**Academic Year 2024-25 Even**

**19CSE313 – Principles of Programming Language**

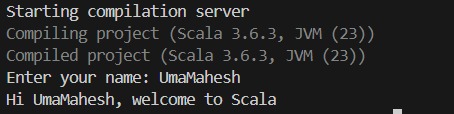
**B.Tech CSE 2022-26 F Section**

**Practice Set 8 – Array, List and Tuples in Scala**

**Scala Arrays**

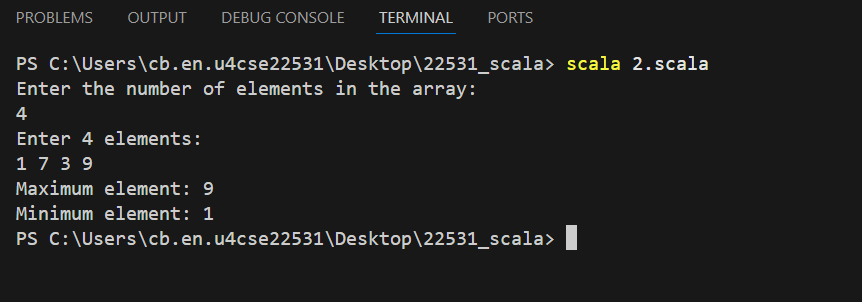
1. Read a name from user and print a welcome message in the format “Hi \_\_\_\_\_\_,welcome to Scala”

Output



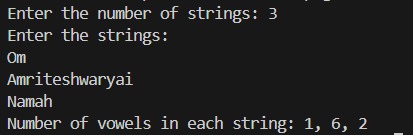
1. Read an array of integers and find the maximum and minimum elements.

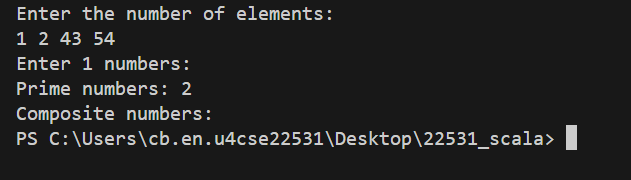
Output:



1. Read an array of N strings. Create another array of integers thatcontains the number of vowels in each string.

Output:



1. Create an array for storing N integers. Split the numbers into two separate arrays such that the first array contains the prime numbers present in the original array and the second one contains the composite numbers. Write and use a separate Scala function to check whether a number is prime or composite.
2. 

**Scala List**

1. Creating a list in Scala

object Scala\_List

{

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

{

println("Create a Scala List:")

println("Lisp style:")

val lisp\_list = 100 :: 200 :: 300 :: Nil :: 400 :: Nil

println(lisp\_list)

println("Java style:")

val nums = List(1,2,3,4,5,6,7)

println(nums)

println("Mixed type values in a list:")

val x = List[Number](100, 200, 110.20, 45d, 0x1)

println(x)

println("Range List:")

val y = List.range(1, 20)

println(y)

val z = List.range(0, 30, 3)

println(z)

println("Uniform List:")

val s = List.fill(5)("Scala")

println(s)

val n = List.fill(3)(4)

println(n)

println("Tabulate List:")

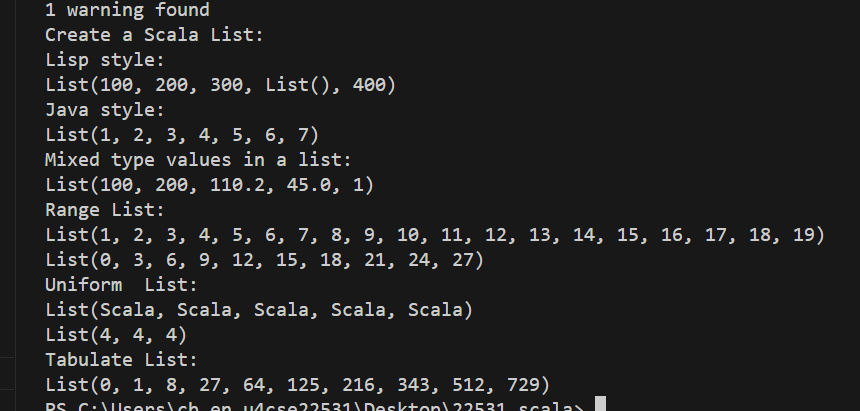
val t = List.tabulate(10)(n => n \* n \* n)

println(t)

}

}

Output:



1. Write a scala program to count the number of elements in the given list.
2. ListBuffer and List – Adding Items

object Scala\_List

{

import scala.collection.mutable.ListBuffer

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

{

//As a List is immutable we use ListBuffer

var colors = new ListBuffer[String]()

println("Add Single element in the said list:")

colors += "Red"

colors += "Green"

colors += "Black"

println(colors)

println("Add multiple elements in the said list:")

colors ++= List("Orange", "Pink", "Black")

println(colors)

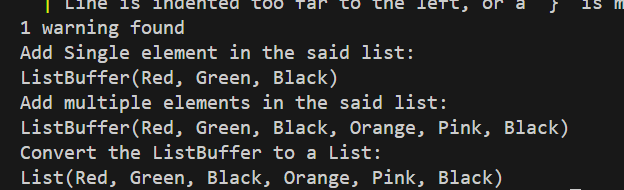
println("Convert the ListBuffer to a List:")

val colors\_list = colors.toList

println(colors\_list)

}

}



1. Removing Items

object Scala\_List

{

import scala.collection.mutable.ListBuffer

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

{

//As a List is immutable we use ListBuffer

//and finally convert the ListBuffer to list.

var colors = new ListBuffer[String]()

colors += "Red"

colors += "Green"

colors += "Black"

colors += "Orange"

colors += "Pink"

println("Original ListBuffer:")

println(colors)

println("Remove one element:")

colors -= "Red"

println(colors)

println("Remove multiple elements:")

println(colors)

colors --= Seq("Black", "Pink")

println("After removing two elements, final ListBuffer:")

println(colors)

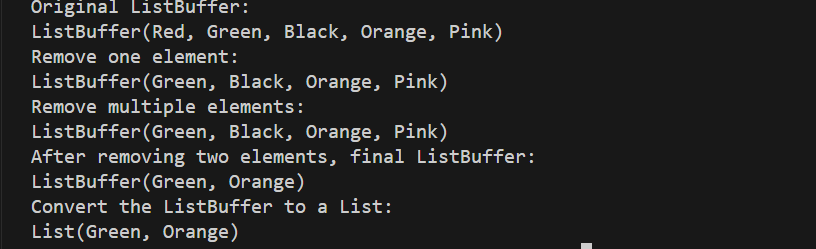
println("Convert the ListBuffer to a List:")

val colors\_list = colors.toList

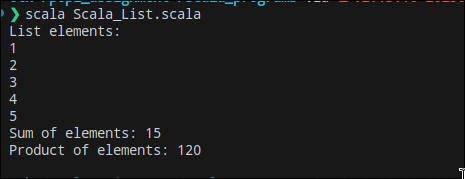
println(colors\_list)

}

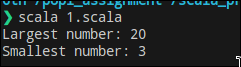
}



1. Edit the program in 3 such that you already have a list and must remove an item from the list [ list is immutable. You need to convert the list to ListBuffer if you need to do any add/delete]
2. Write a Scala program to iterate over a list to print the elements and calculate the sum and product of all elements of this list.



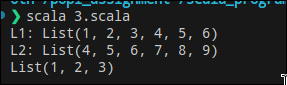
1. Write a Scala program to find the largest and smallest number from a given list.



1. Write a scala program to delete the duplicate elements in a list.



1. Write a scala program to find the difference between two lists [L1-L2 is the set of elements in L1 but not in L2]



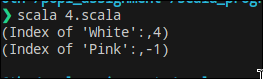
1. Try the following:

val colors = List("Red","Blue","Black","Green","White")

println("Index of 'White':", colors.indexOf("White"))

println("Index of 'Pink':", colors.indexOf("Pink"))

Output:



1. Merging two lists – different ways

val nums1 = List(1,3,5,7,9)

val nums2 = List(2,4,6,8,10)

val nums\_1 = nums1 ++ nums2

println(nums\_1)

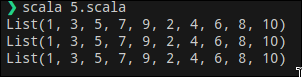
val nums\_2 = nums1 ::: nums2

println(nums\_2)

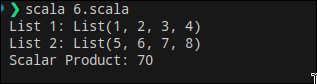
val nums\_3 = List.concat(nums1, nums2)

println(nums\_3)

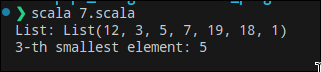
Output:



1. Write a Scala Program to find the scalar product of two lists.



1. Write a Scala Program to find the Kth smallest element in a given list.



**Scala Tuples**

1. Checking whether a tuple is empty or not

object CheckTupleEmptyExample {

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

val tuple1 = ()

val tuple2 = ("Red", 10, true)

val isEmpty1 = tuple1 == ()

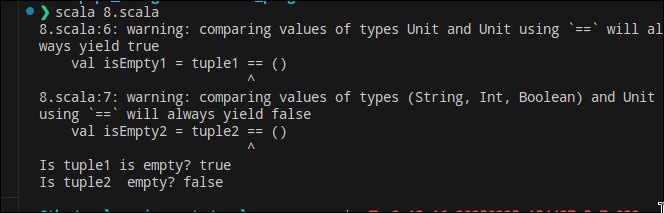
val isEmpty2 = tuple2 == ()

println("Is tuple1 is empty? " + isEmpty1)

println("Is tuple2 empty? " + isEmpty2)

}

}



1. Creating a tuple from a list

val list1 = List("Red", "Green", "Blue")

val list2 = List(1, 3, 5)

val new\_tuple = list1.zip(list2)

println("Tuple from said two lists: " + new\_tuple)

Output:



1. What does the code snippet do?

val tuple1 = (200, "Scala")

val tuple2 = (300, "Exercises")

println("Tuple1: "+tuple1)

val myTuple = (tuple1.\_1 + tuple2.\_1, tuple1.\_2 + " " + tuple2.\_2)

println("Concatenated tuple: " + concatenatedTuple)

1. Write a Scala program to find distinct elements of a tuple.



1. Write a Scala program to check whether two tuples are equal.

Output:

