Homework 7

```
val accounts_file = sc.textFile("/user/lc3397/loudacre/accounts/*")
```

```
// accounts_file: org.apache.spark.rdd.RDD[String] = /user/lc3397/loudacre/ accounts/* MapPartitionsRDD[1] at textFile at :27
```

```
accounts_file.take(5).foreach(println)
```

```
// 1,2008-10-23 16:05:05.0,\N,Donald,Becton,2275 Washburn Street,Oakland,CA,94660,5100032418,2014-03-18 13:29:47.0,2014-03-18 13:29:47.0
// 2,2008-11-12 03:00:01.0,\N,Donna,Jones,3885 Elliott Street,San Francisco,CA,94171,4150835799,2014-03-18 13:29:47.0,2014-03-18 13:29:47.0
// 3,2008-12-21 09:19:50.0,\N,Dorthy,Chalmers,4073 Whaley Lane,San Mateo,CA,94479,6506877757,2014-03-18 13:29:47.0,2014-03-18 13:29:47.0
// 4,2008-11-28 00:08:09.0,\N,Leila,Spencer,1447 Ross Street,San Mateo,CA,94444,6503198619,2014-03-18 13:29:47.0,2014-03-18 13:29:47.0
// 5,2008-11-15 23:06:06.0,\N,Anita,Laughlin,2767 Hill Street,Richmond,CA,94872,5107754354,2014-03-18 13:29:47.0,2014-03-18 13:29:47.0
// [userid],[creation date],[firstname],[lastname]
```

```
val weblog = sc.textFile("/user/lc3397/loudacre/weblog/2014-03-15.log")
```

```
// weblog: org.apache.spark.rdd.RDD[String] = /user/lc3397/loudacre/ weblog/2014-03-15.log MapPartitionsRDD[3] at textFile at :27
```

```
weblog.take(5).foreach(println)
```

```
// 234.206.18.239 - 8495 [15/Mar/2014:23:59:30 +0100] "GET /KBDOC-00082.html HTTP/1.0" 200 9054 "http://www.loudacre.com" "Loudacre Mobile Browser Titanic 2200"
// 234.206.18.239 - 8495 [15/Mar/2014:23:59:30 +0100] "GET /theme.css HTTP/1.0" 200 4552 "http://www.loudacre.com" "Loudacre Mobile Browser Titanic 2200"
// 104.213.2.248 - 22676 [15/Mar/2014:23:58:52 +0100] "GET /KBDOC-00086.html HTTP/1.0" 200 11413 "http://www.loudacre.com" "Loudacre Mobile Browser Ronin Novelty Note 1"
// 104.213.2.248 - 22676 [15/Mar/2014:23:58:52 +0100] "GET /theme.css HTTP/1.0" 200 14482 "http://www.loudacre.com" "Loudacre Mobile Browser Ronin Novelty Note 1"
// 151.200.170.131 - 126729 [15/Mar/2014:23:58:21 +0100] "GET / KBDOC-00165.html HTTP/1.0" 200 9353 "http://www.loudacre.com" "Loudacre Mobile Browser Titanic 4000"
```

```
val userData = accounts_file.map(s => {
val tokens = s.split(",")
(tokens(0), Array(tokens(1),tokens(2),tokens(3),tokens(4),tokens(5),tokens(6)
,tokens(7),tokens(8),tokens(9),tokens(10),tokens(11)))
})
```

```
val joined_result = (userData join weblog_reduced)
```

```
joined_result.take(5).foreach(println)
```

```
// (178,([Ljava.lang.String;@15b41f6a,12))
// (20758,([Ljava.lang.String;@647a7347,4))
// (110253,([Ljava.lang.String;@23266dda,2))
// (990,([Ljava.lang.String;@2ff6153d,2))
// (100870,([Ljava.lang.String;@44ec7064,6))
```

```
// (178,(Array(2008-12-09 12:09:14.0, \N, Kimberly, Mulder, 2383 Patton Lane, San Francisco, CA, 94114, 4150916606,
2014-03-18 13:29:47.0, 2014-03-18 13:29:47.0),12)),
// (110253,(Array(2013-11-04 15:43:44.0, 2014-01-31 10:32:08.0, Larry, Anthony, 3691 Woodland Drive, Oakland, CA,
94529, 5107848633, 2014-03-18 13:33:14.0, 2014-03-18 13:33:14.0),2)),
// (20758,(Array(2011-02-02 19:20:56.0, 2012-11-04 00:26:25.0, Antonio, Lott, 3701 Layman Court, San Francisco, CA,
94005, 4156269937, 2014-03-18 13:30:25.0, 2014-03-18 13:30:25.0),4)),
// (990,(Array(2009-12-27 10:36:48.0, 2014-01-09 09:53:13.0, James, Vargas, 386 Ingram Road, Sacramento, CA,
95766, 9169661747, 2014-03-18 13:29:49.0, 2014-03-18 13:29:49.0),2)),
```

joined_result.take(5).map(s => (s._1, s._2._2, s._2.*1(2), s.* 2._1(3))).foreach(println)

```
// (178,12,Kimberly,Mulder)
// (20758,4,Antonio,Lott)
// (110253,2,Larry,Anthony)
// (990,2,James,Vargas)
// (100870,6,John,Benfield)
```

joined_result.take(5).map(s => s._1.toString() + ' ' + s._2._2.toString() + ' ' + s._2._1(2) + ' ' + s._2._1(3)).foreach(println)

```
// 178 12 Kimberly Mulder
// 20758 4 Antonio Lott
// 110253 2 Larry Anthony
// 990 2 James Vargas
// 100870 6 John Benfield
```

val accounts_in_array = accounts_file.map(s => s.split(','))

```
// accounts_in_array: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[14] at map at :29
```

val postal_code_key_rdd = accounts_in_array.keyBy{a => a(8)}

```
// postal_code_key_rdd: org.apache.spark.rdd.RDD[(String, Array[String])] = MapPartitionsRDD[15] at keyBy at :31
```

postal_code_key_rdd.take(3)

```
// res35: Array[(String, Array[String])] = Array(
// (94660,Array(1, 2008-10-23 16:05:05.0, \N, Donald, Becton, 2275 Washburn Street, Oakland, CA, 94660,
5100032418, 2014-03-18 13:29:47.0, 2014-03-18 13:29:47.0)),
// (94171,Array(2, 2008-11-12 03:00:01.0, \N, Donna, Jones, 3885 Elliott Street, San Francisco, CA, 94171,
4150835799, 2014-03-18 13:29:47.0, 2014-03-18 13:29:47.0)),
// (94479,Array(3, 2008-12-21 09:19:50.0, \N, Dorthy, Chalmers, 4073 Whaley Lane, San Mateo, CA, 94479,
6506877757, 2014-03-18 13:29:47.0, 2014-03-18 13:29:47.0))
// )
```

val postal_code_key_names_value = postal_code_key_rdd.groupByKey().mapValues(s => (s.map(a => (a(3),a(4)))))

```
// postal_code_key_names_value: org.apache.spark.rdd.RDD[(String, Iterable[( String, String)])] = MapPartitionsRDD[27]
at mapValues at :33
```

val postalCode_pair_rdd_sorted = postal_code_key_names_value.sortBy(_._1)

```
// postalCode_pair_rdd_sorted: org.apache.spark.rdd.RDD[(String, Iterable[( String, String)])] = MapPartitionsRDD[37] at
sortBy at :35
```

```
val postal_result = postalCode_pair_rdd_sorted.map(s => ("---" + s._1 + "\n", s._2.map(name => name._1 + "," + name._2 + "\n")))
```

```
postal_result.take(5).foreach(s => (print(s._1), s._2.foreach(print)))
```

```
// ---85000
// Bailey,Sewell
// Daniel,Marin
// Harvey,Allen
// Daniel,Prinz
// Robert,Pascale
// Donna,Brookes
// James,Mackenzie
// Robert,Chamberlain
// Richard,Cunningham
// ---85001
// Issac,Lance
// Vesta,Barnes
// Eva,Fiore
// Keith,Tucker
// Danielle,Medford
// Norman,Spell
// Shelley,Soto
// Kathy,Frantz
// Timothy,Wilkins
// Joseph,Snyder
// Delbert,Flores
// Gail,Eakes
// Bert,Daniels
// Vincent,Carpenter
// Frances,Mendelsohn
// Mary,Watson
// Donald,Brookover
// Brandon,Hathaway
// Crystal,Leonard
// Carrie,Moran
// Marie,Kirksey
// ---85002
// Estella,Baird
// James,Gilbert
// David,McKay
// Laura,Clark
// John,Horn
// Ruby,Whitney
// David,Perry
// Marianne,James
// Nancy,Holiman
// Allen,Roman
// Donna,Manus
// Nancy,Reed
// Jessica,Payne
// Bryant,Stewart
// Jose,Jones
// Wesley,Robinson
// ---85003
// Kevin,Dvorak
// Virginia,Wisniewski
// Mark,Martin
// Catherine,Gibson
// Lindsey,Thies
// Vivian,Ross
// Harry,Tabor
// Kyle,Strickland
// ---85004
// Mary,Kitts
// Kevin,Viola
// Tonya,Meadows
// Sherry,Royalty
// Greg,Collins
// Joseph,Shirley
// Sandra,White
// Timothy,Stern
// Dominic,Johnson
// Mary,Dewitt
// Matthew,Carpenter
// Annie,Ball
// Kathleen,Pate
// ---85000
// Bailey,Sewell
// Daniel,Marin
// Harvey,Allen
// Daniel,Prinz
// Robert,Pascale
// Donna,Brookes
// James,Mackenzie
// Robert,Chamberlain
// Richard,Cunningham
// ---85001
// Issac,Lance
// Vesta,Barnes
// Eva,Fiore
// Keith,Tucker
// Danielle,Medford
// Norman,Spell
// Shelley,Soto
// Kathy,Frantz
// Timothy,Wilkins
// Joseph,Snyder
// Delbert,Flores
// Gail,Eakes
// Bert,Daniels
// Vincent,Carpenter
// Frances,Mendelsohn
// Mary,Watson
// Donald,Brookover
// Brandon,Hathaway
// Crystal,Leonard
// Carrie,Moran
// Marie,Kirksey
// ---85002
// Estella,Baird
// James,Gilbert
// David,McKay
// Laura,Clark
// John,Horn
// Ruby,Whitney
// David,Perry
// Marianne,James
// Nancy,Holiman
// Allen,Roman
// Donna,Manus
// Nancy,Reed
// Jessica,Payne
// Bryant,Stewart
// Jose,Jones
// Wesley,Robinson
// ---85003
// Kevin,Dvorak
// Virginia,Wisniewski
// Mark,Martin
// Catherine,Gibson
// Lindsey,Thies
// Vivian,Ross
// Harry,Tabor
// Kyle,Strickland
// ---85004
// Mary,Kitts
// Kevin,Viola
// Tonya,Meadows
// Sherry,Royalty
// Greg,Collins
// Joseph,Shirley
// Sandra,White
// Timothy,Stern
// Dominic,Johnson
// Mary,Dewitt
// Matthew,Carpenter
// Annie,Ball
// Kathleen,Pate
// Carrie,Lish// Carrie,Lish
```