Solution Q3b: Illustrate algorithm 3.8 (with in-mapper combining. Apply your algorithm Q1).

INPUT	**Input Split-1**	**Input Split-2**

Mapper Input	cat mat rat cat
	cat bat cat pat
	cat bat rat bat
	
	Neighbours:
	
	N(cat) = {mat,rat}
	N(mat) ={rat,cat}
	N(rat) = {cat}
	N(cat) = {}
	
	N(cat) = {bat}
	N(bat) = {cat,pat}
	N(cat) = {pat}
	N(pat) = {}
	
	N(cat) =
	{bat,rat,bat}
	N(bat) = {rat}
	N(rat) = {bat}
	N(bat) = {}
		cat rat bat rat
		bat mat pat bat
		pat cat bat mat
		
		Neighbours:
		
		N(cat) = {rat,bat,rat}
		N(rat) = {bat}
		N(bat) = {rat}
		N(rat) = {}
		
		N(bat) = {mat,pat}
		N(mat) = {pat,bat}
		N(pat) = {bat}
		N(bat) = {}
		
		N(pat) = {cat,bat,mat}
		N(cat) = {bat,mat}
		N(bat) = {mat}
		N(mat) = {}

MAP	**Mapper-1**	**Mapper-2**
Mapper Output	((cat,mat),1) ((cat,rat),2) ((mat,rat),1) ((mat,cat),1) ((rat,cat),1) ((cat,bat),3) ((bat,cat),1) ((bat,pat),1) ((cat,pat),1) ((bat,rat),1) ((rat,bat),1)	((cat,rat),2) ((cat,bat),2) ((rat,bat),1) ((bat,rat),1) ((bat,mat),2) ((bat,pat),1) ((mat,pat),1) ((mat,bat),1) ((pat,bat),2) ((pat,cat),1) ((pat,mat),1) ((cat,mat),1)

PARTITION	**(a-j)**	**(k-z)**
	((cat,mat),1) ((cat,rat),2) ((cat,bat),3) ((bat,cat),1) ((bat,pat),1) ((cat,pat),1) ((bat,rat),1) ((cat,rat),2) ((cat,bat),2) ((bat,rat),1) ((bat,mat),2) ((bat,pat),1) ((cat,mat),1)	((rat,bat),1) ((mat,pat),1) ((mat,bat),1) ((pat,bat),2) ((pat,cat),1) ((pat,mat),1) ((mat,rat),1) ((mat,cat),1) ((rat,cat),1) ((rat,bat),1)
SORT & COMBINE		

Reducer Output
Sorting rule:
 	
class Pair implements
Comparable<Pair> {
String a, b;
int compareTo(Pair p) {
int k = a.compareTo(p.a)
if(k==0) k=b.compareTo(p.b)
return k;
}
}
	((bat,cat), [1])
	((bat,mat), [2])
	((bat,pat), [1,1])
	((bat,rat), [1,1])
	
	((cat,bat), [2,3])
	((cat,mat), [1,1])
	((cat,pat), [1])
	((cat,rat), [2,2])
		((mat,bat), [1])
		((mat,cat), [1])
		((mat,pat), [1])
		((mat,rat), [1])
		
		((pat,bat), [2])
		((pat,cat), [1])
		((pat,mat), [1])
		
		((rat,bat), [1,1])
		((rat,cat), [1])

REDUCE	**Reducer-1**	**Reducer-2**
Reducer Output	((bat,cat), 1) ((bat,mat), 2) ((bat,pat), 2) ((bat,rat), 2) ((cat,bat), 5) ((cat,mat), 2)	((mat,bat), 1) ((mat,cat), 1) ((mat,pat), 1) ((mat,rat), 1) ((pat,bat), 2)

	((cat,pat), 1) ((cat,rat), 4)	((pat,cat), 1) ((pat,mat), 1) ((rat,bat), 2) ((rat,cat), 1)