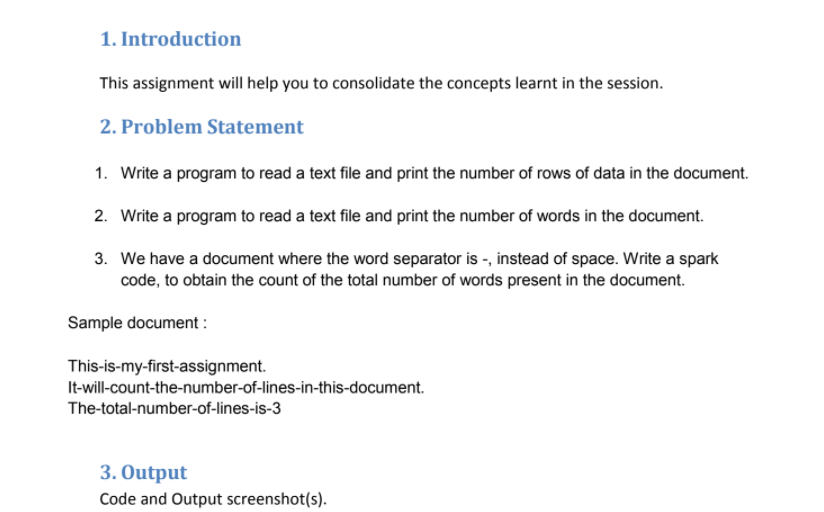


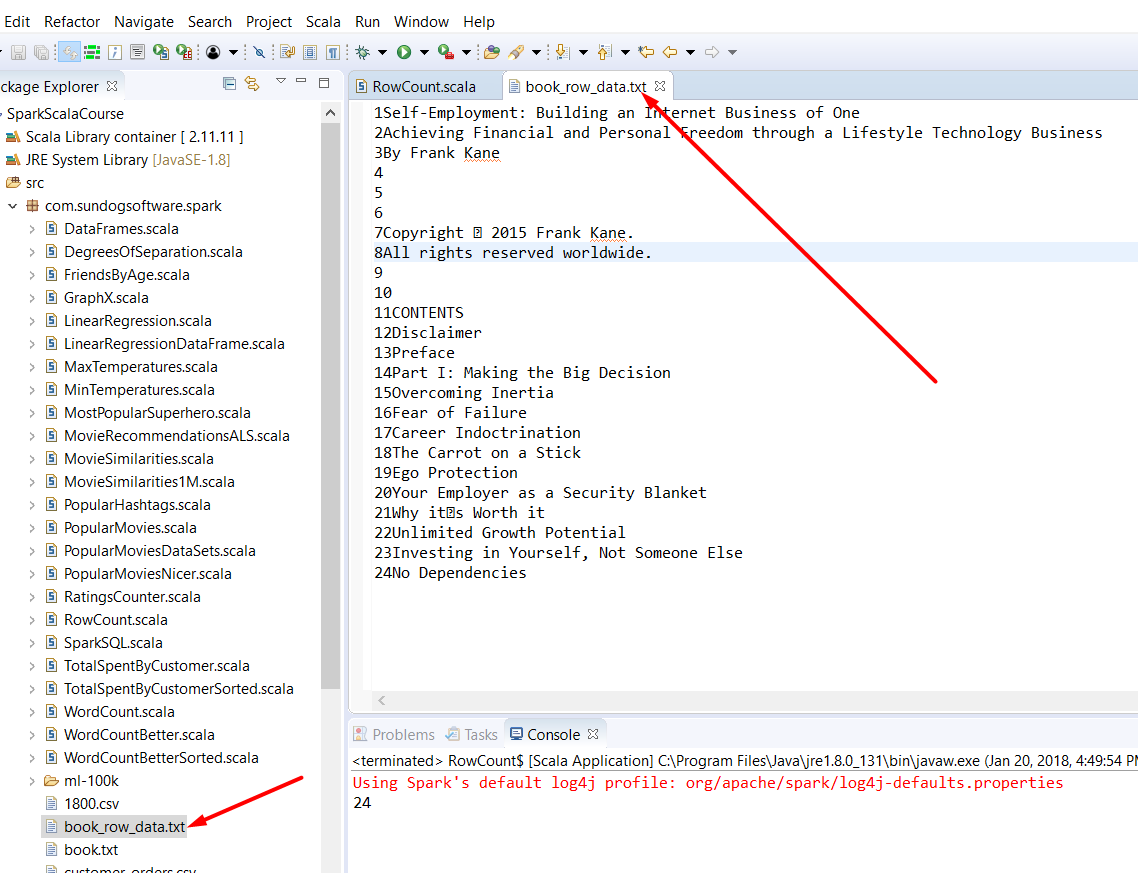
Outline



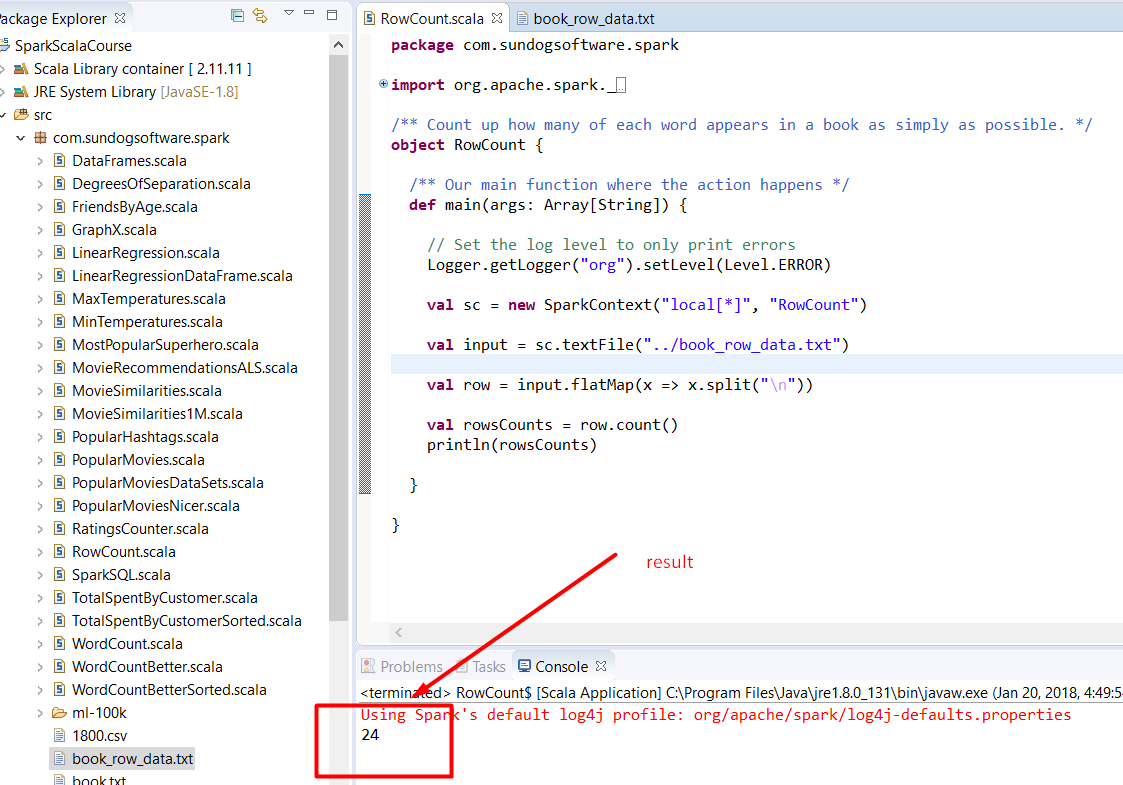
The Problem 1 :



The file with rows



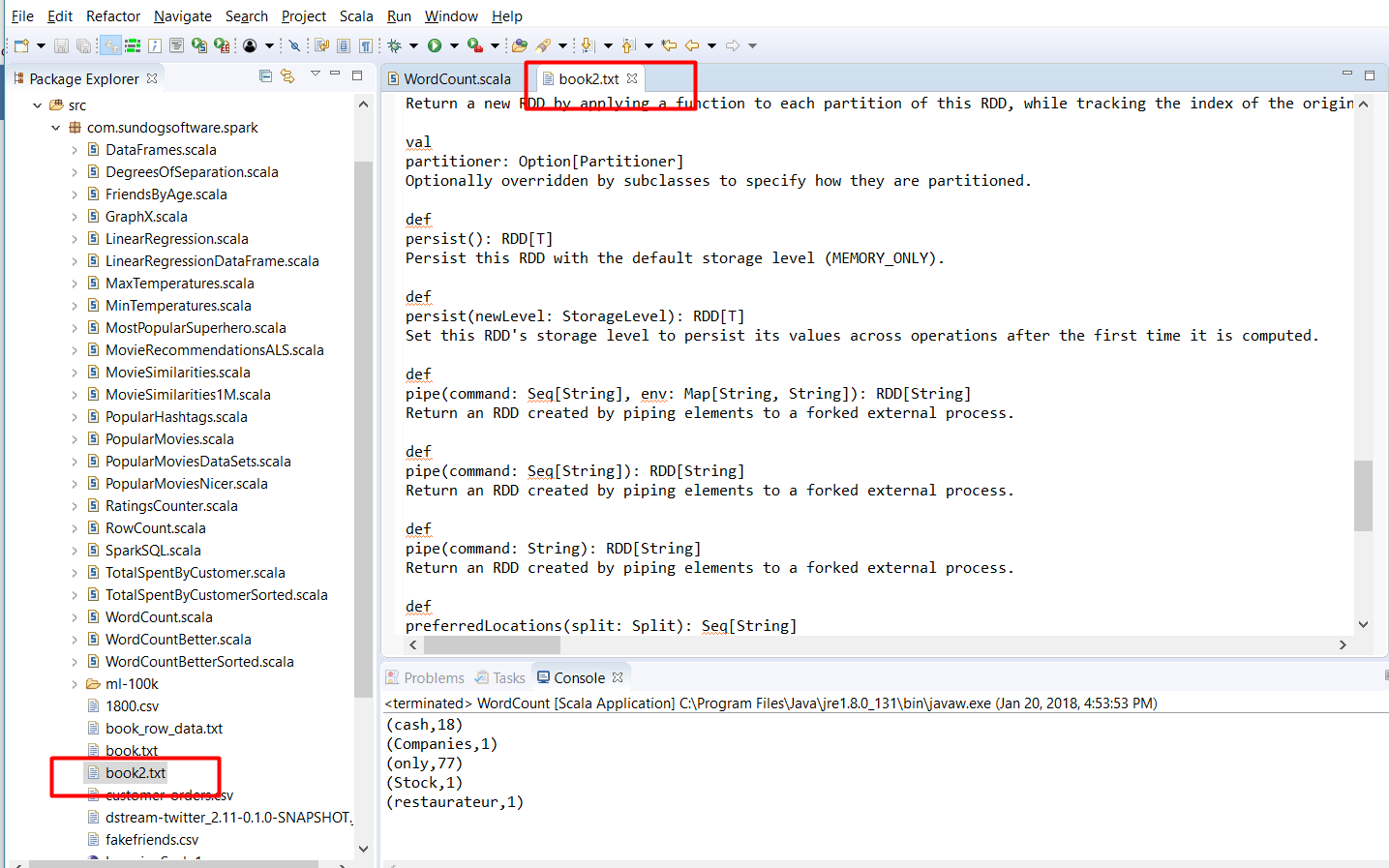
Let's read from this file :



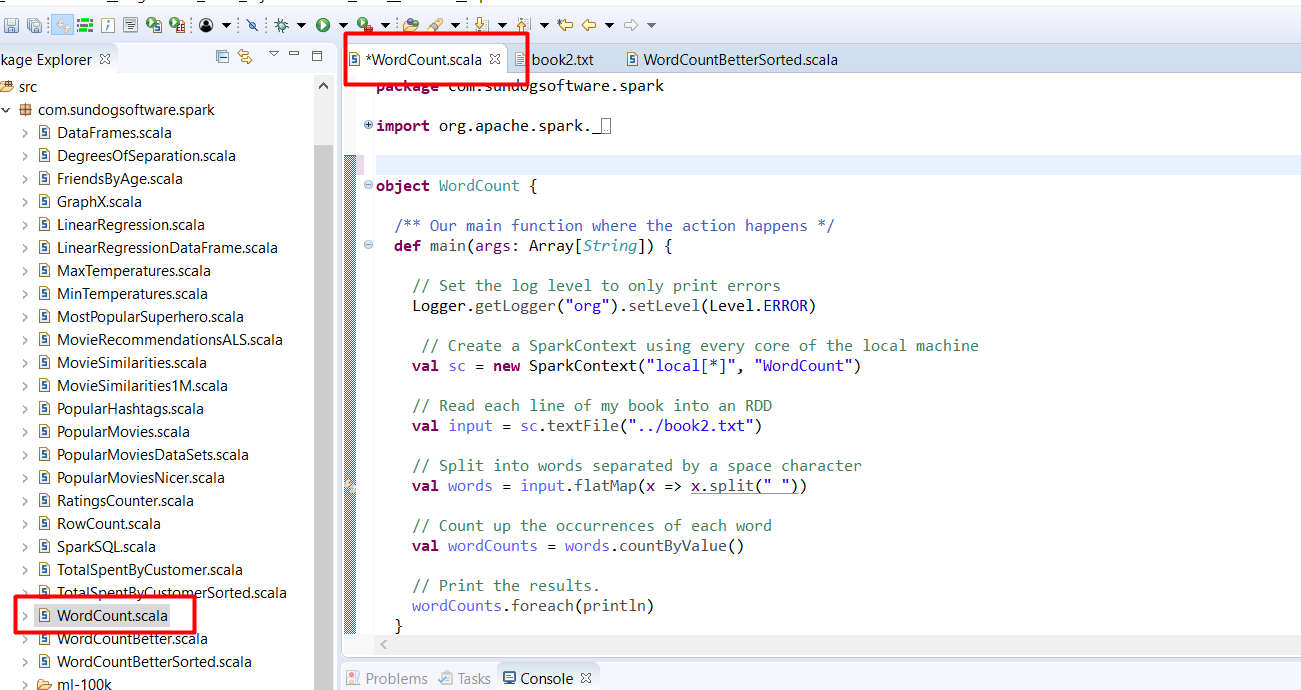
The Problem 2 :



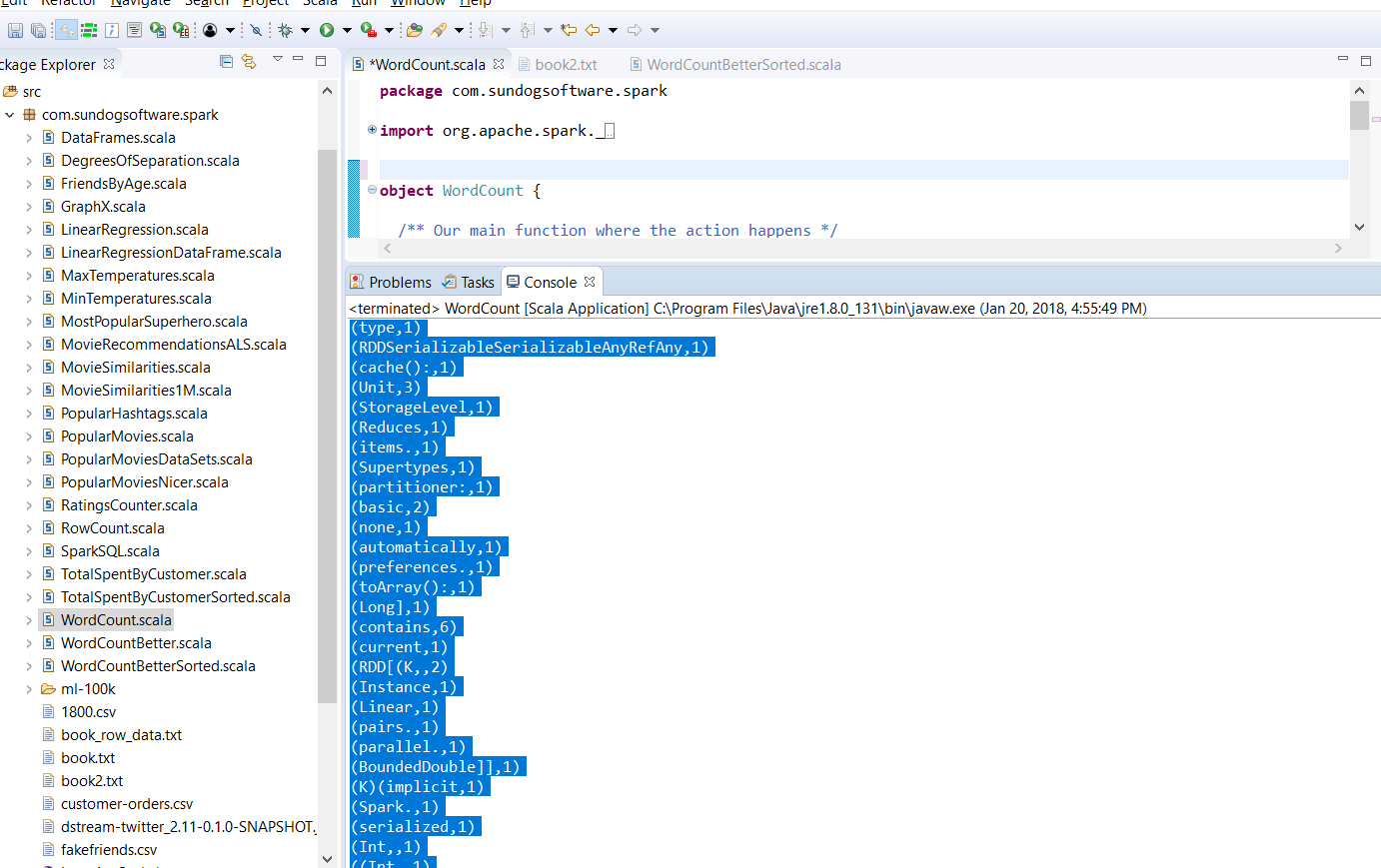
The file with data



The program code :



The result :



(abstraction,1)

(file),1)

(locations,2)

(RDD[Array[T]],1)

(Please,1)

(String]):,1)

(elements,19)

(StorageLevel):,1)

(execution,1)

(context:,1)

(is,,1)

(,48)

(ID,1)

(read,1)

(number,1)

(RDD[T]):,2)

(for,9)

(.distinct(),1)

(objects.,1)

(RDD[T],9)

(Set,2)

(one,,1)

(Return,21)

(SparkContext)(implicit,1)

(satisfy,1)

(parent,1)

(dependencies:,1)

(method,1)

(Split):,3)

(U),1)

(Take,1)

(Internally,,1)

(Int):,3)

(collect():,1)

(Optionally,2)

(Aggregate,2)

(any,2)

(across,1)

(operations,6)

(this,26)

(count,1)

(U)(seqOp:,1)

(countApprox(timeout:,1)

(in,12)

(mapPartitions[U](f:,1)

(SequenceFileRDDFunctions,1)

(based,1)

(overriding,1)

(have,1)

(Partitioner,1)

(are,2)

(is,7)

(How,1)

(addition,,1)

(HDFS,1)

(Function,1)

((use,1)

(RDD's,2)

(TraversableOnce[U])(implicit,1)

(All,1)

(custom,1)

(collection,1)

(first():,1)

(saveAsObjectFile(path:,1)

(given,3)

(using,4)

(union,2)

(scheduling,1)

(applying,4)

(process.,3)

(RDD,,2)

(incomplete,1)

(partition.,2)

(preferred,1)

(forked,3)

(specify,2)

("zero,2)

(This,1)

(In,1)

(binary,1)

(map,1)

(countByValue().,1)

(Known,1)

(saveAsTextFile(path:,1)

(iterator(split:,1)

(Concrete,1)

(SparkContext).,1)

(operated,1)

(Applies,1)

(system),1)

(another,3)

(Get,1)

(if,3)

(T,3)

(overridden,2)

(OrderingAlphabeticBy,1)

(Save,2)

(confidence:,2)

(Array[Split],1)

(own,1)

(=,3)

(glom():,1)

(StorageLevel.NONE,1)

(level,,1)

(SequenceFile,1)

(Dataset,1)

(appear,1)

(U)(implicit,2)

(all,12)

(array.,1)

(RDD(sc:,1)

(representations,1)

(U,1)

(methods,,1)

((value,,1)

(f,1)

(users,1)

(data,1)

(result,1)

(aggregate[U](zeroValue:,1)

(map,,1)

(Array[T],4)

(set.,1)

(it,1)

(Seq[String]):,1)

(reading,1)

(abstract,4)

(Cartesian,1)

(A,5)

(partition,3)

(a,29)

(results.,1)

(computed.,1)

(splits.size):,1)

(compute,2)

((T),6)

(within,2)

(as,6)

(pairs,,1)

(RDD;,1)

((Experimental),2)

(Optionally,,2)

(id:,1)

(Long,,2)

(PartialResult[Map[T,,1)

(Double,,1)

(count():,1)

(persist(newLevel:,1)

(text,1)

(computing,3)

(numSplits:,1)

(reduce(f:,1)

(SparkContext,2)

(string,1)

(specified,1)

(InheritedHide,1)

(or,2)

(piping,3)

(These,1)

(union(other:,1)

(RDD[String],3)

(filter(f:,1)

(⇒,12)

(main,1)

(Resilient,1)

(subclasses,2)

((partitions),1)

(take(num:,1)

(file,,1)

(such,2)

(they,1)

(Double,2)

(seed:,2)

(each,10)

(Int)(implicit,1)

(way,1)

(operator.,1)

(associative,2)

(getStorageLevel:,1)

(Value,2)

(T)(op:,1)

(immutable,,1)

(join;,1)

(RDD[U])(implicit,1)

((a,,1)

(foreach(f:,1)

(distinct(numSplits:,1)

(DoubleRDDFunctions,1)

((within,1)

(RDDs,,1)

(SequenceFiles.,1)

(spark.SparkContext.\_.,1)

(that,9)

(T),3)

(num,1)

(Iterator[U])(implicit,2)

(groupByKey,1)

(to,17)

(element,1)

(cache,1)

(Represents,1)

(b),1)

(PairRDDFunctions,1)

(you,1)

(Long,1)

(Int,2)

(array,2)

(these,2)

(Map[T,,1)

(filter,,1)

(functions.,1)

(was,1)

(flatMap[U](f:,1)

(version,2)

(them).,1)

(persist,1)

(potentially,1)

(env:,1)

(placement,1)

(value".,2)

(block,1)

(elements.,2)

((T,,2)

(through,1)

(depends,1)

(VisibilityPublicAll,1)

(The,1)

(Iterator[T]),1)

(import,1)

(partitioned.,1)

(++(other:,1)

(results,2)

(count),1)

(inheritance,1)

(U,,1)

(flattening,1)

(allowing,1)

(can,3)

(on,11)

(one.,2)

(tasks,1)

(saved,1)

(external,3)

(hash-partitioned),1)

(splits,1)

(how,1)

(b,1)

(Seq[String],1)

(grouped,2)

(after,1)

(persist():,1)

(spark,1)

(Abstract,1)

(properties:,1)

(subset,1)

(Constructors,1)

(AllShow,1)

(persist.,1)

(sampled,1)

(conversions,1)

(cartesian[U](other:,1)

(implicit,1)

(final,1)

(Persist,2)

(by,12)

(K,,1)

(tracking,1)

(returns,1)

(then,3)

(split,2)

((RDD),,1)

(even,1)

((U,,2)

((e.g.,4)

(function,7)

(fraction:,1)

(distinct,1)

(new,8)

(List[spark.Dependency[\_]],1)

(default,2)

(will,2)

(Approximate,2)

(pairs,1)

(count(),1)

(paper,1)

(Iterator[T],2)

(while,1)

(predicate.,1)

(key-value,2)

(Seq[String],,1)

(RDD.,11)

(not,1)

(partitioned,1)

(times,1)

(Internal,1)

(with,2)

(from,2)

(partition,,2)

(refer,1)

(Int)],1)

(RDD,31)

(combOp:,1)

(itself.,1)

(takeSample(withReplacement:,1)

(Unit):,1)

(implement,2)

(Map[String,,1)

(preferredLocations(split:,1)

(T):,2)

((MEMORY\_ONLY).,2)

(first,4)

(RDD[(Int,,1)

(combine,1)

(Spark,2)

(Subclasses,1)

(details,1)

(Distributed,1)

(eliminate,1)

(done,1)

(available,5)

(values,1)

(storage,5)

(class,2)

(its,3)

(extends,1)

(map[U](f:,1)

(on.,1)

(mapPartitionsWithSplit[U](f:,1)

(say,1)

(an,11)

(compute(split:,1)

(original,1)

(def,40)

(five,1)

((Iterator[T]),1)

(applicable,,1)

(be,2)

(timeout,,1)

(RDD[U],4)

(Boolean,,2)

(U)],1)

(finished.,1)

(into,1)

(functions,1)

(where,1)

(Option[Partitioner],1)

(Doubles;,1)

(RDDs.,1)

(RDD[(T,,1)

(Serializable,1)

(partitions,1)

(time,1)

(coalescing,1)

(multiple,1)

(Boolean):,1)

(pipe(command:,3)

(more,1)

(neutral,2)

(countByValueApprox(timeout:,1)

(type,1)

(RDDSerializableSerializableAnyRefAny,1)

(cache():,1)

(Unit,3)

(StorageLevel,1)

(Reduces,1)

(items.,1)

(Supertypes,1)

(partitioner:,1)

(basic,2)

(none,1)

(automatically,1)

(preferences.,1)

(toArray():,1)

(Long],1)

(contains,6)

(current,1)

(RDD[(K,,2)

(Instance,1)

(Linear,1)

(pairs.,1)

(parallel.,1)

(BoundedDouble]],1)

(K)(implicit,1)

(Spark.,1)

(serialized,1)

(Int,,1)

((Int,,1)

(arg0:,9)

(value,1)

(characterized,1)

(when,1)

(internals.,1)

(dependencies,1)

(it.,1)

(ClassManifest[T]),1)

(unique,2)

(Seq[T])],2)

(fold(zeroValue:,1)

(Any,1)

(other,1)

(of,37)

(splits:,1)

(PartialResult[BoundedDouble],1)

(sample(withReplacement:,1)

(containing,2)

(and,13)

(ClassManifest[K]):,2)

(val,3)

(ClassManifest[U]):,6)

(partitions,,2)

(other.,1)

(right,1)

(String):,3)

(list,3)

(Indeed,,1)

(otherwise,1)

(index,1)

(created,5)

(RDDs,6)

(num:,1)

(product,1)

(level,3)

(groupBy[K](f:,2)

(the,35)

(0.95):,2)

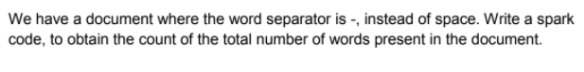
(identical,1)

(Members,2)

(countByValue():,1)

(only,3)

The Problem 3 :



**The data file :**

This-is-my-first-assignment.

It-will-count-the-number-of-lines-in-this-document.

The-total-number-of-lines-is-three

**The program and the results:**

