Spark Scala ASSIGNMenT:

Go to terminal:

mescoe@mescoe-Lenovo-S510:~$ source ~/.bashrc

mescoe@mescoe-Lenovo-S510:~$ spark-shell

Programs:

1. **helloworld**

scala> print("Hello WOrld")

Hello WOrld

1. **addition**

scala> print(3+4)

7

1. **Largest of 2nos**

scala> :paste

// Entering paste mode (ctrl-D to finish)

object FindLargestEx{

def main(args: Array[String]) {

var num1 = 24;

var num2 = 76;

if( num1 > num2 ) {

println("Largest number is: " + num1);

}

else{

println("Largest number is: " + num2);

}

}

}

// Exiting paste mode, now interpreting.

defined object FindLargestEx

scala> FindLargestEx.main(Array.empty[String])

Largest number is: 76

1. **Sum of 2 nos**:

scala> :paste

// Entering paste mode (ctrl-D to finish)

object SumEx{

def main(args: Array[String]){

var a = 24;

var b = 76;

var sum = a+b;

println("Sum of two numbers is: " + sum);

}

}

// Exiting paste mode, now interpreting.

defined object SumEx

scala> SumEx.main(Array.empty[String])

Sum of two numbers is: 100

**5. Check whether number is prime or not**

scala> :paste

// Entering paste mode (ctrl-D to finish)

object PrimeCheck {

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

print("Enter a number: ")

val num = scala.io.StdIn.readInt()

val isPrime = num > 1 && !(2 until num).exists(num % \_ == 0)

println(if (isPrime) s"$num is Prime" else s"$num is Not Prime")

}

}

// Exiting paste mode, now interpreting.

defined object PrimeCheck

scala> PrimeCheck.main(Array.empty[String])

Enter a number: 9 is Not Prime

**6. Check whether no is odd or even**

scala> :paste

// Entering paste mode (ctrl-D to finish)

object EvenOddCheck {

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

print("Enter a number: ")

val num = scala.io.StdIn.readInt()

println(if (num % 2 == 0) s"$num is Even" else s"$num is Odd")

}

}

// Exiting paste mode, now interpreting.

defined object EvenOddCheck

scala> EvenOddCheck.main(Array.empty[String])

Enter a number: 8 is Even

**7. Printing fibonacci series**

scala> :paste

// Entering paste mode (ctrl-D to finish)

import scala.io.StdIn

object FibonacciSeries {

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

print("Enter the number of terms: ")

val n = StdIn.readInt() // Using StdIn.readInt() after import

var a = 0

var b = 1

print("Fibonacci Series: ")

for (\_ <- 1 to n) {

print(s"$a ")

val temp = a + b

a = b

b = temp

}

println() // For a new line after printing

}

}

// Exiting paste mode, now interpreting.

import scala.io.StdIn

defined object FibonacciSeries

scala> FibonacciSeries.main(Array.empty[String])

Enter the number of terms: Fibonacci Series: 0 1 1 2

scala>

Hadoop link:

<https://medium.com/@abhikdey06/apache-hadoop-3-3-6-installation-on-ubuntu-22-04-14516bceec85>

viva questions:  
Perfect, Vishwas!  
Here are the **basic Scala + Apache Spark viva questions with short, simple answers** for you ✅:

### 📚 General Questions:

**Q1. What is Apache Spark?**  
**Ans:** Spark is an open-source, fast, distributed computing framework for big data processing. It supports in-memory computation to speed up processing.

**Q2. Why do we use Scala with Apache Spark?**  
**Ans:** Spark is written in Scala, so Scala provides better compatibility, performance, and access to low-level Spark features compared to other languages.

**Q3. What is RDD in Spark?**  
**Ans:** RDD (Resilient Distributed Dataset) is a distributed collection of data across nodes in a cluster. It is the fundamental data structure of Spark.

**Q4. What are the main features of Apache Spark?**  
**Ans:**

* In-memory computation
* Fault tolerance
* Lazy evaluation
* Supports multiple languages (Scala, Python, Java, R)
* Easy integration with Hadoop

**Q5. What is lazy evaluation in Spark?**  
**Ans:** Spark doesn't compute data immediately; it builds a lineage. Computation happens only when an action is called.

**Q6. What is the difference between map() and flatMap() in Spark?**  
**Ans:**

* map() returns one output element for each input element.
* flatMap() can return multiple output elements for one input element.

**Q7. Explain the Spark execution flow.**  
**Ans:**

* Create RDD → Apply Transformations → Execute Actions → Job → Tasks → Stage → Execute on Cluster.

**Q8. What are transformations and actions in Spark?**  
**Ans:**

* **Transformation:** Lazy operations (e.g., map, filter) that define a new RDD.
* **Action:** Operations (e.g., collect, count) that trigger execution.

**Q9. How is Spark better than Hadoop MapReduce?**  
**Ans:** Spark uses in-memory processing, which is much faster compared to Hadoop's disk-based processing.

**Q10. What are different cluster managers Spark can use?**  
**Ans:**

* Standalone cluster manager
* YARN (Hadoop)
* Mesos
* Kubernetes

### 🛠️ Scala + Spark Program Specific:

**Q11. How do you initialize a SparkContext in Scala?**  
**Ans:**

val spark = SparkSession.builder().appName("AppName").master("local").getOrCreate()

val sc = spark.sparkContext

**Q12. What is the purpose of SparkSession?**  
**Ans:** SparkSession is the entry point to work with RDD, DataFrame, and Dataset APIs in Spark.

**Q13. Write a basic Scala program that counts words using Spark.**  
**Ans:**

val textFile = sc.textFile("path/to/file.txt")

val words = textFile.flatMap(line => line.split(" "))

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

wordCounts.collect().foreach(println)

**Q14. How do you read a text file in Spark using Scala?**  
**Ans:**

val data = sc.textFile("path/to/yourfile.txt")

**Q15. How do you perform filtering of RDDs in Spark with Scala?**  
**Ans:**

val rdd = sc.parallelize(List(1,2,3,4,5))

val filtered = rdd.filter(x => x > 2)

filtered.collect()

**Q16. What is the difference between RDD, DataFrame, and Dataset?**  
**Ans:**

* **RDD:** Low-level distributed data.
* **DataFrame:** Table-like structure with schema (like SQL table).
* **Dataset:** Type-safe, combines RDD and DataFrame features (only in Scala/Java).

**Q17. Can you explain how to create an RDD manually in Scala Spark?**  
**Ans:**

val rdd = sc.parallelize(Seq(1,2,3,4))

**Q18. What is an accumulator in Spark?**  
**Ans:** A variable used to perform counters and sums across workers in parallel.

**Q19. What is caching in Spark? Why and how would you use it?**  
**Ans:**

* Caching stores intermediate RDDs in memory to speed up repeated computations.
* Use .cache() method. Example: rdd.cache()

**Q20. What is the significance of .collect() in Spark?**  
**Ans:**  
collect() returns all elements of the RDD/DataFrame to the driver node.

### ⚡ Short Conceptual Questions:

**Q21. What are key-value RDDs?**  
**Ans:** RDDs where each element is a pair (key, value), useful for aggregation operations.

**Q22. What is partitioning in Spark?**  
**Ans:** Dividing data into parts for parallel processing across multiple nodes.

**Q23. What happens if we call an action without any transformation?**  
**Ans:** The action will execute immediately on the existing data without transformation.

**Q24. What is the role of parallelize() in Spark?**  
**Ans:** It creates an RDD from an existing collection (like a list or array).

**Q25. Can Spark run without Hadoop?**  
**Ans:**  
**Yes,** Spark can run independently using its own cluster manager or in standalone mode.

Would you also like me to give you a **very small full Scala+Spark sample program** ready for viva if you want to show your examiner? 📜🚀  
(Just reply "Yes sample program" if you want!)