Write the scala code to implement bubble sort algorithmA computer screen shot of a program code

Description automatically generated

Write scala code to find the length of each word from the array.

A computer screen shot of white text

Description automatically generated

Write scala code to find the number of books published by each author, referring to the collection given below, with (authorName, BookName)

{ (‘ Dr. Seuss’: ‘How the Grinch Stole Christmas!’) ,(‘ Jon Stone’: ‘Monsters at the End of This Book’ ) , (‘ Dr. Seuss’: ‘The Lorax’ ) ,( ‘Jon Stone’: ‘Big Bird in China’ ) ( ‘ Dr. Seuss’ :’ One Fish, Two Fish, Red Fish, Blue Fish’ ) }

A computer code with white text

Description automatically generated

Write the program to illustrate the use of pattern matching in scala, for the following

Matching on case classes.

Define two case classes as below:

abstract class Notification

case class Email(sender: String, title: String, body: String) extends Notification

case class SMS(caller: String, message: String) extends Notification

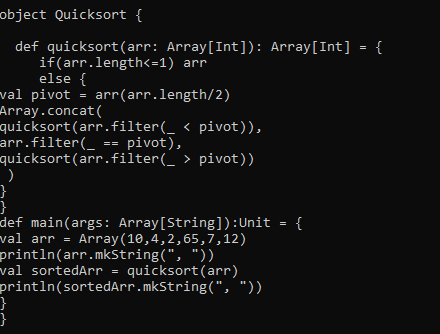
Define a function showNotification which takes as a parameter the abstract type Notification and matches on the type of Notification (i.e. it figures out whether it’s an Email or SMS).

In the case it’s an Email(email, title, \_) return the string: s"You got an email from $email with title: $title“

In the case it’s an SMS return the String: s"You got an SMS from $number! Message: $message“

A computer screen with white text

Description automatically generated

Write the scala program using imperative style to implement quick sort algorithm.

Write a scala function to convert the each word to capitalize each word in the given sentence.

A computer screen shot of white text

Description automatically generated

Write scala code to show functional style program to implement quick sort algorithm.

A computer screen shot of a program

Description automatically generated

For the below given collection of items with item-names and quantity, write the scala code for the given statement.

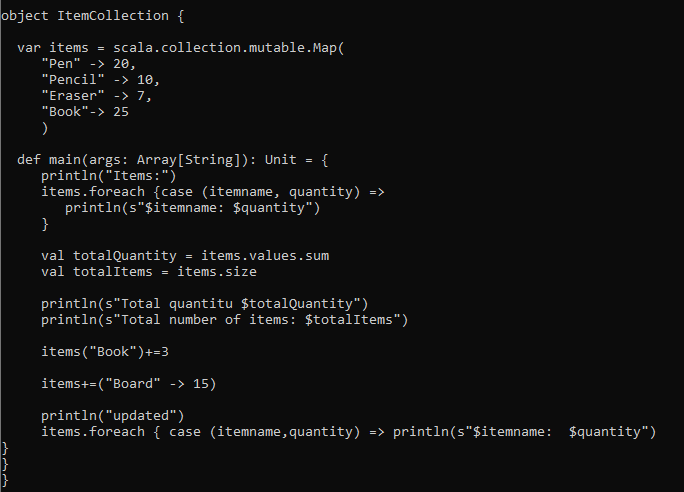
Items = {(“Pen”:20), (“Pencil”:10), (“Erasor”:7), (“Book”:25), (“Sheet”:15)}

i. Display item-name and quantity

ii. Display sum of quantity and total number of items

iii. Add 3 Books to the collection

Add new item “Board” with quantity 15 to the collection



Develop a scala code to search an element in the given list of numbers. The function Search() will take two arguments: list of numbers and the number to be searched. The function will write True if the number is found, False otherwise.

A computer screen shot of a black screen

Description automatically generated

Illustrate the implementation by writing scala code to generate a down counter from 10 to 1.

A computer screen with white text

Description automatically generated

Design a scala function to perform factorial item in the given collection. The arguments passed to the function are the collection of items. Assume the type of the argument for the function suitably. The return type is to be integer.

A computer screen with white text

Description automatically generated

12 same as pen pencil eraser

Implement function for binary search using recursion in Scala to find the number, given a list of numbers. The function will have two arguments: Sorted list of numbers and the number to be searched.A computer screen shot of a black screen

Description automatically generated

Write a function to find the length of each word and return the word with highest length .

Ex for the collection of words = (“games”, “television”,”rope”,”table”)

The function should return (“television”,10). The word with the highest length .

Read the words from the keyboard.

A computer screen shot of a black background

Description automatically generated

Spark programs

Analyze the application of fold() and aggregate() functions in Spark by considering a scenario where all the items in a collection are updated by a count of 100. Evaluate the efficiency, performance, and suitability of both.

A black screen with white text

Description automatically generatedA black screen with white text

Description automatically generated

Consider a text file text.txt. Develop Spark code to read the file and count the number of occurrences of each word using Spark RDD. Store the result in a file. Display the words which appear more than 4 times.

A screenshot of a computer program

Description automatically generated

Consider the content of text file text.txt. Perform the counting of occurrences of each word using pair RDD.

Write the Spark Code to print the top 10 tweeters.

Tweet Mining: A dataset with the 8198 reduced tweets, reduced-tweets.json will be provided. The data contains reduced tweets as in the sample below:

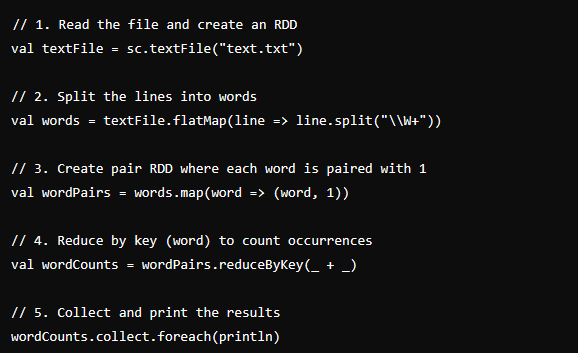
{"id":"572692378957430785",

"user":"Srkian\_nishu smile",

"text":"@always\_nidhi @YouTube no idnt understand bti loved of this mve is rocking",

"place":"Orissa",

"country":"India"}

A function to parse the tweets into an RDD will be provided.

A screenshot of a computer program

Description automatically generated

Qn 19 is weird  
import org.apache.spark.sql.SparkSession

import org.apache.spark.sql.functions.\_

// Initialize Spark session

val spark = SparkSession.builder

.appName("SocketWordCount")

.master("local[\*]") // Use 'local[\*]' for local testing

.getOrCreate()

// Reduce logging verbosity

spark.sparkContext.setLogLevel("ERROR")

// Create a StreamingContext with a 5-second batch interval

val ssc = new org.apache.spark.streaming.StreamingContext(spark.sparkContext, org.apache.spark.streaming.Seconds(5))

// Create a DStream that represents streaming data from a TCP source

val lines = ssc.socketTextStream("localhost", 9999)

// Split each line into words

val words = lines.flatMap(\_.split(" "))

// Count each word in each batch

val wordCounts = words.map(word => (word, 1)).reduceByKey(\_ + \_)

// Print the word counts

wordCounts.print()

// Start the streaming computation

ssc.start()

// Wait for the streaming to finish

ssc.awaitTermination()