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

1. Data For Task

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

Inceptez1.txt

I am learning Apache Spark from Inceptez Learning Resources

I am learning Apache Hadoop from Inceptez Learning Resources

I have created my technical profile at [www.QuickTechie.com](http://www.quicktechie.com/)

I am learning Apache Spark from Training4exam Learning Resources

Accomplish the followings:-

1. Create this text file in HDFS

2. Once file is created, write the spark application which will read from HDFS as an RDD

3. Once RDD loaded, do the line count of this RDD

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

2. Data For Task

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

Inceptez2.txt

I am learning Apache Spark from Inceptez Learning Resources

I am learning Apache Hadoop from Inceptez Learning Resources

I have created my technical profile at [www.QuickTechie.com](http://www.quicktechie.com/)

I am learning Apache Spark from Training4exam Learning Resources

I am learning Apache Spark from Training4exam Learning Resources

Accomplish the followings:-

1. Create this file in HDFS

2. Once file is created, write a spark application which will read this file from HDFS as an RDD

3. Filter all records contains 'Inceptez' in line and count of lines

4. Filter all records does not contains 'Inceptez' in line and count the lines

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

3. Data For Task

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

("We", "Are" ,"Learning" , "Hadoop" , "From" , "Inceptez" , "We", "Are" ,"Learning" , "Spark" , "From" , "Inceptez.com" , "hadoop" , "HADOOP")

Accomplish the followings:-

1. Create an RDD using using the given words

2. Once RDD is created count all the words

3. Now filter out all the words which does not have Hadoop keyword, however make sure it would count all the different cases(upper/lower) as well

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

4. Data For Task

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

hdpcd/Inceptez4A.txt

Inceptez.com QuickTechie.com Training4Exam.com

hdpcd/Inceptez4B.txt

Hadoop Spark Scala Python Java Cloud Science

hdpcd/Inceptez4C.txt

India USA UK Canada Australia

Accomplish the followings:-

1. Load all 3 files in different RDDS

2. Concatnate all the data in single RDDS

3. Count all the words in of all 3 files

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

5. Data For Task

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

hdpcd/dir5A/InceptezA.txt

Inceptez.com QuickTechie.com Training4Exam.com

hdpcd/dir5B/InceptezB.txt

Hadoop Spark Scala Python Java Cloud Science

hdpcd/dir5C/InceptezC.txt

India USA UK Canada Australia

Accomplish the followings:-

1. Load the data given in all 3 directories

2. Once the data is loaded filter the words which contains "Hadoop"

3. Now persist the data in memory

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

6. Data For Task

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

You have given the follwing data in a file hadoophortonworks6.txt

CourseName,Price,TaxandOthersInPercen

Hadoop,2900,15

Spark,3500,14

AWS,2700,13

Azure,2800,11

Java,3000,16

HBase,3200,20

Accomplish the followings:-

1. Load this csv file in RDD

2. Now calculate the final price using tax

3. Save the final data in HDFS

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

7. Data For Task

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

You have given 3 csv files hdfs as below:-

EmployeeManager.csv

E01,Vishnu

E02,Satyam

E03,Shiv

E04,Sundar

E05,John

E06,Pallavi

E07,Tanvir

E08,Shekhar

E09,Vinod

E10,Jitendra

EmployeeName.csv

E01,Lokesh

E02,Bhupesh

E03,Amit

E04,Ratan

E05,Dinesh

E06,Pavan

E07,Tejas

E08,Sheela

E09,Kumar

E10,Venkat

EmployeeSalary.csv

E01,50000

E02,50000

E03,45000

E04,45000

E05,50000

E06,45000

E07,50000

[E08,10000](https://maps.google.com/?q=E08,10000&entry=gmail&source=g)

[E09,10000](https://maps.google.com/?q=E09,10000&entry=gmail&source=g)

E10,10000

Accomplish the following:-

Using spark and its API you have to generate a joined output as below and save as text file (Separated by comma)

for final distribution and output must be sorted by id

id,name,salary,managerName

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

8. Data For Task

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

You have given following 2 files:-

Content.txt

Hello this is Inceptez.com

This is QuickTechie.com

Apache Spark Training

This is Spark Learning Session

Spark is faster than MapReduce

Remove.txt

Hello, is, this, the

Accomplish the following:-

write the spark program which reads the content.txt file and load as RDD, remove all the words from the Remove.txt(which is loaded as an RDDs of words from Remove.txt). And count the occurances of each word and save it as text file in HDFS

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

9. Data For Task

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

You have given following 3 files as below:-

spark3/sparkdir1/file1.txt

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework

spark3/sparkdir2/file2.txt

The core of Apache Hadoop consists of a storage part known as Hadoop Distributed File System (HDFS) and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers packaged code for nodes to process in parallel based on the data that needs to be processed.

spark3/sparkdir3/file3.txt

his approach takes advantage of data locality nodes manipulating the data they have access to to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking

Accomplish the following:-

write the spark program which loads the all 3 file from hdfs and do the word count by filtering the following words

And result should be sorted by word count in reverse order

Filter words("a","the","an","as","a","with","this","these","is","are","in","for","to","and","The","of")

Also please make sure you load all 3 files as Single RDD(All 3 files must be loaded using single API call)

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

10. Data For Task

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

Given filename spark8/data.csv

data.csv

1,Lokesh

2,Bhupesh

2,Amit

2,Ratan

2,Dinesh

1,Pavan

1,Tejas

2,Sheela

1,Kumar

1,Venkat

Accomplish the followings:-

Load this file from hdfs, save it back as (id,(all names of same type)) in result directory.

However make sure while saving it should be able to write in a single file.

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

11. Data For Task

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

Given a file name spark6/user.csv

user.csv

id,topic,hits

Rahul,scala,120

Nikita,spark,80

Mithun,spark,1

myself,cca175,180

Accomplish the followings:-

Write spark code in scala which will remove the header part and create RDD of values as below,for all rows.

And also if id is "myself" than filter out row.

Map(id->om,topic->scala,hits->120)

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

12. Data For Task

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

Given a filename spark7/EmployeeName.csv

EmployeeName.csv

E01,Lokesh

E02,Bhupesh

E03,Amit

E04,Ratan

E05,Dinesh

E06,Pavan

E07,Tejas

E08,Sheela

E09,Kumar

E10,Venkat

Accomplish the followings:-

Load the file from hdfs and sort it by name and save it back as (id,name) in results directory.

However make sure while saving it should be able to write in a single file

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

13. Data For Task

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

Given 2 files

spark16/file1.txt

1,9,5

2,7,4

3,8,3

spark16/file2.txt

1,g,h

2,i,j

3,k,l

Accomplish the followings:-

Load these 2 files as Spark RDD and join them to produce the below results

(1,((9,5),(g,h)))

(2,((7,4),(i,j)))

(3,((8,3),(k,l)))

And write code snippet which will sum the second columns of above joined results (5+4+3)

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

14. Data For Task

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

You have given file spark10/sales.txt below:-

spark10/sales.txt

Department,Designation,costToCompany,State

Sales,Trainee,12000,UP

Sales,Lead,32000,AP

Sales,Lead,32000,LA

Sales,Lead,32000,TN

Sales,Lead,32000,AP

Sales,Lead,32000,TN

Sales,Lead,32000,LA

Sales,Lead,32000,LA

Marketing,Associate,18000,TN

Marketing,Associate,18000,TN

HR,Manager,58000,TN

Accomplish the following:-

Produce the output as csv with group by department,designation,state and additional column with sum(costToCompany) and

TotalEmployeeCount

Should get results like

Dept,Desg,State,empcount,totalcost

Sales,Lead,AP,2,64000

Sales,Lead,LA,3,96000

Sales,Lead,TN,2,64000

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

15. Data For Task

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

You have been given below list in scala (name,sex,cost) for each work done.

List( ("Deepak" , "male", 4000), ("Deepak" , "male", 2000), ("Deepika" , "female", 2000),("Deepak" , "female", 2000), ("Deepak" , "male", 1000) , ("Neeta" , "female", 2000))

Accomplish the followings:-

Now write a spark program to load this list as an RDD and do the sum of cost for combination of name and sex (as key)

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

16. Data For Task

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

val a = sc.parallelize(List("dog","tiger","lion","cat","panther","eagle"))

val b = a.map(x => (x,x.length))

Accomplish the following:-

Array[(Int, String)] = Array((4,lion), (5,tigereagle), (3,dogcat), (7,panther))

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

17. Data For Task

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

val pairRDD1 = sc.parallelize(List(("cat",2),("cat",5),("book",4),("cat",12)))

val pairRDD2 = sc.parallelize(List(("cat",2),("cup",5),("mouse",4),("cat",12)))

operation 1

Accomplish the following:-

Write code for operation 1 which will produce desired output shown below

 Array[(String, (Option[Int], Option[Int]))] = Array((book,(Some(4),None)), (mouse,(None,Some(4))), (cup,(None,Some(5))), (cat,(Some(2),Some(2))), (cat,(Some(2),Some(12))), (cat,(Some(5),Some(2))), (cat,(Some(5),Some(12))), (cat,(Some(12),Some(2))), (cat,(Some(12),Some(12))))