CSE 587 Data Intensive Computing

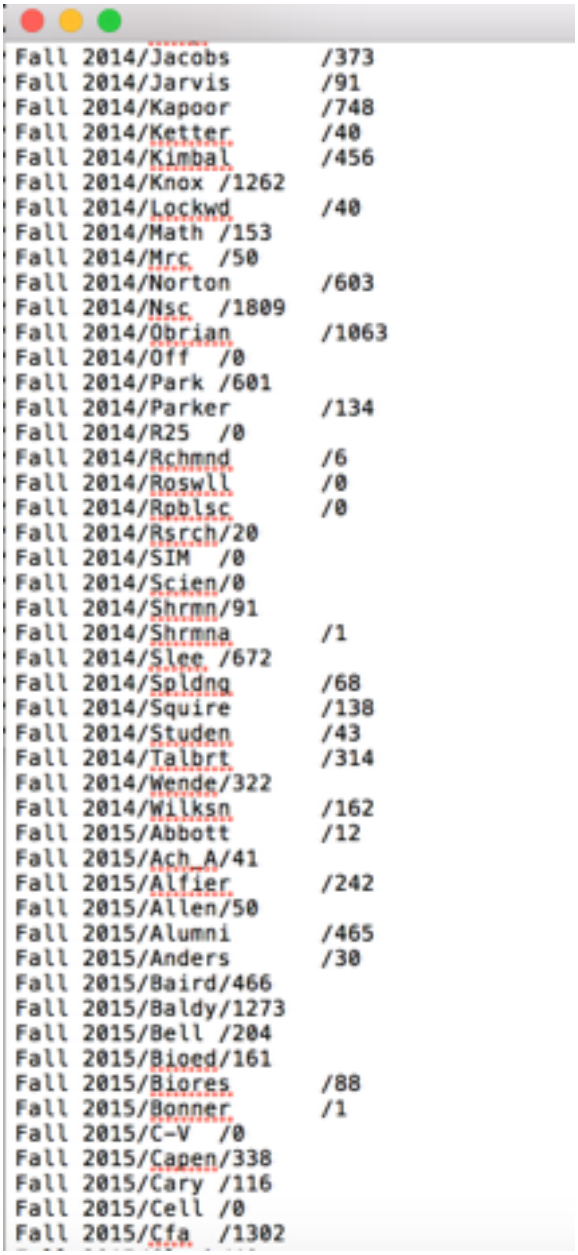
Project 2

Parallel Processing Using Hadoop Map Reduce

Falguni Bharadwaj - 50163471
Malavika Reddy Tappeta - 50169248

Introduction:
We have designed 20 questions which help in analyzing the dataset and whose answers give some details that will make class scheduling easier for next semesters. We used one mapper & reducer to solve each question.

Question 1:
What is the total seating capacity of each building semester wise?



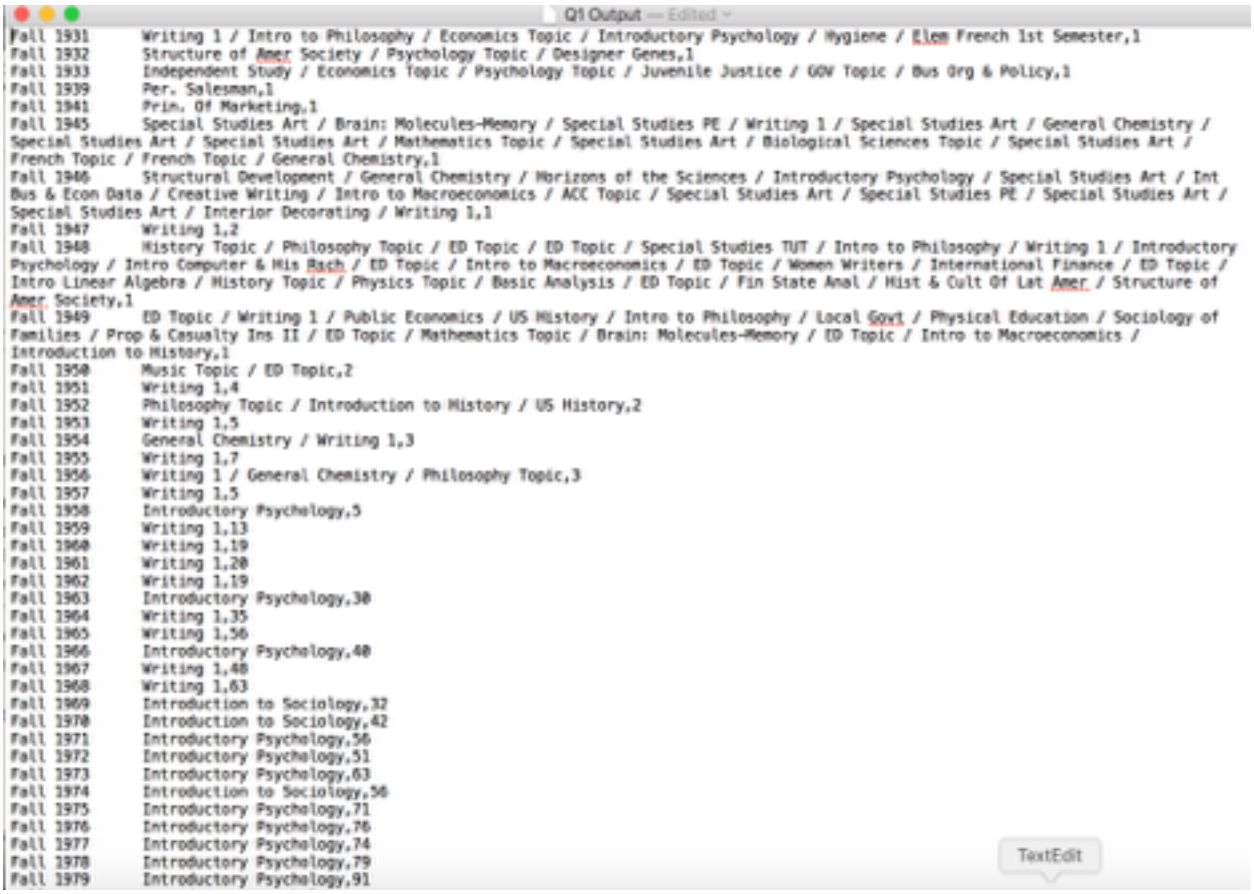
For this problem, the mapper splits building from room number. Room number passes as value (along with total capacity) and building as key. Output of mapper is <Key = Semester / Building Name, Value = Room Number / Total Capacity> where "/" is being used as a separator. Thus in reducer, if a room number is being repeated, its capacity is taken only once and added with others from same building. Thus final output contains total capacity of each building. Output of reducer is <Key = Semester/ Building Name, Value = /Total Capacity>.

From the output we can see that some buildings have a lot of lecture halls like "Nsc", "Baldy", "Obrian" whereas some hardly have classrooms like "Ketter" or "Allen". So more classes will be scheduled in these buildings. We can also observe that some are reserved for "Rsrch" and have no lecture halls. National Sciences Complex Nsc has the largest seating capacity of more than 1800 seats.

Question 2:
Which course has maximum number of students each semester?

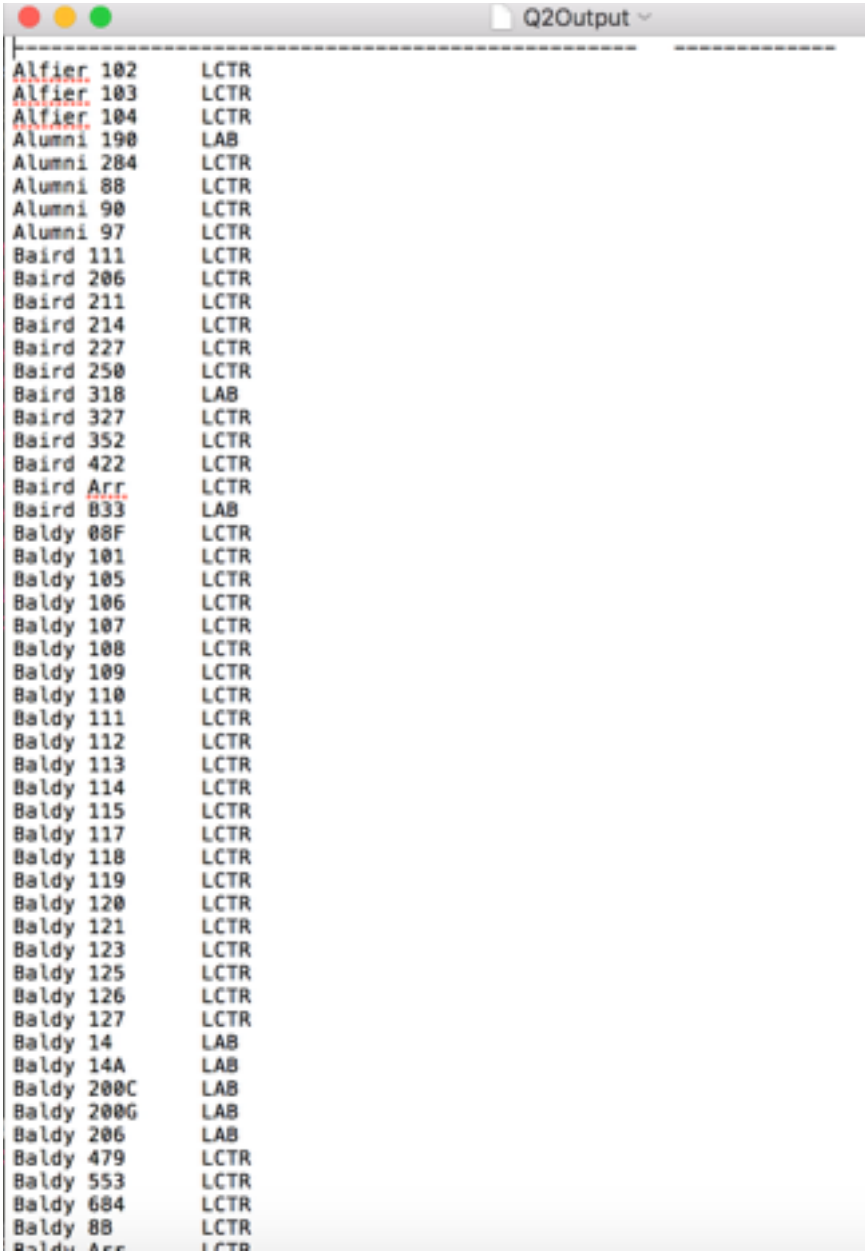
For this question, our Mapper output was of the form <Key = Semester Year, Value = (Course Name, Total Students enrolled)>. The reducer split the value in terms of total number of students enrolled and for each semester it calculated maximum value of total students. If there are more than one course with the same number of students as the maximum number, then they are appended to the value field of the reducer. The final output of the reducer has <Key = Semester Year, Value = (Course Name, Maximum Students>. Following is a sample output where as we can see for Fall 1931, maximum students according to the dataset provided is 1 for courses "Writing 1","Intro. to Philosophy", "Economics Topic" etc. Hence the output has the courses separated by a "/ " followed by the maximum number of students for these courses at the end. Some courses like Fall 1963 has only one course with maximum students which is Introductory Psychology.

Looking at the output we can infer that in the early years from around 1971-1998, "Introductory Psychology" was a popular course choice. Earlier than that "Writing 1" had the most number of students. In the recent years "Corporation Finance" has become more popular and has maximum students of around 500 for the past couple of years. Another popular course over the summer is "Gross Human Anatomy" with around 150-200 students every year. We can say that in earlier years courses like "Writing 1" were more popular but recently science and management courses have started being in demand. We also realize from the data that maximum number of students in the earlier years used to be extremely less compared to recent years. For example, maximum number of students for Fall 1955 were 7 whereas for Fall 2014 were 574. So we can say that number of students attending university has drastically increased over the decades. On an average, "Introductory Psychology" seems like a popular choice for students throughout the years.

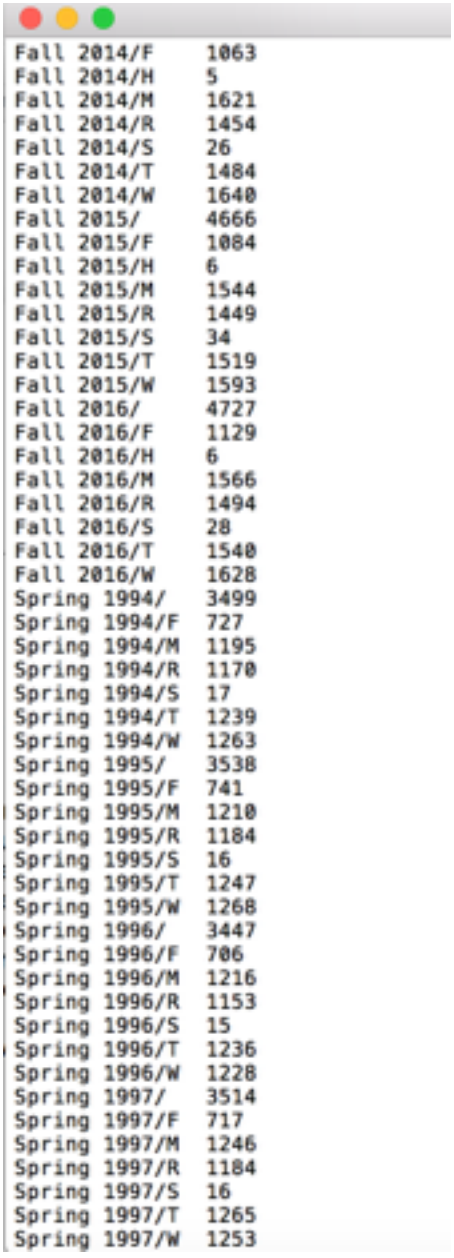


Question 3:
Which rooms were used for lectures and which were used for labs or otherwise reserved?

We use the new tsv dataset for this problem. So the description contains whether the room is used for lab or lecture. In this, our output is of the form <Key = Room Number, Value = Description>. The output gives us a very clear idea of which rooms are Labs and which are Lecture Halls. This information can be further used to schedule extra lab sessions of extra lectures for next semesters. We noticed that not every building has a Lab in it whereas buildings like Baldy & Bioed had more than 4 Labs. We also noticed some halls like Farber 180 or Kimbal 720 which were "Reserved".



Question 4:
Which day was busiest each semester?

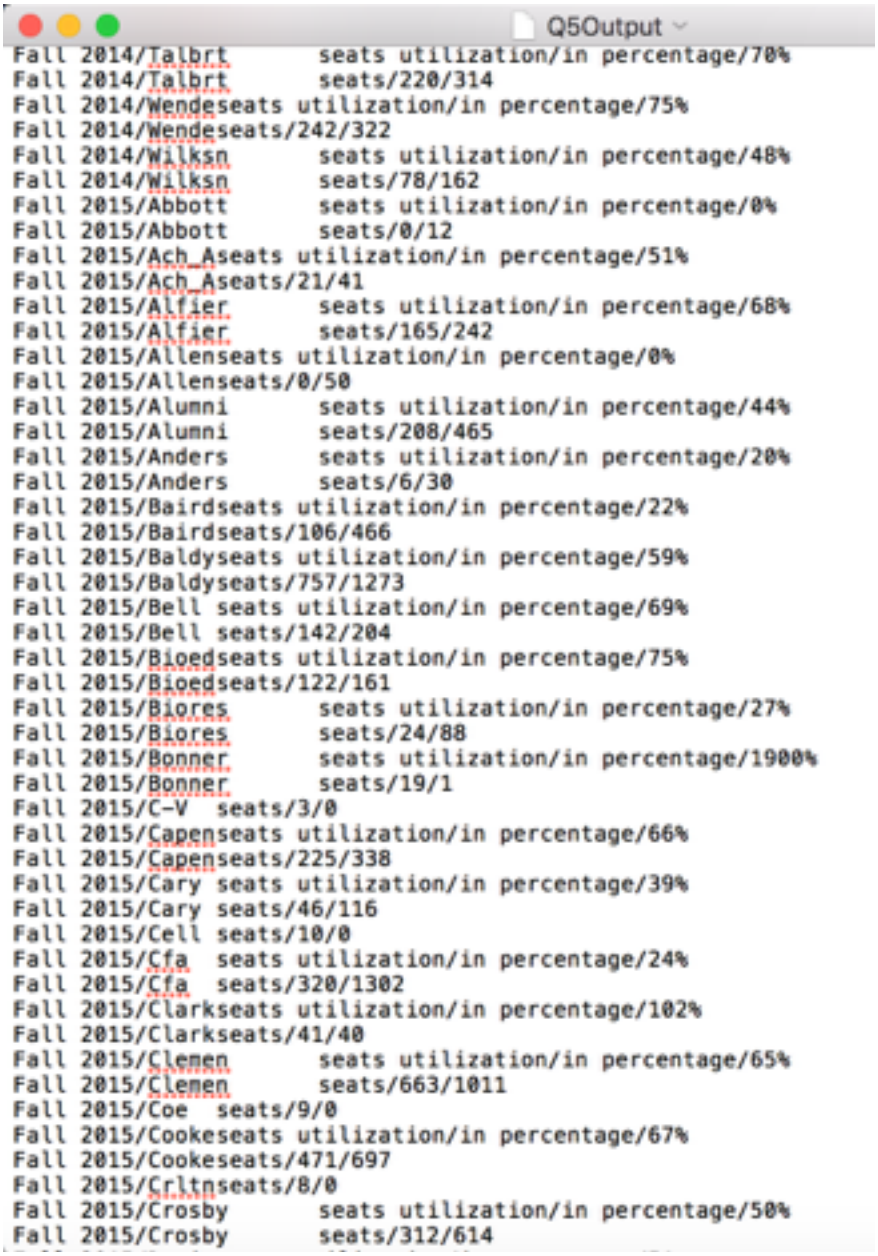


```
Fall 2014/F     1063
Fall 2014/H     5
Fall 2014/M     1621
Fall 2014/R     1454
Fall 2014/S     26
Fall 2014/T     1484
Fall 2014/W     1640
Fall 2015/      4666
Fall 2015/F     1084
Fall 2015/H     6
Fall 2015/M     1544
Fall 2015/R     1449
Fall 2015/S     34
Fall 2015/T     1519
Fall 2015/W     1593
Fall 2016/      4727
Fall 2016/F     1129
Fall 2016/H     6
Fall 2016/M     1566
Fall 2016/R     1494
Fall 2016/S     28
Fall 2016/T     1540
Fall 2016/W     1628
Spring 1994/    3499
Spring 1994/F   727
Spring 1994/M   1195
Spring 1994/R   1170
Spring 1994/S   17
Spring 1994/T   1239
Spring 1994/W   1263
Spring 1995/    3538
Spring 1995/F   741
Spring 1995/M   1210
Spring 1995/R   1184
Spring 1995/S   16
Spring 1995/T   1247
Spring 1995/W   1268
Spring 1996/    3447
Spring 1996/F   706
Spring 1996/M   1216
Spring 1996/R   1153
Spring 1996/S   15
Spring 1996/T   1236
Spring 1996/W   1228
Spring 1997/    3514
Spring 1997/F   717
Spring 1997/M   1246
Spring 1997/R   1184
Spring 1997/S   16
Spring 1997/T   1265
Spring 1997/W   1253
```

For this problem, in the mapper function we have split each combination of days to individual days (for ex. MWF as M,W,F) and then we passed it to reducer which collected how many times each day was being passed (sum of values of each day which corresponds to how many courses were scheduled each day). The output shows how busy each day was every semester in terms of hours. We infer from the data that generally Monday, Tuesday, Wednesday and Thursday have similar values and Friday has lesser than them. Saturday and Sunday on the other hand hardly have classes.

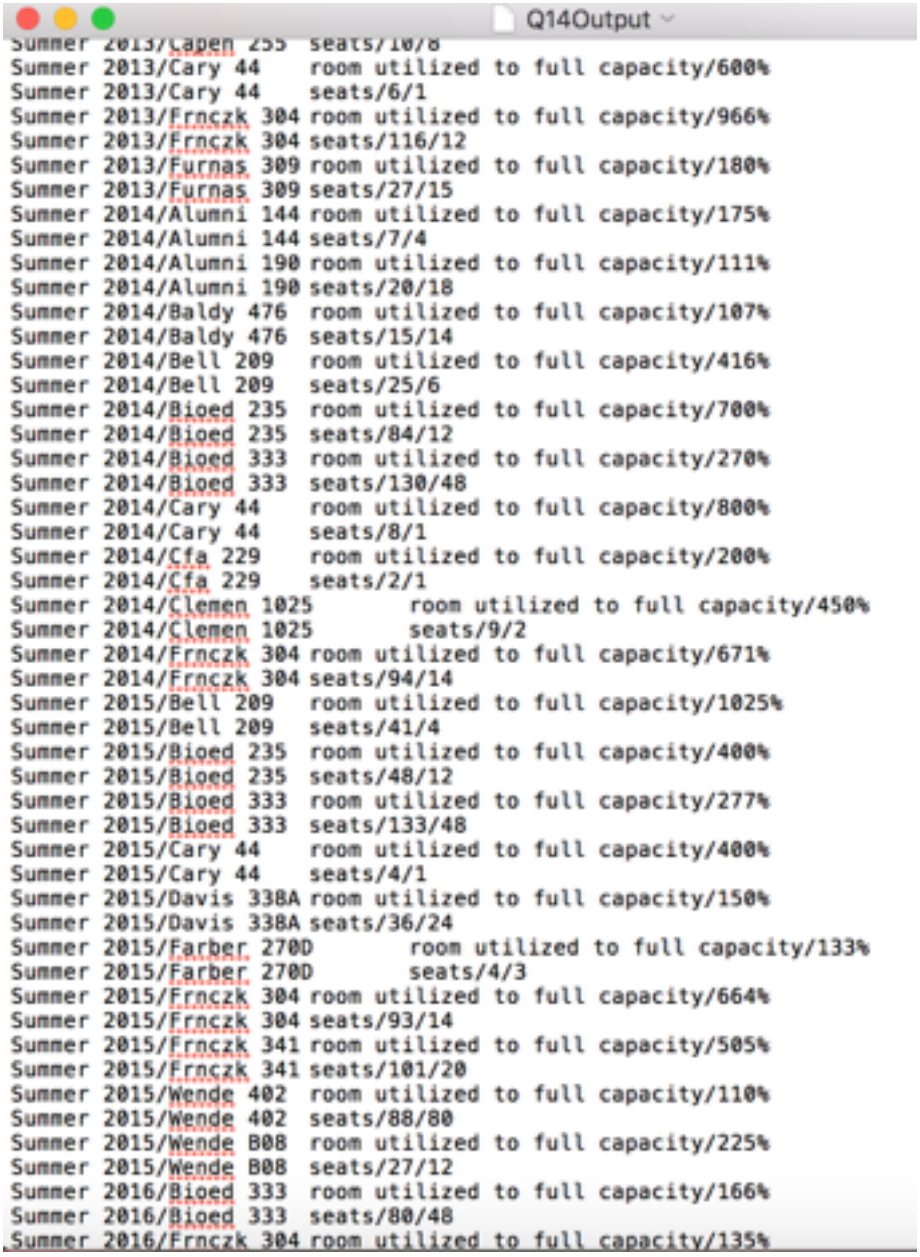
Wednesday being in the middle of the week seems to be the busiest day for the overall data.

Question 5:
Which building has the largest seat utilization?



For this problem, the mapper is sending total capacity of room along with number of students enrolled for that course to reducer. Then reducer is calculating seat utilization and displaying percentage along with the actual values. Output of mapper is <Key = Semester/Building Name, Value = Room Number/ Number of enrolled students/total capacity off room> where "/" is used as a separator.
The output of dataset shows mostly buildings are not used to their full capacity but sometimes maybe due to error in dataset, number of students enrolled is greater than room capacity and results in more than 100% seat utilization. Also seat utilization is not an accurate indicator of capacity of each building. For example, Bonner Hall has 1900% seat utilization in Fall 2015 and Baldy hall has 75% but still number of seats in Bonner Hall is just 1 whereas it is 1273 in Baldy Hall.

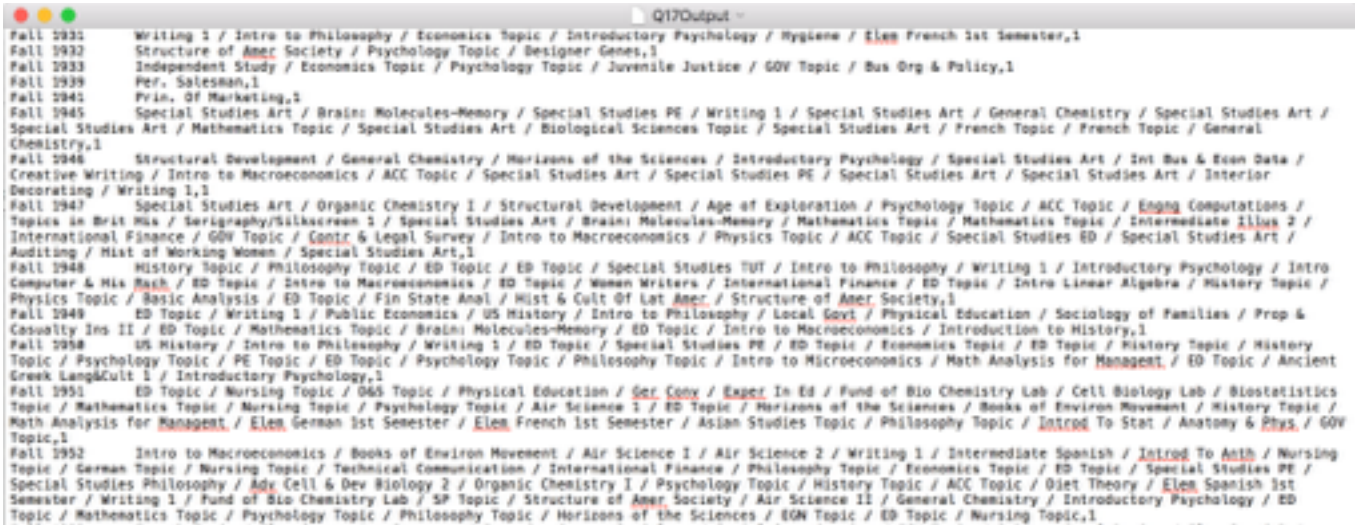
Question 6:
Which rooms were used to their full capacity?



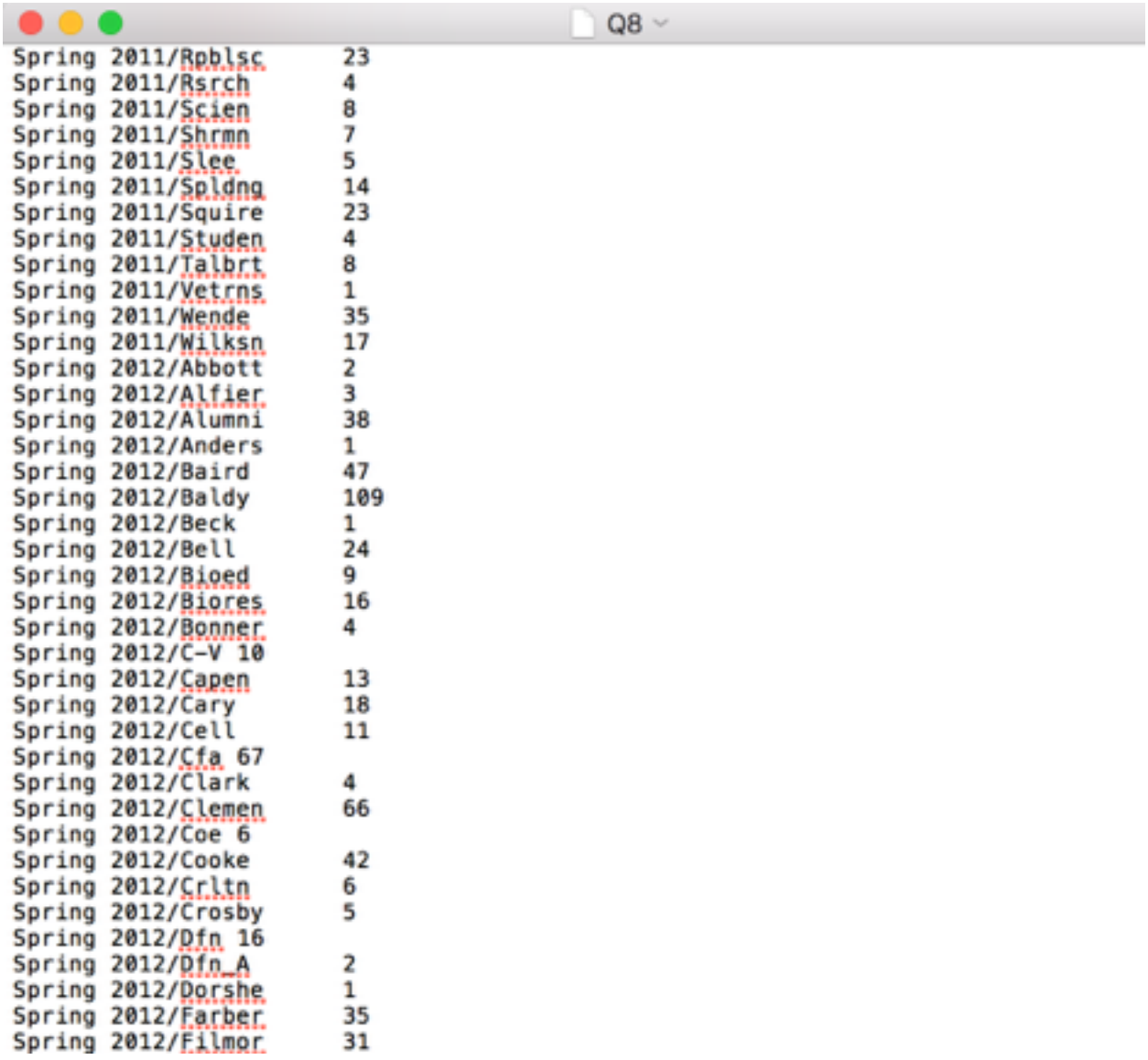
For this problem, the mapper sends each room as key to reducer and reducer calculates if the room is being used to its full capacity. Output of mapper is <Key = Semester/Hall Room Number, Value = Number of Students enrolled/ Total Number of seats> where "/" is being used as a separator. Output of reducer is two rows: one with number of students enrolled and total capacity of room, and second with percentage value of room used.
This is a good indicator of which room is over utilized. We can see that in recent years, rooms from Bioed, Frnczk Hall etc are being fully used. Using this information to re assign rooms for courses with higher number of students is also a good idea.

Question 7:
Which courses had least enrollment?

The output shows Key as each semester and Value as all the courses with minimum enrollment of students. As we can see usually every semester has some courses where only 1 student has enrolled. As we go through the output we also notice that there are many courses which have 0 students. This is a good indication of which courses are least popular.



Question 8:
How many rooms are there in each building?
The following output shows how many rooms were used for classes for every building each semester. This will give a good idea about which building can fit more students. From the data we see that Baldy has the most number of rooms whereas "Anders" & "BUTLER" Halls have only one room for courses, which might mean that they are administrative buildings or library etc.

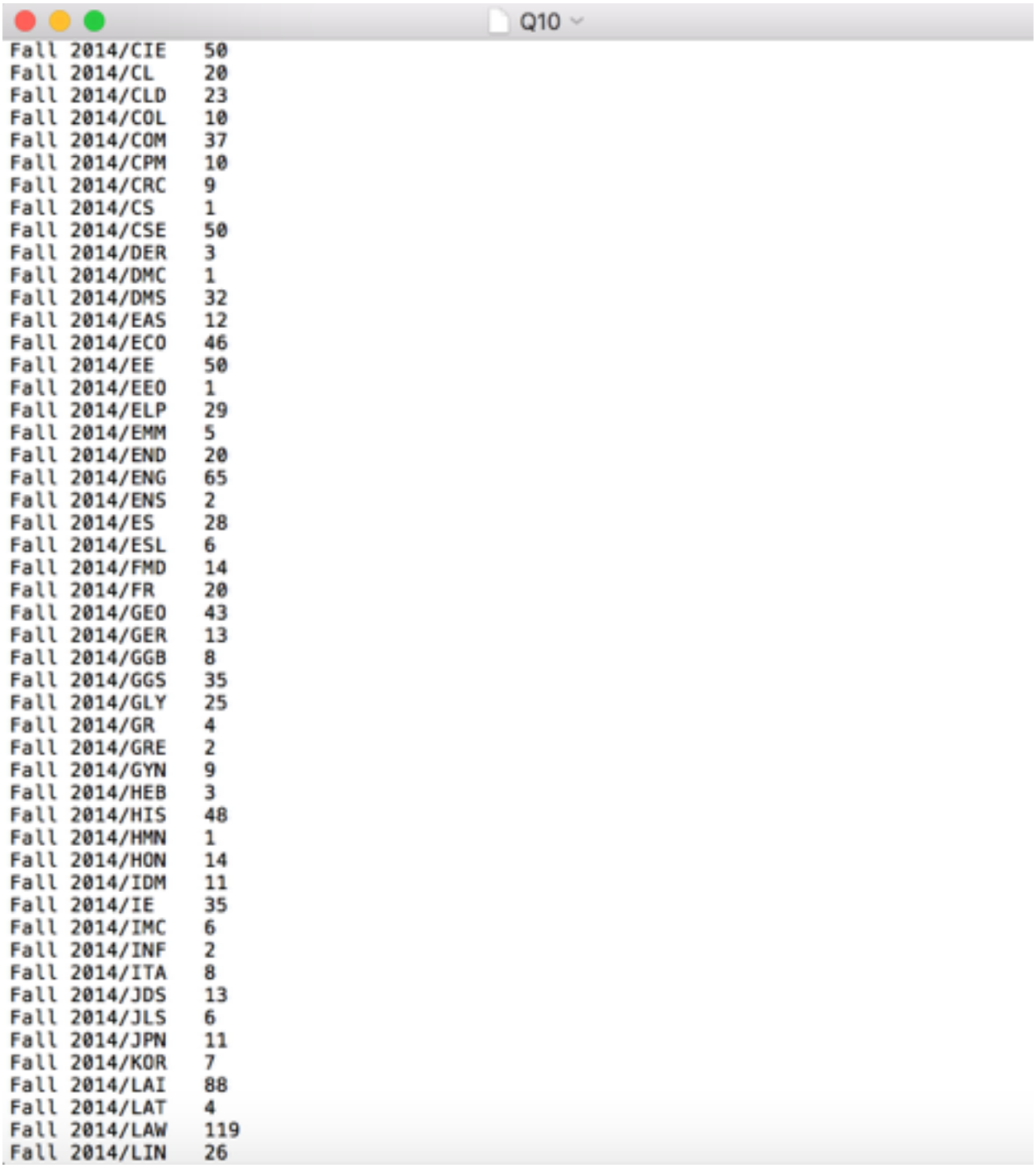


Question 9:

Over the years how many different rooms has each course been taught in?

For this problem, the mapper output is <Key = Course Name, Value = Room Number> The reducer adds all the rooms where the course has been taught and returns that as the value. As seen below from the sample output gives a good account of where each course has been taken. This can help in reassigning rooms for coming semesters. If there is a time clash it can be moved to one of it's previous rooms by easily looking at these values. From the output we can infer that most Accounting course classes happen at Jacobs Hall, Adv Physical Anthropology takes place at Spldng Building and so on. Using this data we can also infer if some building is specific to a department or not. For example since allDental classes happen at Squire we can say that Squire is the Dental department building.

Question 10:
How many courses does each department have?
For this problem the output of mapper is <Key = Semester/Department, Value = Course>. Reducer calculates how many courses are there in a department and its output is <Key = Semester/Department, Value = Number of Courses>. This is a good measure to see if any new courses are being offered compared to previous years. We can see that for Fall 2014, DMS department offered 32 courses, EE offered 50 courses while management offered 16 courses. There was a decrease of 5 in the number of courses offered by CSE department in 2013 vs. 2014.



Question 11:
What are the total number of courses offered each semester?

```
Fall_1931    6
Fall_1932    3
Fall_1933    6
Fall_1939    1
Fall_1941    1
Fall_1945    4
Fall_1946    9
Fall_1947    9
Fall_1948    9
Fall_1949    15
Fall_1950    22
Fall_1951    35
Fall_1952    10
Fall_1953    10
Fall_1954    24
Fall_1955    56
Fall_1956    36
Fall_1957    72
Fall_1958    38
Fall_1959    25
Fall_1960    52
Fall_1961    66
Fall_1962    37
Fall_1963    28
Fall_1964    47
Fall_1965    57
Fall_1966    102
Fall_1967    214
Fall_1968    155
Fall_1969    168
Fall_1970    108
Fall_1971    164
Fall_1972    159
Fall_1973    74
Fall_1974    151
Fall_1975    107
Fall_1976    158
Fall_1977    163
Fall_1978    49
Fall_1979    8
Fall_1980    71
Fall_1981    8
Fall_1982    245
Fall_1983    151
Fall_1984    151
Fall_1985    169
Fall_1986    10
Fall_1987    178
Fall_1988    32
Fall_1989    166
Fall_1990    21
Fall_1991    187
```

In this problem we start by creating the key for each row of data in the 'Semester_Year' format by using the second field of the data. Next, we assign the Course Id of each row as the value and this <key,value> pair is the output of the mapper. In the reducer we keep a string 's' which contains the course ids. For every <key,value> pair we first check if the value is already present in the 's' , if it isn't then we increment the counter by 1. This way we are only counting for courses with unique course ids. At the end we append a check value to the result so that for the next round in the reducer it directly prints the output.

In the output we have the semester and number of unique courses offered for that semester. From the output we can infer that the number of courses offered were lesser in the earlier years.

Question 12:
Which hall is used the most each semester?

```
Fall_1993       Clemen
Fall_1994       Clemen
Fall_1995       Clemen
Fall_1996       Clemen
Fall_1997       Clemen
Fall_1998       Clemen
Fall_1999       Baldy
Fall_2000       Baldy
Fall_2001       Baldy
Fall_2002       Baldy
Fall_2003       Baldy
Fall_2004       Baldy
Fall_2005       Baldy
Fall_2006       Baldy
Fall_2007       Baldy
Fall_2008       Clemen
Fall_2009       Clemen
Fall_2010       Baldy
Fall_2011       Baldy
Fall_2012       Baldy
Fall_2013       Clemen
Fall_2014       Park
Fall_2015       Park
Fall_2016       Park
Spring_1994     Clemen
Spring_1995     Clemen
Spring_1996     Clemen
Spring_1997     Clemen
Spring_1998     Clemen
Spring_1999     Clemen
Spring_2000     Clemen
Spring_2001     Clemen
Spring_2002     Clemen
Spring_2003     Baldy
Spring_2004     Baldy
Spring_2005     Clemen
Spring_2006     Clemen
Spring_2007     Clemen
Spring_2008     Clemen
Spring_2009     Clemen
Spring_2010     Clemen
Spring_2011     Clemen
Spring_2012     Baldy
Spring_2013     Baldy
Spring_2014     Baldy
Spring_2015     Baldy
Spring_2016     Baldy
Summer_1994     Clemen
Summer_1995     Clemen
Summer_1996     Nsc
Summer_1997     Nsc
Summer_1998     Clemen
Summer_1999     Clemen
Summer_2000     Clemen
Summer_2001     Clemen
Summer_2002     Clemen
Summer_2003     Baldy
Summer_2004     Baldy
Summer_2005     Baldy
Summer_2006     Baldy
```

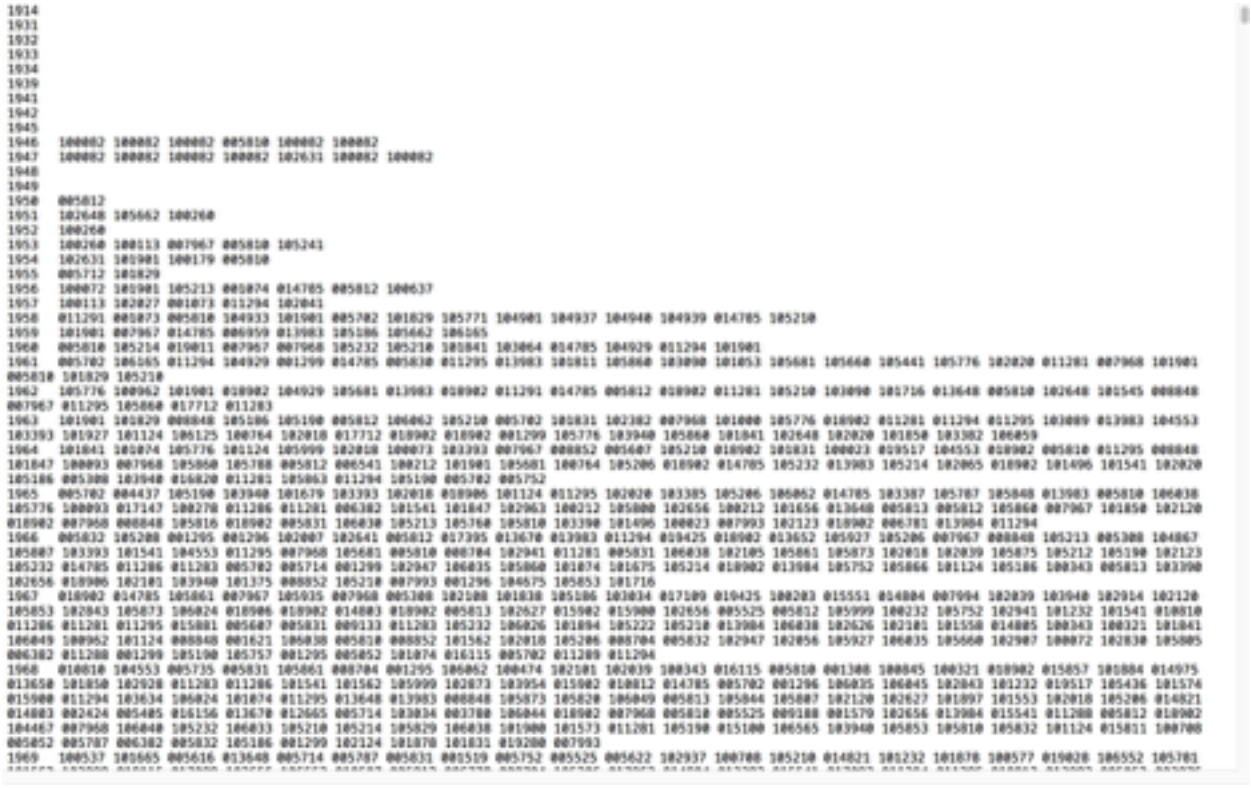
For this problem also we create <key,value> pairs where key contains 'Semester_Year' and value contains the name of the hall in each row. We check if the field for the hall name contains 'Unknown' or 'Arr Arr', if so we ignore rows containing these values. In the reducer we make use of a hashmap. The key in the hashmap is the name of the hall and the value is the number of times it appears in the data for each semester. For each <key,value> pair the reducer first checks if the the value is already present in the hash map, if so it increments the value of the hashmap for that key by 1. If it is not present then it adds a key to the hashmap where key is the hall name and the value is set to 1. Next, we use Collections.max() and look for the highest value in the hashmap and set max to it. Then we iterate over the hash map and look for values equal to max value and print out the key wherever we find the match.

The output contains the semester and the name of the hall that is used the most in that semester. If there is more than one hall then all are mentioned like in the case of semester Winter_2017 we get Math and Cary as the halls used the most in this semester. From the output we can see that Clemen and Baldy seem the most popular halls and this could be because there are more classrooms in these halls.

Question 13:
Which courses are offered in both fall and spring semester each year?

Here, we first create <key,value> pairs where key contains the year and value is in the format 'Semester-CourseId' and this is the output of the mapper. In the reducer for each <key,value> pair we first split the value by using the separator "-" . We keep two local strings, one for fall and one for spring. To these we keep adding the unique course id present in the value. We check if any of the split strings contain the string 'fall' or 'spring'. If it contains fall then we check if that course id is present in the local string spring , if it is then we add this course id to the global string 's' and then append the course id to fall. If it contains spring then we check if that course id is present in the local string fall , if it is then we add this course id to the global string 's' and then append the course id to spring. Finally, after all the values are iterated over the key is set to the year and value is set to the string 's' and this is written. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

The output contains the year and the list of courses that are offered in both fall and spring semester that year. In the output we can see that the number of common courses increases over the years. Some years don't have any courses in common.



Question 14:
What is the total enrollment in each semester?

```
Fall_1931	6
Fall_1932	3
Fall_1933	6
Fall_1939	1
Fall_1941	1
Fall_1945	15
Fall_1946	15
Fall_1947	27
Fall_1948	23
Fall_1949	15
Fall_1950	24
Fall_1951	48
Fall_1952	45
Fall_1953	58
Fall_1954	46
Fall_1955	79
Fall_1956	75
Fall_1957	97
Fall_1958	85
Fall_1959	165
Fall_1960	239
Fall_1961	303
Fall_1962	375
Fall_1963	648
Fall_1964	812
Fall_1965	1043
Fall_1966	1127
Fall_1967	1224
Fall_1968	1518
Fall_1969	1584
Fall_1970	1957
Fall_1971	2112
Fall_1972	2484
Fall_1973	2735
Fall_1974	3003
Fall_1975	3387
Fall_1976	3335
Fall_1977	3777
Fall_1978	4297
Fall_1979	5068
Fall_1980	5965
Fall_1981	6664
Fall_1982	7296
Fall_1983	8634
Fall_1984	11459
Fall_1985	13679
Fall_1986	17384
Fall_1987	21430
Fall_1988	28978
Fall_1989	38402
Fall_1990	53947
Fall_1991	70861
```

For this problem we create <key,value> pairs where the key is in the 'Semester_year' format and the value is the total enrollment in each course/row for that semester. This is the output of the mapper. In the reducer for each <key,value> pair we take the value and covert it to an integer value(we send it as Text from mapper to reducer) using Integer.parseInt() and this value is added to an integer variable 'sum'. At the end we do a write operation for <key,value> pair where key is the 'Semester_year' and value is the sum(converted to Text). We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

The output file contains the Semester and the total enrollment next to it. From the output we can infer that the total enrollment is less in the earlier years and increases over the years. This is also due to the fact that the number of courses offered increased over the years.

Question 15:
What was the busiest part of the day during exams in each semester?

```
Fall_2011    Morning
Fall_2012    Afternoon
Fall_2013    Afternoon
Fall_2014    Afternoon
Fall_2015    Afternoon
Spring_2012  Afternoon
Spring_2013  Evening
Spring_2014  Evening
Spring_2015  Evening
Spring_2016  Evening
```

In this problem we create <key,value> pairs for each row where key is set to the 'Semester_Year' and value is set to the start time of the exam. What we do here is we split the start time using the separator ":" and the first part of the string is set as the value and sent to the reducer. This is the output of the mapper. In the reducer for each value we get the integer value using Integer.parseInt() and then check for three conditions. If the value is less than 12 then we increment the counter of morning. Otherwise, if the value is greater than 12 but less than 16 then we increment the counter of afternoon. Otherwise if the value is grater than 16 then we increment the counter of evening. Next, we check which of morning,afternoon and evening is the highest and accordingly set the value to morning/afternoon/evening, indicating that that part of the day is the busiest. We do a write operation with key set to 'Semester_Year' and value set to morning/afternoon/evening depending on which has the highest count. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

In the output file we have the semester and the part of the day that is the busiest in that semester. From the results we can see that over the years exams take place more often in the afternoon then other parts of the days and morning are relatively free.

Question 16:
What is the most popular course in each semester?

```
Fall_1993    Intro to Pharmacology
Fall_1994    Intensive English Program
Fall_1995    Intensive English Program
Fall_1996    Intensive English Program
Fall_1997    Intensive English Program
Fall_1998    Intensive English Program
Fall_1999    Intensive English Program
Fall_2000    Intensive English Program
Fall_2001    Intensive English Program
Fall_2002    Intensive English Program
Fall_2003    Productn & Operatns Mgmt
Fall_2004    Intensive English Program
Fall_2005    Community Serv Internship
Fall_2006    Practicum
Fall_2007    Principles Public Hlth
Fall_2008    Independent Study
Fall_2009    Independent Study
Fall_2010    Dynamics of Leadership
Fall_2011    Eval Res Evid I
Fall_2012    Discovery Seminar Program
Fall_2013    Tx Plan & Cases 3
Fall_2014    Tx Plan & Cases 3
Fall_2015    Undergrad Superv Teach
Fall_2016
Spring_1994  Exercise Physiology
Spring_1995  Exercise Physiology
Spring_1996  Medical Biophysics
Spring_1997  Professional Problems
Spring_1998  Neuroscience II
Spring_1999  Neuroscience 2
Spring_2000  Community Serv Internship
Spring_2001  Community Serv Internship
Spring_2002  Community Serv Internship
Spring_2003  Community Serv Internship
Spring_2004  Community Serv Internship
Spring_2005  Community Serv Internship
Spring_2006  Human Anatomy
Spring_2007  Env Dsn Workshop 2
Spring_2008  School Media Ctr Prac
Spring_2009  Independent Study
Spring_2010  Independent Study
Spring_2011  Internship in Communication
Spring_2012  Internship in Communication
Spring_2013  Internship in Communication
Spring_2014  Tx Plan & Cases 2
Spring_2015  Tx Plan & Cases 2
Spring_2016  Internship
Summer_1994  Cad Applications
Summer_1995  Intro PT Evaluation Techs
Summer_1996  Psychological Statistics
Summer_1997  Social Problems
Summer_1998  Introduction to Sociology
```

Here, we set the <key,value> pair for each row by setting the key to the 'Semester_Year' and the value is a a concatenation of the course name and the percentage of enrollment. We calculate (enrollement/total capacity)*100 for each row where the total capacity is not equal to 0 and this is concatenated with the course id and set as the value. This is the output of the mapper. In the reducer we keep an integer variable max which is initially set to 0 and then for each <key,value> pair the value is split using the separator "-" and the second part of the string is converted to integer and compared to the max value and if it is greater than max is set to the greater value and a global variable 'sub' is set to the first part of the split string which the course name. When all the values of the same key are iterated the final values of max contains the maximum percentage of enrollment and 'sub' contains the subject associated with that value. We do a write operation using these <key,value> pairs. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

The output file contains the Semester and the course most popular in that semester. From the output we can infer that Intensive English Program was a highly preferred course in fall semester as it turned out to be the most popular for nine consecutive years and for spring and summer semester Community Serv Internship is the most popular.

Question 17:
What was the least busy part of the day during exams in each semester?

```
Fall_2011     Evening
Fall_2012     Evening
Fall_2013     Evening
Fall_2014     Evening
Fall_2015     Morning
Spring_2012   Morning
Spring_2013   Morning
Spring_2014   Morning
Spring_2015   Morning
Spring_2016   Morning
```

In this problem we create <key,value> pairs for each row where key is set to the 'Semester_Year' and value is set to the start time of the exam. What we do here is we split the start time using the separator ":" and the first part of the string is set as the value and sent to the reducer. This is the output of the mapper. In the reducer for each value we get the integer value using Integer.parseInt() and then check for three conditions. If the value is less than 12 then we increment the counter of morning. Otherwise, if the value is greater than 12 but less than 16 then we increment the counter of afternoon. Otherwise if the value is grater than 16 then we increment the counter of evening. Next, we check which of morning,afternoon and evening is the least and accordingly set the value to morning/afternoon/evening, indicating that that part of the day is the least busy. We do a write operation with key set to 'Semester_Year' and value set to morning/afternoon/evening depending on which has the least count. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

In the output file we have the semester and the part of the day that is the least busy in that semester. From the results we can see that in the past few years mornings are free as compared to afternoons and evenings so there can be more exams scheduled in the morning so there is even distribution and this might give more time gap between two exams.

Question 18:
Each year between fall and spring which semester has higher enrollment?

```
1914   Fall & Spring-0
1931   Fall-6
1932   Spring-6
1933   Fall & Spring-6
1934   Spring-6
1939   Fall-1
1941   Fall-1
1942   Spring-1
1945   Fall-15
1946   Fall-15
1947   Fall-27
1948   Fall-23
1949   Spring-22
1950   Fall-24
1951   Fall-48
1952   Spring-52
1953   Fall-58
1954   Spring-56
1955   Fall-79
1956   Spring-77
1957   Fall-97
1958   Spring-101
1959   Fall-165
1960   Fall-239
1961   Fall-303
1962   Fall-376
1963   Fall-649
1964   Fall-812
1965   Fall-1043
1966   Fall-1127
1967   Fall-1224
1968   Fall-1518
1969   Fall-1584
1970   Fall-1957
1971   Fall-2112
1972   Fall-2484
1973   Fall-2735
1974   Fall-3003
1975   Fall-3387
1976   Fall-3335
1977   Fall-3777
1978   Fall-4297
1979   Fall-5068
1980   Fall-5966
1981   Fall-6664
1982   Fall-7296
1983   Fall-8634
1984   Fall-11461
1985   Fall-13679
1986   Fall-17385
1987   Fall-21430
1988   Fall-28979
```

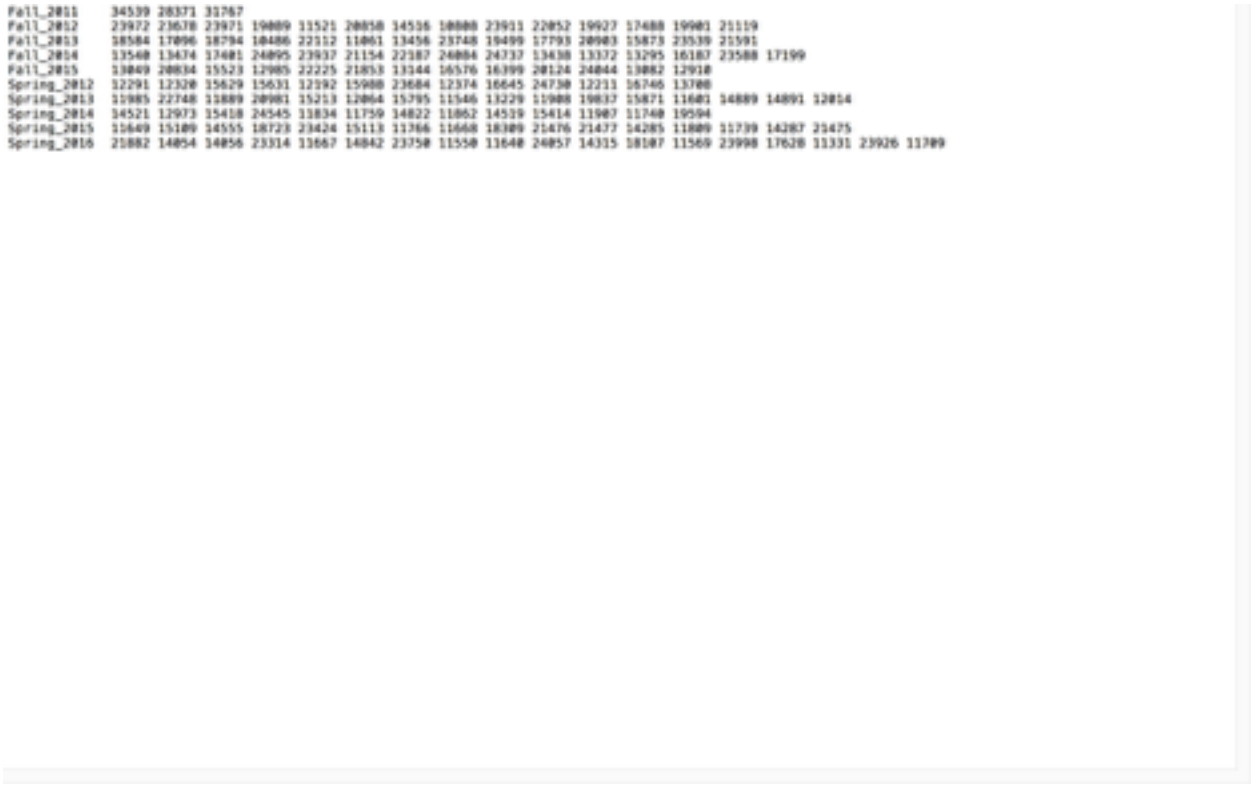
In this problem we set the <key,value> pairs in the mapper as <year, semester_enrollment> where semester is either fall or spring. This is the output of the mapper. In the reducer for each <key,value> pair we check if the value contains fall or spring. We split the value using the separator "-". If the first string contains fall then we convert the enrollment to integer and add its value to local variable fall and if the first string contains spring then we convert the enrollment to integer and add its value to local variable spring. Next we check for which value is greater between fall and spring and set the value to that semester and its total enrollment value and perform write operation for this <key,value> pair. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

In the output file we have year and the semester that has highest enrollment along with the value of enrollment. In some cases we see that the enrollment is 0 and in some cases we find that Fall and Spring have the same enrollment. It is also observed that enrollment increases over the years and is 0 for the year 2017 as we still haven't entered this year.

Question 19:
Which is the most used classroom in each semester during exams?

In this problem we set <key,value> pairs as <Semester_year, classnumber> and this is the output of the mapper. In the reducer for each reducer we keep a hash map which stores the classnumber and a counter for each classnumber. For each new <key,value> pairs we check if the key is already present in the hash map, if so we increment the value of the corresponding key in the hashmap by 1. If it is not present we add it to the hashmap and set its value to 1.
Next, we find the maximum value among all the values of the hashmap and then search for the keys that have the maximum value and add those keys to a string 's'. This string is set as the value and the semester and the key and write operation is done. We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

In the output file we have the semester and the classrooms used the most in that semester.



Question 20:
Which building uses the most labs each semester during exams?

```
Fall_2011    Bioed Dfn_A :2
Fall_2012    Cfa :16
Fall_2013    Park :4
Fall_2014    Cfa :46
Fall_2015    Cfa :33
Spring_2012  Cfa :5
Spring_2013  Bioed Cfa :4
Spring_2014  Cfa :46
Spring_2015  Cfa :41
Spring_2016  Cfa :36
```

For this problem we set the <key,value> pair. Key is the term where as for value we check if the facility type is 'LAB', if it is then we set the value to the building name. If not we ignore that case. This is the output of the mapper. In the reducer we keep a hashmap whose key is the building name and the value is a counter for each building. For each <key,value> pair we check if the value i.e. the building is present in the hashmap, if it is present then we increment the value of that key by 1 and if it is not present then we add that building to the hashmap. Next, we find the highest value in the hashmap and find the keys that have this value and append those keys to the string 's'. Finally we do a write operation on the initial key(Semester_year) and value('s').
We also append a check value to the value so that when it reached the second round of iteration it is written to the output file.

In the output file we have the semester and the names of the buildings that have the most labs during exams. If two or more buildings have the same number of labs, we mention all. From the output we observe that Cfa building has the most labs.

Conclusion:
Hence thesis questions give some useful insights about the dataset.