**Scala Programs**

**Assignment No.1 (Control Structure)**

**1] Calculate Average of all Numbers between n1 and n2**

import scala.io.StdIn.readInt

object avg extends App{

var start:Int=0

var end:Int=0

var avg:Float=0

println("Enter the first number")

var n1=readInt()

println("Enter the second number")

var n2=readInt()

if (n1<n2)

{

start=n1

end=n2

}

else{

start=n2

end=n1

}

var total:Float=0

while(start<=end){

total=total+start

start=start+1

println("start= "+start+ "total= "+total)

}

print((n1-n2).abs)

avg=(total/((n1-n2).abs+1))

println("avg of "+n1+"and "+n2+"is "+avg)

}

**2] Write a Program to read five random numbers and check that Random Numbers areperfect or not.**

import scala.io.StdIn.readInt

import scala.util.control.\_

object perfect {

def main(args: Array[String])=

{

for(j <- 1 to 5)

{

var n1=scala.util.Random

var n=n1.nextInt(50)

var sum=0

for(i <-1 to n-1)

{

if(n%i == 0){

sum=sum+i

}

}

if(sum==n)

println(+n+" is a perfect number")

else

println(+n+" is not a perfect number")

}

}

}

**4] Read Integer from user and convert it to Binary and Octal using user difined function.**

import scala.io.StdIn.readInt

import scala.util.control.\_

object avg extends App{

def binarycon(n : Int)=

{

var i=0

var num=n

var a=new Array[Int](10)

while(num>0)

{

a(i)=num%2

i=i+1

num=num/2

}

println("binary equivalent")

for(j<-i-1 to 0 by -1)

print(a(j))

}

def octalcon(n : Int)=

{

var i=0

var num=n

var a=new Array[Int](10)

while(num>0)

{

a(ei)=num%8

i=i+1

num=num/8

}

println("octal equivalent")

for(j<-i-1 to 0 by -1)

print(a(j))

}

def main(args: Array[String])={

println("enter the number")

var n=scala.io.StdIn.readInt

println(n)

octalcon(n)

binarycon(n)

}

}

**Assignment No.2 (Array/Matrix)**

**1] Write a Program to find maximum and minimum of an array .**

object MaxMin {

def main(args: Array[String]): Unit = {

val numbers = Array(10, 5, 8, 2, 15, 7)

val min = findMin(numbers)

val max = findMax(numbers)

println(s"Minimum value: $min")

println(s"Maximum value: $max")

}

def findMin(arr: Array[Int]): Int = {

var min = arr(0)

for (num <- arr) {

if (num < min) {

min = num

}

}

min

}

def findMax(arr: Array[Int]): Int = {

var max = arr(0)

for (num <- arr) {

if (num > max) {

max = num

}

}

max

}

}

**2] Write a Program to calculate ditrminant of a matrix .**

object MatrixDeterminant {

def main(args: Array[String]): Unit = {

val matrix = Array(Array(3, 5), Array(2, 7))

if (matrix.length != 2 || matrix(0).length != 2 || matrix(1).length != 2)

{

println("Matrix is not 2x2, Determinant calculation is not possible.")

}

else

{

val determinant = calculateDeterminant(matrix)

println(s"Determinant of the matrix is: $determinant")

}

}

def calculateDeterminant(matrix: Array[Array[Int]]): Int = {

matrix(0)(0) \* matrix(1)(1) - matrix(0)(1) \* matrix(1)(0)

}

}

**3] Write a Program to check if the matrix is upper triangular or not.**

object UpperTriangularMatrix {

def main(args: Array[String]): Unit = {

val matrix = Array(

Array(1, 2, 3),

Array(0, 4, 5),

Array(0, 1, 6)

)

val isUpperTriangular = checkUpperTriangular(matrix)

if (isUpperTriangular) {

println("The matrix is upper triangular.")

}

else {

println("The matrix is not upper triangular.")

}

}

def checkUpperTriangular(matrix: Array[Array[Int]]): Boolean = {

val n = matrix.length

for (i <- 1 until n) {

for (j <- 0 until i) {

if (matrix(i)(j) != 0) {

return false

}

}

}

true

}

}

**4] Write a Program to sort the matrix using insertion sort.**

object MatrixInsertionSort {

def main(args: Array[String]): Unit = {

val matrix = Array(

Array(5, 2, 9),

Array(1, 7, 6),

Array(4, 3, 8)

)

println("Original Matrix:")

printMatrix(matrix)

val sortedMatrix = sortMatrix(matrix)

println("\nSorted Matrix:")

printMatrix(sortedMatrix)

}

def sortMatrix(matrix: Array[Array[Int]]): Array[Array[Int]] = {

for (row <- matrix.indices) {

matrix(row) = insertionSort(matrix(row))

}

matrix

}

def insertionSort(arr: Array[Int]): Array[Int] = {

val n = arr.length

for (i <- 1 until n) {

val key = arr(i)

var j = i - 1

while (j >= 0 && arr(j) > key) {

arr(j + 1) = arr(j)

j -= 1

}

arr(j + 1) = key

}

arr

}

def printMatrix(matrix: Array[Array[Int]]): Unit = {

for (row <- matrix) {

for (elem <- row) {

print(s"$elem\t")

}

println()

}

}

}

**Assignment no. 3 (String)**

**1]**

Object TwoStrings1 {

def main(args: Array[String]): Unit = {

println("Enter the first string:")

val s1 = scala.io.StdIn.readLine()

println("Enter the second string:")

val s2 = scala.io.StdIn.readLine()

// a. Concatenate two strings

val concatenatedString = s1 + s2

println(concatenatedString)

// b. Check if the first string ends with "la"

val endsWithLa = s1.endsWith("la")

println(endsWithLa)

// c. Find the index of character 'a' in the second string

val indexOfA = s2.indexOf('a')

println(indexOfA)

}

}

**2] object TwoStrings2 {**

def main(args: Array[String]): Unit = {

println("Enter the first string:")

val s1 = scala.io.StdIn.readLine()

println("Enter the second string:")

val s2 = scala.io.StdIn.readLine()

val isEqualUsingOperator = s1 == s2

println(isEqualUsingOperator)

val isEqualUsingEquals = s1.equals(s2)

val isEqualUsingCompareTo = s1.compareTo(s2)

println(isEqualUsingEquals)

println(isEqualUsingCompareTo)

if (s1.length >= 5) {

val charAtPosition5 = s1.charAt(5)

println(charAtPosition5)

} else {

println("The first string is not long enough to find the character at position 5.")

}

}

}

**Assignment No.5 (List,Map & Set)**

**List**

**1] Create Lists using Five different methods (Lisp style,Java style,fill,range & tabulate methods)**

import scala.collection.immutable.\_

object assi5\_1 extends App{

//Lisp style

print("Using Lisp Style : ")

val l1 = 49 :: 23 :: 5 :: 56 :: Nil

println(l1)

//Java style

print("Using Java Style : ")

val l2 = List(34,90,67,24)

println(l2)

//fill method

print("Using fill method : ")

val l3 = List.fill(5)("TANAYA")

println(l3)

//range method

print("Using range method : ")

val l4 = List.range(23, 45)

println(l4)

//tubalate method

print("Using tabulate method : ")

val l5 = List.tabulate(20)(n => n\*2)

println(l5)

}

**2] Create Two Lists and merge it and store the sorted in asending order.**

import scala.collection.immutable.\_

object assi5\_2 extends App{

val l1 = List(12,15,56,67,78,90,87)

val l2 = List(57,76,99,22,33)

//merge using ++

var L = l1 ++ l2

println(L)

//merge using :::

var L1 = l1 ::: l2

println(L1)

//merge using concat

var L3 = List.concat(l1,l2)

println(L3)

//sorting

println(L.sorted)

}

**3] Create List of Integers divisible by 3 from list containing numbers from 1 to 50**

import scala.collection.immutable.\_

import scala.collection.mutable.ListBuffer

object assi5\_3 extends App{

val a = List.range(1,50)

println(a)

var b: ListBuffer[Int] = ListBuffer[Int]()

a.foreach((element : Int)=>if(element%3==0)(b+=element))

print("Integers which are divisible by 3 upto 50 : ")

println(b)

}

**4] Create List of Even numbers up to 10 and calculate its product.**

import scala.collection.mutable.ListBuffer

object assi5\_4 extends App{

val a = List.range(1,11)

println(a)

var b = ListBuffer[Int]()

a.foreach((element : Int)=>if(element%2==0)(b+=element))

print("Even numbers upto 50 : ")

println(b)

}

**5] Write a Program to Create List with 10 members using function 3n^2+4n+6**

import scala.collection.immutable.\_

object assi5\_5 extends App{

val l1 = List.tabulate(10)(n=>3\*n\*n+4\*n+6)

println(l1)

}

**Map**

**1] Write a user difined functions to convert Lowercase letter to uppercase and call the function using Map.**

import scala.collection.immutable.\_

object assi5m\_1 extends App{

var friends = Map((3,"SHweTA"),(10,"SANDHYa"),(11,"Geeta"),(12,"Sakshii"))

var days = ("first"->"sunday" , "second"-> "monday" , "third"->"tuesday")

print("list of FRiends : ")

println(friends)

print("list of days : ")

println(days)

friends.foreach{

case(key,value) => println(value.toUpperCase())

}

}

**2] Write a Program to Create Map with Rollno and FirstName . Print all student information with same FirstName.**

import scala.collection.immutable.\_

object assi5m\_2 extends App{

var student = Map(57->"TANAYA",10->"SANDHYA",39->"GEETA",42->"SAKSHII" )

println(student)

}

**Set**

**1] Write a Program to Create 2 Sets and find the common elements between them**

import scala.collection.immutable.\_

object assi5S\_1 extends App{

val s1 = Set("t","e","j","s","i")

val s2= Set("5","s","t","i")

println("Set 1 : "+s1)

println("Set 2 : "+s2)

println("Set of Commom elements : "+s1.intersect(s2))

}

**2] Write a Program to Display largest and smallest element of the set**

import scala.collection.immutable.\_

object assi5S\_2 extends App{

val s1 = Set(67,34,89,29,75,10,8,4)

println(s1)

var(max,min)=(0,9999)

for(ele <- s1 )

{

if(ele > max){

max = ele

}

if(ele < min){

min = ele

}

}

println("Largest element is : "+max)

println("Smallest elements is : "+min)

}

**3] Write a Program to Merge Two sets and calculate product and average of all elements of the set.**

import scala.collection.immutable.\_

import math.\_

object assi5s\_3 extends App{

val s1 = Set(2,3,1)

val s2 = Set(5,3,6)

println("Set 1 : "+s1)

println("Set 2 : "+s2)

var s3 = s1 ++ s2

println("Merged list : " +s3)

val prod = s3.product

println(prod)

val len = s3.size

var prod1=1

for(i<-0 until s3.size){

prod1 = prod1 \* s3(i)

}

println(prod1)

}