker and Manager. Salary of worker should be calculated on hourly basis of work and Salary of Manager should be calculated on monthly basis with additional incentives. Create five objects each of Worker and Manager class, and display their details.

(abstract class and function :10M subclasses:5M salary calculation: 5 M each display:5M)

**CODE :**

abstract class Employees

{

def computeSalary():Int;

}

class manager(var mno:Int,var mname:String,var month:Int) extends Employees

{

def computeSalary():Int=

{

var salary=month\*30000;

return salary;

}

def display()

{

println("---------------------------------------------------------------------");

println("Manager No:"+mno);

println("Manager Name:"+mname);

println("Salary:"+computeSalary());

}

}

class worker(var wno:Int,var wname:String,var hr:Int) extends Employees

{

def computeSalary():Int=

{

var salary=hr\*300;

return salary;

}

def display()

{

println("-------------------------------------------------------------------------");

println("Worker No:"+wno);

println("Worker Name:"+wname);

println("Salary:"+computeSalary());

}

}

object Slip11

{

def main(args:Array[String])

{var manager1=new manager(1,"Akshay Borse",4);

var manager2=new manager(2,"Sumit Amritkar",3);

var manager3=new manager(3," Vishnu Khatale",2);

var manager4=new manager(4,"Roshan Adke",5);

var manager5=new manager(5,"Tushar Amrutkar",9);

var worker1=new worker(1,"Ganesh Darade",12);

var worker2=new worker(2,"Viraj Gadekar",16);

var worker3=new worker(3," Abhi Chavhan",10);

var worker4=new worker(4,"Kalpesh Deshmukh",5);

var worker5=new worker(5,"Abhijit Rakibe",4);

println("-----------------------MANAGER--------------------------------------------");

manager1.display();

manager2.display();

manager3.display();

manager4.display();

manager5.display();

println("-----------------------WORKER--------------------------------------------");

worker1.display();

worker2.display();

worker3.display();

worker4.display();

worker5.display();

}

}

**OUTPUT :**

-----------------------MANAGER--------------------------------------------

---------------------------------------------------------------------

Manager No:1

Manager Name:Akshay Borse

Salary:120000

---------------------------------------------------------------------

Manager No:2

Manager Name:Sumit Amritkar

Salary:90000

---------------------------------------------------------------------

Manager No:3

Manager Name: Vishnu Khatale

Salary:60000

---------------------------------------------------------------------

Manager No:4

Manager Name:Roshan Adke

Salary:150000

---------------------------------------------------------------------

Manager No:5

Manager Name:Tushar Amrutkar

Salary:270000

-----------------------WORKER--------------------------------------------

-------------------------------------------------------------------------

Worker No:1

Worker Name:Ganesh Darade

Salary:3600

-------------------------------------------------------------------------

Worker No:2

Worker Name:Viraj Gadekar

Salary:4800

-------------------------------------------------------------------------

Worker No:3

Worker Name: Abhi Chavhan

Salary:3000

-------------------------------------------------------------------------

Worker No:4

Worker Name:Kalpesh Deshmukh

Salary:1500

-------------------------------------------------------------------------

Worker No:5

Worker Name:Abhijit Rakibe

Salary:1200

**SLIP 12**

1. Write a program to create a list of 1 to 100 numbers. Create second list from first list selecting numbers which are perfect square. Display it. (create first list: 10M find perfect square : 10M create second list :5 M display : 5M)

**CODE :**

import scala.collection.mutable.ListBuffer

object Slip12

{

def main(args:Array[String])

{

val l1=List.range(1,101);

var l2:ListBuffer[Int]=ListBuffer();

for(i<-l1)

{

for(j<-1 to i)

{

if(i==j\*j)

l2+=i;

}

}

println("Perfect Numbers:"+l2);}

}

**OUTPUT :**

Perfect Numbers:ListBuffer(1, 4, 9, 16, 25, 36, 49, 64, 81, 100)

**SLIP 13**

1. Write user defined functions to reverse the case of a given string and call the function using MAP. (user defined function :10 M reverse case logic : 10M call using MAP : 10M)

**CODE :**

object Slip13

{

def reverse(ch:Char):Char=

{

if(ch.isLower)

ch.toUpper;

else

ch.toLower;

}

def main(args:Array[String])

{

var str=" ";

println("Enter String:");

str=scala.io.StdIn.readLine();

var str2=str.map(reverse);

println(str2);

}

}

**OUTPUT :**

Enter String:

I love programming in Scala.

i LOVE PROGRAMMING IN sCALA.

**SLIP 14**

1. 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 :10M display:5M)

**CODE :**

object Slip14

{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)

}

}

**OUTPUT :**

Set(Hello, good, Morning)

Set(Hello, good, night)

Set(Morning)

Set(night)

Set(Morning, night)

**SLIP 15**

1. Define a class SavingAccount (accNo, name, balance, minBalance). Define appropriate constructors and operations withdraw(), deposit(), viewBalance(). Create an array of SavingAccount objects and perform operations and display them. (define class:5M constructors: 10M operations: 5 M each)

**CODE :**

class SavingAccount(var acno:Int,var name:String,var balance:Int,var

minbalance:Int)

{

def withdraw()

{

println("Enter Amount:");

var n1=scala.io.StdIn.readInt();

balance=balance-n1;

if(balance<minbalance)

{

println("TRANSACTION FAILED:");

balance=balance+n1;

}

else

println("TRANSACTION SUCCESSFULL");

}

def deposite()

{

println("Enter Amount:");var n1=scala.io.StdIn.readInt();

balance=balance+n1;

}

def viewbalance()

{

println("Account Number:"+acno);

println("Name:"+name);

println("Balance:"+balance);

println("Minimum Balance:"+minbalance);

}

}

object Slip15

{

def main(args:Array[String])

{

val s1=new Array[SavingAccount](5)

var ch=0;

s1(0)=new SavingAccount(1,"Akshay Borse",20000,10000);

s1(1)=new SavingAccount(2,"Sumit Amritkar",30000,15000);

s1(2)=new SavingAccount(3,"Vishnu Khatale",40000,6000);

s1(3)=new SavingAccount(4,"Ganesh Darade",50000,3000);

s1(4)=new SavingAccount(5,"Tushar Amrutkar",55000,10000);

println("Enter Account Number:");

var ac=scala.io.StdIn.readInt();

for(i<-0 to 4)

{

if(s1(i).acno==ac)

{

println("Account number Exsists");

println("1.Cash Withdraw:");

println("2.Cash Deposite:");

println("3.View Balance:");

println("4.Exit");

while(ch!=5)

{

println("Enter Your Choice:");

var

ch=scala.io.StdIn.readInt();

ch match

{

case

1=>s1(i).withdraw();

case

2=>s1(i).deposite();

case

3=>s1(i).viewbalance();

case

4=>System.exit(1);

}

}

}}

println()

}

}

**OUTPUT :**

Enter Account Number:

1

Account number Exsists

1.Cash Withdraw:

2.Cash Deposite:

3.View Balance:

4.Exit

Enter Your Choice:

1

Enter Amount:

10000

TRANSACTION SUCCESSFULL

Enter Your Choice:

2

Enter Amount:

50000

Enter Your Choice:

3

Account Number:1

Name:Akshay Borse

Balance:60000

Minimum Balance:10000

Enter Your Choice:

4

**SLIP 17**

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

(create abstract class :5M extend classes : 5M functions: 5M each display: 10M)

**CODE :**

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 Slip17

{

def main(args:Array[String])

{

val cyl=new Cylinder(1,1);

cyl.display();val cub=new Cube(3);

cub.display();

}

}

**OUTPUT :**

Volume Cylinder :3.14

Volume of cube:27.0

**SLIP 18**

1. Create array of strings and read a new string from user. Display all the strings from the array that contain the new string. (accept array and string: 10M check string logic : 10M display 10M)

**CODE :**

object Slip18

{

def main(args:Array[String])

{

var str:Array[String]=Array("Hello Good Morning","Hello Good Night","Hello Good Afternoon");

var str1=" ";

println("Enter string:");

str1=scala.io.StdIn.readLine();

for(j<-0 to str.length-1)

{

if(str(j).contains(str1))

println(str(j))

}

}

}

**OUTPUT :**

Enter string:

Night

Hello Good Night

**SLIP 19**

1. Create a list of 10 random numbers. Create another list from members of first list using function 3n2+4n+6. Display second list in ascending order. (random numbers list: 10M create list with function: 10M display with ascending order : 10M)

**CODE :**

import scala.util.\_

object Slip19 {

def main(args: Array[String]) {

var l1: List[Int] = List();

var l2: List[Int] = List();

var n1 = 0;

for (i <- 1 to 10) {

l1 ::= Random.nextInt(10);

}

println("List1:" + l1.sorted);

for (j <- l1) {

n1 = 3 \* j \* j + 4 \* j + 6;

l2 ::= n1;

}

println("list2:" + l2.sorted)

}

}

**OUTPUT :**

List1:List(1, 2, 3, 4, 5, 5, 6, 8, 8, 9)

list2:List(13, 26, 45, 70, 101, 101, 138, 230, 230, 285)

**SLIP 20**

1. Create a MAP for storing the following information about 5 students, where each Student is described with Rollno, Name and marks in two subjects. Display Student information with highest percentage. (create MAP : 10M select with highest logic: 10M display : 10M)

**CODE :**

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 Slip20

{

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()

}

}

}

**OUTPUT :**

Roll No:4

Name:Aniket Borse

Percentage:94.0