Project 1.1

Project 1.1 - USA Crime Analysis

Student Name: Abarajithan SA

Course: Big Data Hadoop & Spark Training

Contents

Note: - we are using PIG Commands to achieve the tasks requested in this Project

Introduction

This dataset contains attributes related to crimes taking place in various areas like type of crime, FBI code related to that criminal case, arrest frequency, location of crime etc.

Pre Requisite

You should have Hadoop cluster installed in your system and download the below dataset file and upload the file to your local file system or into HDFS, then start all Hadoop daemons.

Start hadoop daemons using command, ***Start-all.sh***

Associated Data Files

https://drive.google.com/file/d/0B1QaXx7tpw3SaUJHOHBZclBXWG8/view?usp=sharing

Dataset Description

ID, CaseNumber, Date,Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBICode, X Coordinate, Y Coordinate, Year, Updated On, Latitude, Longitude, Location.

Problem Statement

1. Write a MapReduce/Pig program to calculate the number of cases investigated under each FBI code

2. Write a MapReduce/Pig program to calculate the number of cases investigated under FBI code 32.

3. Write a MapReduce/Pig program to calculate the number of arrests in theft district wise.

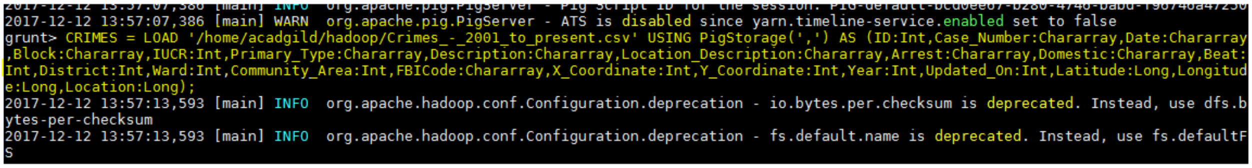
4. Write a MapReduce/Pig program to calculate the number of arrests done between October 2014 and October 2015.

Load the data into the PIG

Enter into PIG Grunt shell as local mode, ***grunt> Pig –x local***

Before proceeding with the tasks, we will Load the downloaded dataset into the PIG using the below PIG command,

CRIMES = LOAD '/home/acadgild/hadoop/Crimes_-_2001_to_present.csv' USING PigStorage(',') AS (ID:Int,Case_Number:Chararray,Date:Chararray,Block:Chararray,IUCR:Int,Primary_Type:Chararray,Description:Chararray,Location_Description:Chararray,Arrest:Chararray,Domestic:Chararray,Beat:Int,District:Int,Ward:Int,Community_Area:Int,FBICode:Chararray,X_Coordinate:Int,Y_Coordinate:Int,Year:Int,Updated_On:Int,Latitude:Long,Longitude:Long,Location:Long);



```
2017-12-12 13:57:07,386 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-[illegible]
2017-12-12 13:57:07,386 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt> CRIMES = LOAD '/home/acadgild/hadoop/Crimes_-_2001_to_present.csv' USING PigStorage(',') AS (ID:Int,Case_Number:Chararray,Date:Chararray
,Block:Chararray,IUCR:Int,Primary_Type:Chararray,Description:Chararray,Location_Description:Chararray,Arrest:Chararray,Domestic:Chararray,Beat:
Int,District:Int,Ward:Int,Community_Area:Int,FBICode:Chararray,X_Coordinate:Int,Y_Coordinate:Int,Year:Int,Updated_On:Int,Latitude:Long,Longitud
e:Long,Location:Long);
2017-12-12 13:57:13,593 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.b
ytes-per-checksum
2017-12-12 13:57:13,593 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultF
S
```

Task 1

1. Write a MapReduce/Pig program to calculate the number of cases investigated under each FBI code

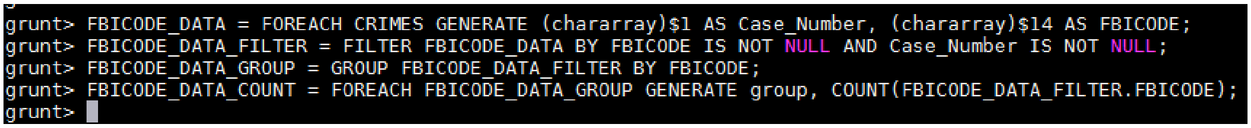
Steps and Explanation,

1. Generate the columns for the loaded dataset with column data types,
2. Filter the columns which you require to in the second step,
3. Group the filtered column in the second step with the key column value **FBICode**,
4. For the each group generated using the FBICode, count them for each FBICode.
5. DUMP the result,

PIG Commands,

1. ***FBICODE_DATA = FOREACH CRIMES GENERATE (chararray)$1 AS Case_Number, (chararray)$14 AS FBICODE;***
2. ***FBICODE_DATA_FILTER = FILTER FBICODE_DATA BY FBICODE IS NOT NULL AND Case_Number IS NOT NULL;***
3. ***FBICODE_DATA_GROUP = GROUP FBICODE_DATA_FILTER BY FBICODE;***
4. ***FBICODE_DATA_COUNT = FOREACH FBICODE_DATA_GROUP GENERATE group, COUNT(FBICODE_DATA_FILTER.FBICODE);***
5. ***DUMP FBICODE_DATA_COUNT;***

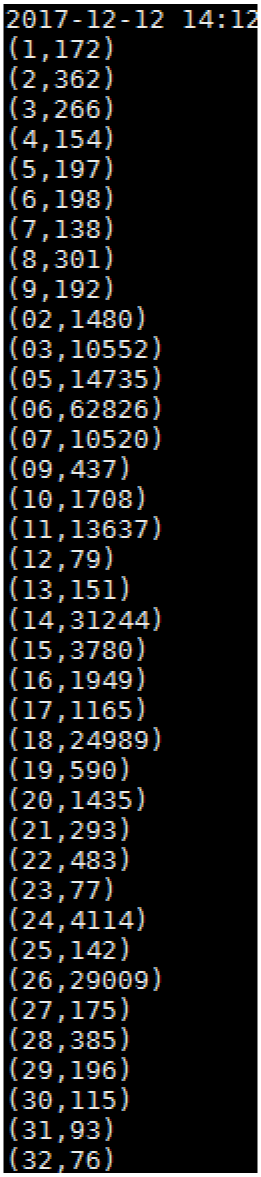
Screen Shot,



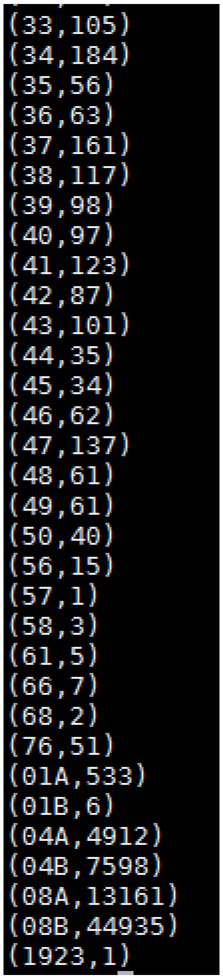
```
grunt> FBICODE_DATA = FOREACH CRIMES GENERATE (chararray)$1 AS Case_Number, (chararray)$14 AS FBICODE;
grunt> FBICODE_DATA_FILTER = FILTER FBICODE_DATA BY FBICODE IS NOT NULL AND Case_Number IS NOT NULL;
grunt> FBICODE_DATA_GROUP = GROUP FBICODE_DATA_FILTER BY FBICODE;
grunt> FBICODE_DATA_COUNT = FOREACH FBICODE_DATA_GROUP GENERATE group, COUNT(FBICODE_DATA_FILTER.FBICODE);
grunt>
```

1. As per the command 1, we are defining the Datatype for the required column loaded from the dataset
2. We are eliminating the rows where the value is NULL,
3. For each FBIcode we are creating a bag and set of tuples,
4. In the step number 4, the number of cases for each FBICode is counted using the COUNT() expression.

Required Output



```
2017-12-12 14:12
(1,172)
(2,362)
(3,266)
(4,154)
(5,197)
(6,198)
(7,138)
(8,301)
(9,192)
(02,1480)
(03,10552)
(05,14735)
(06,62826)
(07,10520)
(09,437)
(10,1708)
(11,13637)
(12,79)
(13,151)
(14,31244)
(15,3780)
(16,1949)
(17,1165)
(18,24989)
(19,590)
(20,1435)
(21,293)
(22,483)
(23,77)
(24,4114)
(25,142)
(26,29009)
(27,175)
(28,385)
(29,196)
(30,115)
(31,93)
(32,76)
```



```
(33,105)
(34,184)
(35,56)
(36,63)
(37,161)
(38,117)
(39,98)
(40,97)
(41,123)
(42,87)
(43,101)
(44,35)
(45,34)
(46,62)
(47,137)
(48,61)
(49,61)
(50,40)
(56,15)
(57,1)
(58,3)
(61,5)
(66,7)
(68,2)
(76,51)
(01A,533)
(01B,6)
(04A,4912)
(04B,7598)
(08A,13161)
(08B,44935)
(1923,1)
```

Task2

2. Write a MapReduce/Pig program to calculate the number of cases investigated under FBI code 32.

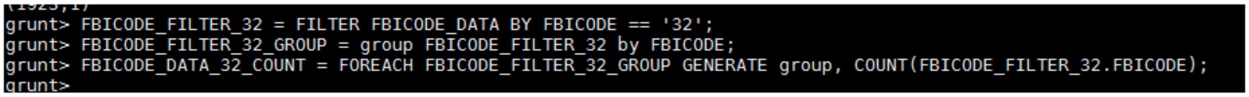
Steps and Explanation

1. Generate the columns for the loaded dataset with column data types,
2. Filter the columns by setting the FBIcode = 32 as it is requested,
3. Group the Filtered value, a
4. For each group, count the number of cases.

PIG Command

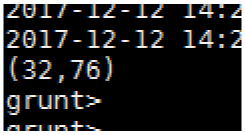
1. ***FBICODE_DATA = FOREACH CRIMES GENERATE (chararray)$1 AS Case_Number, (chararray)$14 AS FBICODE;***
2. ***FBICODE_FILTER_32 = FILTER FBICODE_DATA BY FBICODE == '32';***
3. ***FBICODE_FILTER_32_GROUP = group FBICODE_FILTER_32 by FBICODE;***
4. ***FBICODE_DATA_32_COUNT = FOREACH FBICODE_FILTER_32_GROUP GENERATE group, COUNT(FBICODE_FILTER_32.FBICODE);***
5. ***DUMP FBICODE_DATA_32_COUNT;***

Screen Shot



```
(1923,1)
grunt> FBICODE_FILTER_32 = FILTER FBICODE_DATA BY FBICODE == '32';
grunt> FBICODE_FILTER_32_GROUP = group FBICODE_FILTER_32 by FBICODE;
grunt> FBICODE_DATA_32_COUNT = FOREACH FBICODE_FILTER_32_GROUP GENERATE group, COUNT(FBICODE_FILTER_32.FBICODE);
grunt>
```

Required Output



```
2017-12-12 14:2
2017-12-12 14:2
(32,76)
grunt>
grunt>
```

Task3

3. Write a MapReduce/Pig program to calculate the number of arrests in theft district wise.

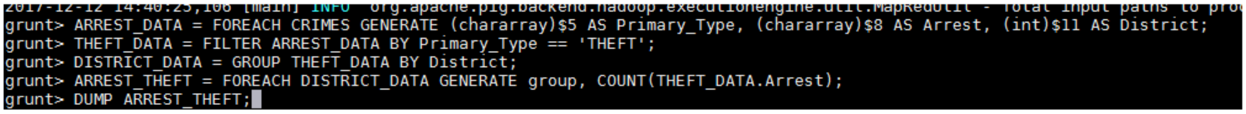
Steps and Explanation,

1. Generate the columns for the loaded dataset with column data types,
2. Filter the columns Primary_Type by THEFT, as we need the number of Arrests for the Primary type 'THEFT' district wise,
3. Group the filtered value using the column district,
4. Find the number of arrests using COUNT() expression

PIG Command

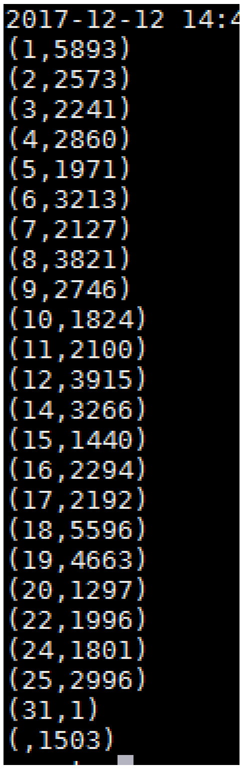
1. ***ARREST_DATA = FOREACH CRIMES GENERATE (chararray)$5 AS Primary_Type, (chararray)$8 AS Arrest, (int)$11 AS District;***
2. ***THEFT_DATA = FILTER ARREST_DATA BY Primary_Type == 'THEFT';***
3. ***DISTRICT_DATA = GROUP THEFT_DATA BY District;***
4. ***ARREST_THEFT = FOREACH DISTRICT_DATA GENERATE group, COUNT(THEFT_DATA.Arrest);***
5. ***DUMP ARREST_THEFT;***

Screen shot,



```
2017-12-12 14:40:25,106 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to proc
grunt> ARREST_DATA = FOREACH CRIMES GENERATE (chararray)$5 AS Primary_Type, (chararray)$8 AS Arrest, (int)$11 AS District;
grunt> THEFT_DATA = FILTER ARREST_DATA BY Primary_Type == 'THEFT';
grunt> DISTRICT_DATA = GROUP THEFT_DATA BY District;
grunt> ARREST_THEFT = FOREACH DISTRICT_DATA GENERATE group, COUNT(THEFT_DATA.Arrest);
grunt> DUMP ARREST_THEFT;
```

Required Output



```
2017-12-12 14:4
(1,5893)
(2,2573)
(3,2241)
(4,2860)
(5,1971)
(6,3213)
(7,2127)
(8,3821)
(9,2746)
(10,1824)
(11,2100)
(12,3915)
(14,3266)
(15,1440)
(16,2294)
(17,2192)
(18,5596)
(19,4663)
(20,1297)
(22,1996)
(24,1801)
(25,2996)
(31,1)
(,1503)
```

Task4

4. Write a MapReduce/Pig program to calculate the number of arrests done between October 2014 and October 2015.

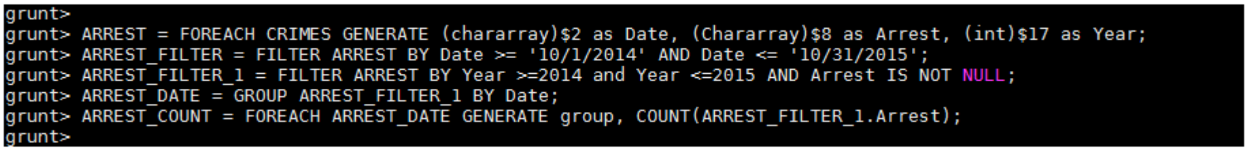
Steps and Explanation,

1. Generate the columns for the loaded dataset with column data types,
2. Filter the column **Dates** between October **2014** and October **2015** as per the request,
3. Filter the column **Year** between **2014** and **2015** where **Arrest** is not null,
4. Group the Year as per the Dates and
5. For each group in E, generate the count of number of arrest using COUNT() expression,

PIG Command

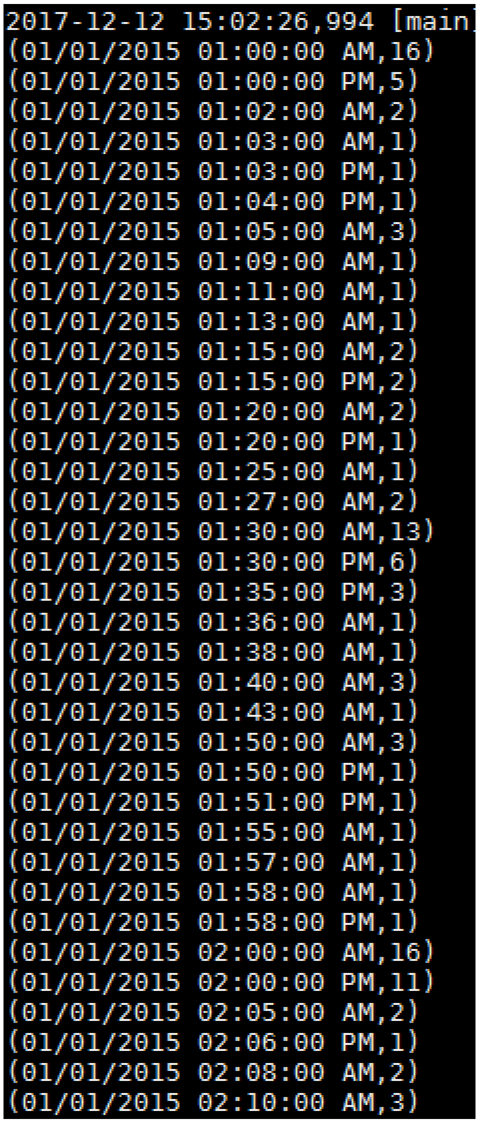
1. ***ARREST = FOREACH CRIMES GENERATE (chararray)$2 as Date, (Chararray)$8 as Arrest, (int)$17 as Year;***
2. ***ARREST_FILTER = FILTER ARREST BY Date >= '10/1/2014' AND Date <= '10/31/2015';***
3. ***ARREST_FILTER_1 = FILTER ARREST BY Year >=2014 and Year <=2015 AND Arrest IS NOT NULL;***
4. ***ARREST_DATE = GROUP ARREST_FILTER_1 BY Date;***
5. ***ARREST_COUNT = FOREACH ARREST_DATE GENERATE group, COUNT(ARREST_FILTER_1.Arrest);***
6. ***DUMP ARREST_COUNT;***

Screen Shot,

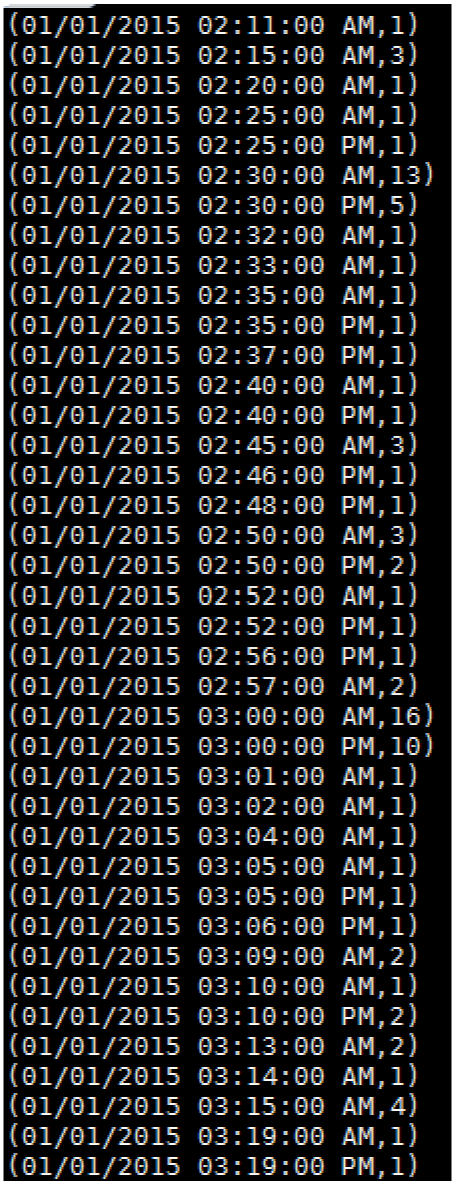


```
grunt>
grunt> ARREST = FOREACH CRIMES GENERATE (chararray)$2 as Date, (Chararray)$8 as Arrest, (int)$17 as Year;
grunt> ARREST_FILTER = FILTER ARREST BY Date >= '10/1/2014' AND Date <= '10/31/2015';
grunt> ARREST_FILTER_1 = FILTER ARREST BY Year >=2014 and Year <=2015 AND Arrest IS NOT NULL;
grunt> ARREST_DATE = GROUP ARREST_FILTER_1 BY Date;
grunt> ARREST_COUNT = FOREACH ARREST_DATE GENERATE group, COUNT(ARREST_FILTER_1.Arrest);
grunt>
```

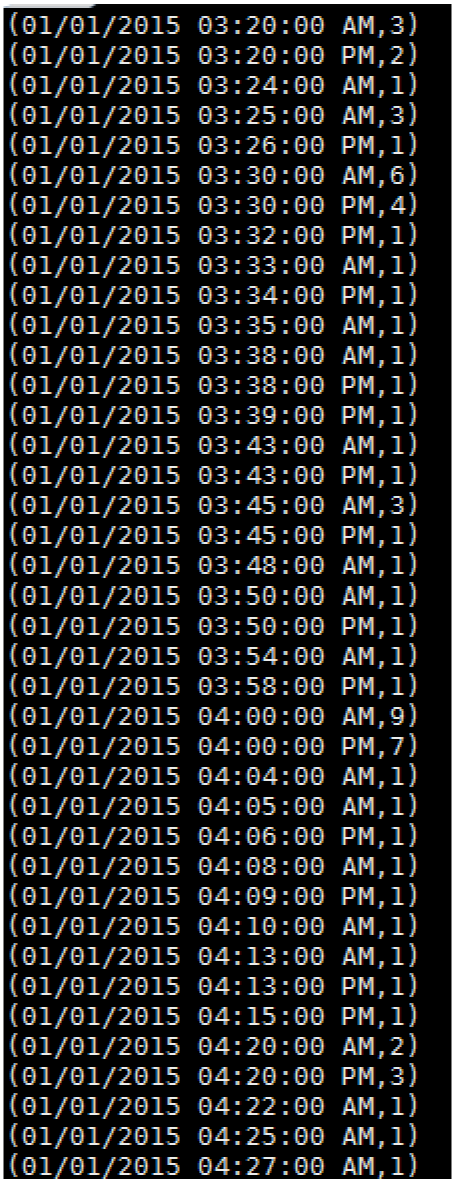
Required Output



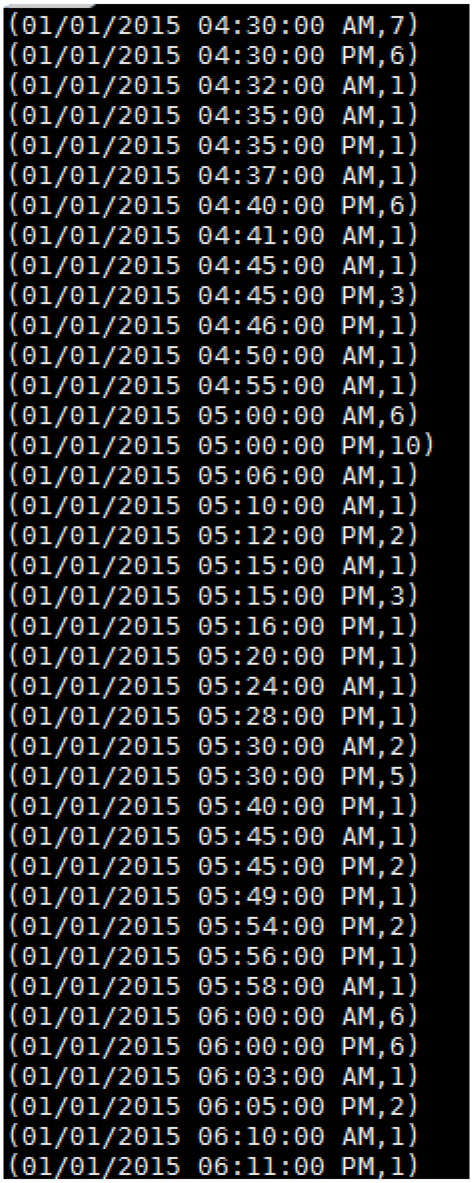
```
2017-12-12 15:02:26,994 [main
(01/01/2015 01:00:00 AM,16)
(01/01/2015 01:00:00 PM,5)
(01/01/2015 01:02:00 AM,2)
(01/01/2015 01:03:00 AM,1)
(01/01/2015 01:03:00 PM,1)
(01/01/2015 01:04:00 PM,1)
(01/01/2015 01:05:00 AM,3)
(01/01/2015 01:09:00 AM,1)
(01/01/2015 01:11:00 AM,1)
(01/01/2015 01:13:00 AM,1)
(01/01/2015 01:15:00 AM,2)
(01/01/2015 01:15:00 PM,2)
(01/01/2015 01:20:00 AM,2)
(01/01/2015 01:20:00 PM,1)
(01/01/2015 01:25:00 AM,1)
(01/01/2015 01:27:00 AM,2)
(01/01/2015 01:30:00 AM,13)
(01/01/2015 01:30:00 PM,6)
(01/01/2015 01:35:00 PM,3)
(01/01/2015 01:36:00 AM,1)
(01/01/2015 01:38:00 AM,1)
(01/01/2015 01:40:00 AM,3)
(01/01/2015 01:43:00 AM,1)
(01/01/2015 01:50:00 AM,3)
(01/01/2015 01:50:00 PM,1)
(01/01/2015 01:51:00 PM,1)
(01/01/2015 01:55:00 AM,1)
(01/01/2015 01:57:00 AM,1)
(01/01/2015 01:58:00 AM,1)
(01/01/2015 01:58:00 PM,1)
(01/01/2015 02:00:00 AM,16)
(01/01/2015 02:00:00 PM,11)
(01/01/2015 02:05:00 AM,2)
(01/01/2015 02:06:00 PM,1)
(01/01/2015 02:08:00 AM,2)
(01/01/2015 02:10:00 AM,3)
```



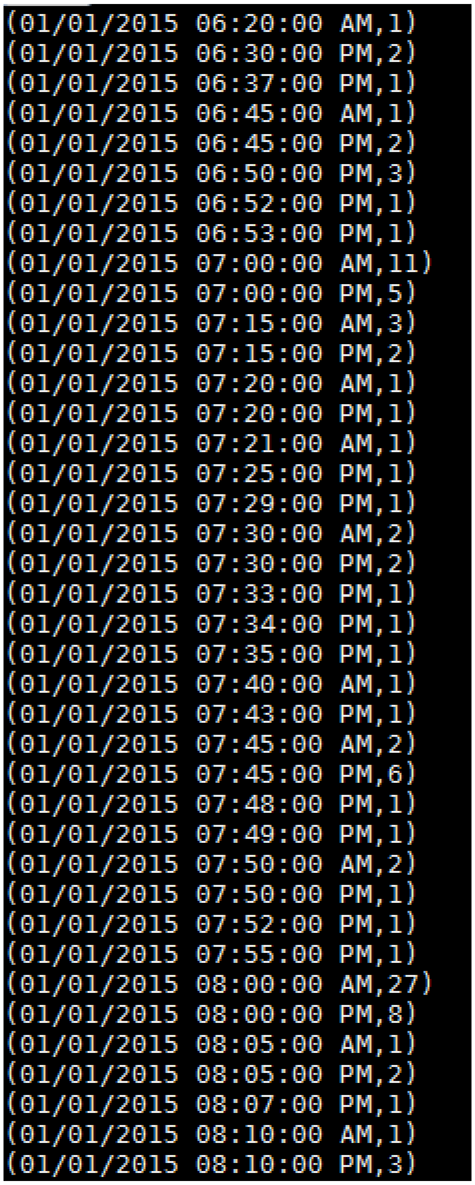
```
(01/01/2015 02:11:00 AM,1)
(01/01/2015 02:15:00 AM,3)
(01/01/2015 02:20:00 AM,1)
(01/01/2015 02:25:00 AM,1)
(01/01/2015 02:25:00 PM,1)
(01/01/2015 02:30:00 AM,13)
(01/01/2015 02:30:00 PM,5)
(01/01/2015 02:32:00 AM,1)
(01/01/2015 02:33:00 AM,1)
(01/01/2015 02:35:00 AM,1)
(01/01/2015 02:35:00 PM,1)
(01/01/2015 02:37:00 PM,1)
(01/01/2015 02:40:00 AM,1)
(01/01/2015 02:40:00 PM,1)
(01/01/2015 02:45:00 AM,3)
(01/01/2015 02:46:00 PM,1)
(01/01/2015 02:48:00 PM,1)
(01/01/2015 02:50:00 AM,3)
(01/01/2015 02:50:00 PM,2)
(01/01/2015 02:52:00 AM,1)
(01/01/2015 02:52:00 PM,1)
(01/01/2015 02:56:00 PM,1)
(01/01/2015 02:57:00 AM,2)
(01/01/2015 03:00:00 AM,16)
(01/01/2015 03:00:00 PM,10)
(01/01/2015 03:01:00 AM,1)
(01/01/2015 03:02:00 AM,1)
(01/01/2015 03:04:00 AM,1)
(01/01/2015 03:05:00 AM,1)
(01/01/2015 03:05:00 PM,1)
(01/01/2015 03:06:00 PM,1)
(01/01/2015 03:09:00 AM,2)
(01/01/2015 03:10:00 AM,1)
(01/01/2015 03:10:00 PM,2)
(01/01/2015 03:13:00 AM,2)
(01/01/2015 03:14:00 AM,1)
(01/01/2015 03:15:00 AM,4)
(01/01/2015 03:19:00 AM,1)
(01/01/2015 03:19:00 PM,1)
```



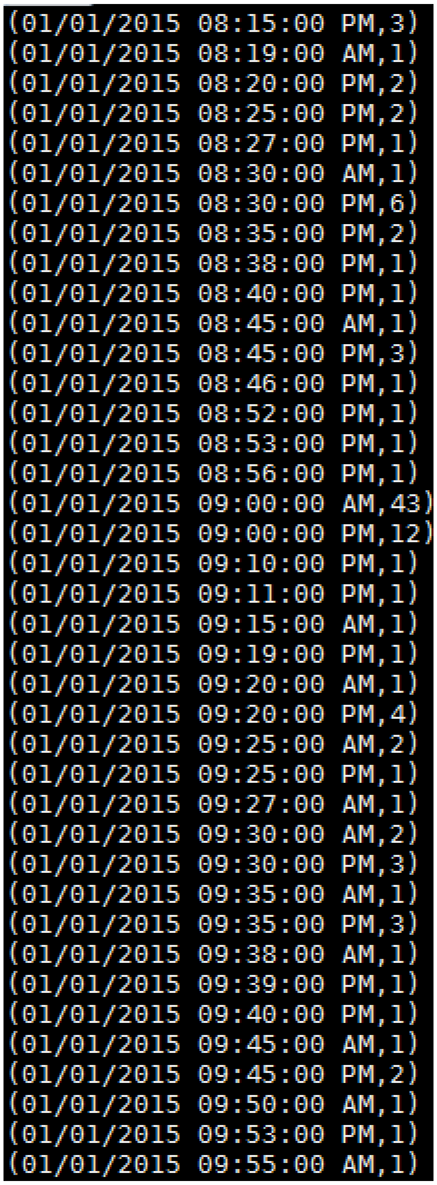
```
(01/01/2015 03:20:00 AM,3)
(01/01/2015 03:20:00 PM,2)
(01/01/2015 03:24:00 AM,1)
(01/01/2015 03:25:00 AM,3)
(01/01/2015 03:26:00 PM,1)
(01/01/2015 03:30:00 AM,6)
(01/01/2015 03:30:00 PM,4)
(01/01/2015 03:32:00 PM,1)
(01/01/2015 03:33:00 AM,1)
(01/01/2015 03:34:00 PM,1)
(01/01/2015 03:35:00 AM,1)
(01/01/2015 03:38:00 AM,1)
(01/01/2015 03:38:00 PM,1)
(01/01/2015 03:39:00 PM,1)
(01/01/2015 03:43:00 AM,1)
(01/01/2015 03:43:00 PM,1)
(01/01/2015 03:45:00 AM,3)
(01/01/2015 03:45:00 PM,1)
(01/01/2015 03:48:00 AM,1)
(01/01/2015 03:50:00 AM,1)
(01/01/2015 03:50:00 PM,1)
(01/01/2015 03:54:00 AM,1)
(01/01/2015 03:58:00 PM,1)
(01/01/2015 04:00:00 AM,9)
(01/01/2015 04:00:00 PM,7)
(01/01/2015 04:04:00 AM,1)
(01/01/2015 04:05:00 AM,1)
(01/01/2015 04:06:00 PM,1)
(01/01/2015 04:08:00 AM,1)
(01/01/2015 04:09:00 PM,1)
(01/01/2015 04:10:00 AM,1)
(01/01/2015 04:13:00 AM,1)
(01/01/2015 04:13:00 PM,1)
(01/01/2015 04:15:00 PM,1)
(01/01/2015 04:20:00 AM,2)
(01/01/2015 04:20:00 PM,3)
(01/01/2015 04:22:00 AM,1)
(01/01/2015 04:25:00 AM,1)
(01/01/2015 04:27:00 AM,1)
```



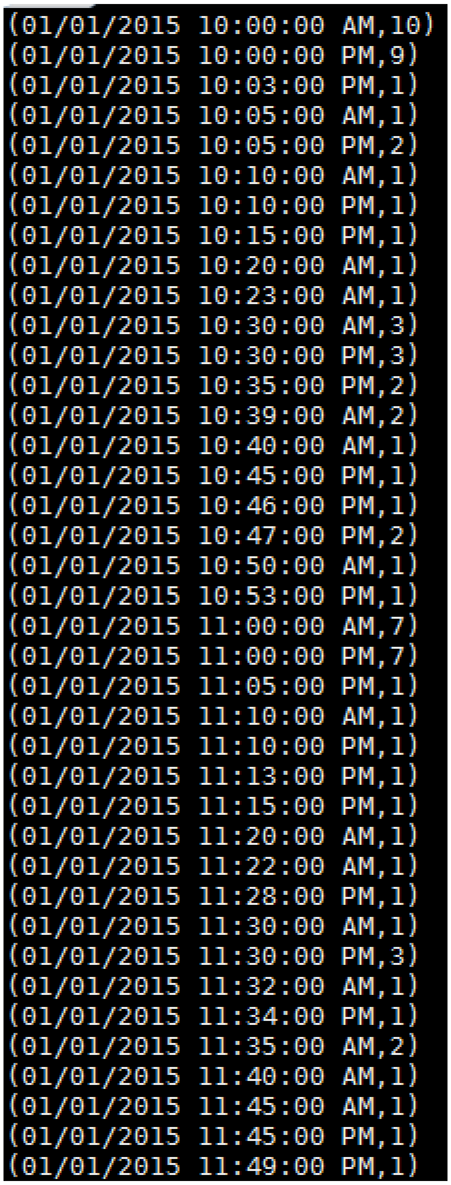
```
(01/01/2015 04:30:00 AM,7)
(01/01/2015 04:30:00 PM,6)
(01/01/2015 04:32:00 AM,1)
(01/01/2015 04:35:00 AM,1)
(01/01/2015 04:35:00 PM,1)
(01/01/2015 04:37:00 AM,1)
(01/01/2015 04:40:00 PM,6)
(01/01/2015 04:41:00 AM,1)
(01/01/2015 04:45:00 AM,1)
(01/01/2015 04:45:00 PM,3)
(01/01/2015 04:46:00 PM,1)
(01/01/2015 04:50:00 AM,1)
(01/01/2015 04:55:00 AM,1)
(01/01/2015 05:00:00 AM,6)
(01/01/2015 05:00:00 PM,10)
(01/01/2015 05:06:00 AM,1)
(01/01/2015 05:10:00 AM,1)
(01/01/2015 05:12:00 PM,2)
(01/01/2015 05:15:00 AM,1)
(01/01/2015 05:15:00 PM,3)
(01/01/2015 05:16:00 PM,1)
(01/01/2015 05:20:00 PM,1)
(01/01/2015 05:24:00 AM,1)
(01/01/2015 05:28:00 PM,1)
(01/01/2015 05:30:00 AM,2)
(01/01/2015 05:30:00 PM,5)
(01/01/2015 05:40:00 PM,1)
(01/01/2015 05:45:00 AM,1)
(01/01/2015 05:45:00 PM,2)
(01/01/2015 05:49:00 PM,1)
(01/01/2015 05:54:00 PM,2)
(01/01/2015 05:56:00 PM,1)
(01/01/2015 05:58:00 AM,1)
(01/01/2015 06:00:00 AM,6)
(01/01/2015 06:00:00 PM,6)
(01/01/2015 06:03:00 AM,1)
(01/01/2015 06:05:00 PM,2)
(01/01/2015 06:10:00 AM,1)
(01/01/2015 06:11:00 PM,1)
```



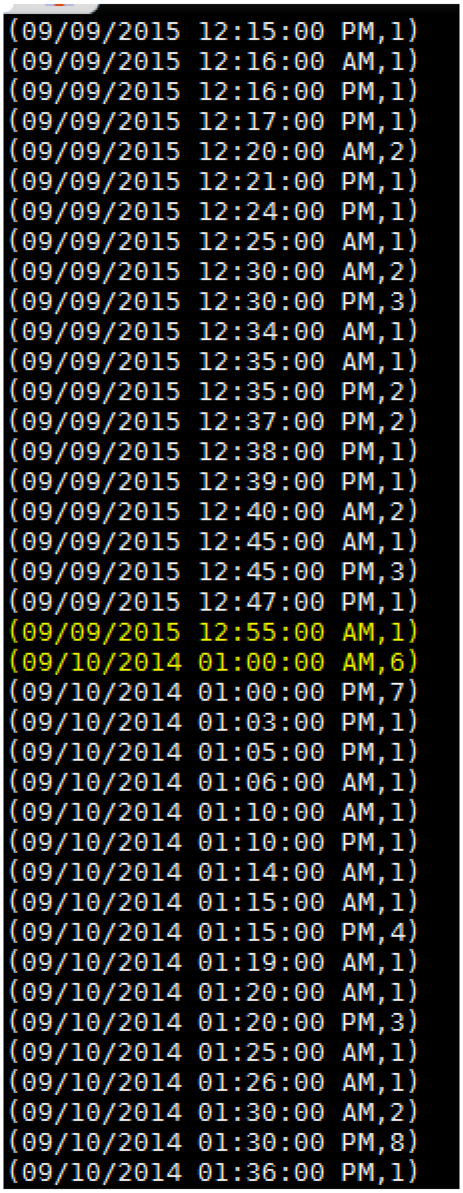
```
(01/01/2015 06:20:00 AM,1)
(01/01/2015 06:30:00 PM,2)
(01/01/2015 06:37:00 PM,1)
(01/01/2015 06:45:00 AM,1)
(01/01/2015 06:45:00 PM,2)
(01/01/2015 06:50:00 PM,3)
(01/01/2015 06:52:00 PM,1)
(01/01/2015 06:53:00 PM,1)
(01/01/2015 07:00:00 AM,11)
(01/01/2015 07:00:00 PM,5)
(01/01/2015 07:15:00 AM,3)
(01/01/2015 07:15:00 PM,2)
(01/01/2015 07:20:00 AM,1)
(01/01/2015 07:20:00 PM,1)
(01/01/2015 07:21:00 AM,1)
(01/01/2015 07:25:00 PM,1)
(01/01/2015 07:29:00 PM,1)
(01/01/2015 07:30:00 AM,2)
(01/01/2015 07:30:00 PM,2)
(01/01/2015 07:33:00 PM,1)
(01/01/2015 07:34:00 PM,1)
(01/01/2015 07:35:00 PM,1)
(01/01/2015 07:40:00 AM,1)
(01/01/2015 07:43:00 PM,1)
(01/01/2015 07:45:00 AM,2)
(01/01/2015 07:45:00 PM,6)
(01/01/2015 07:48:00 PM,1)
(01/01/2015 07:49:00 PM,1)
(01/01/2015 07:50:00 AM,2)
(01/01/2015 07:50:00 PM,1)
(01/01/2015 07:52:00 PM,1)
(01/01/2015 07:55:00 PM,1)
(01/01/2015 08:00:00 AM,27)
(01/01/2015 08:00:00 PM,8)
(01/01/2015 08:05:00 AM,1)
(01/01/2015 08:05:00 PM,2)
(01/01/2015 08:07:00 PM,1)
(01/01/2015 08:10:00 AM,1)
(01/01/2015 08:10:00 PM,3)
```



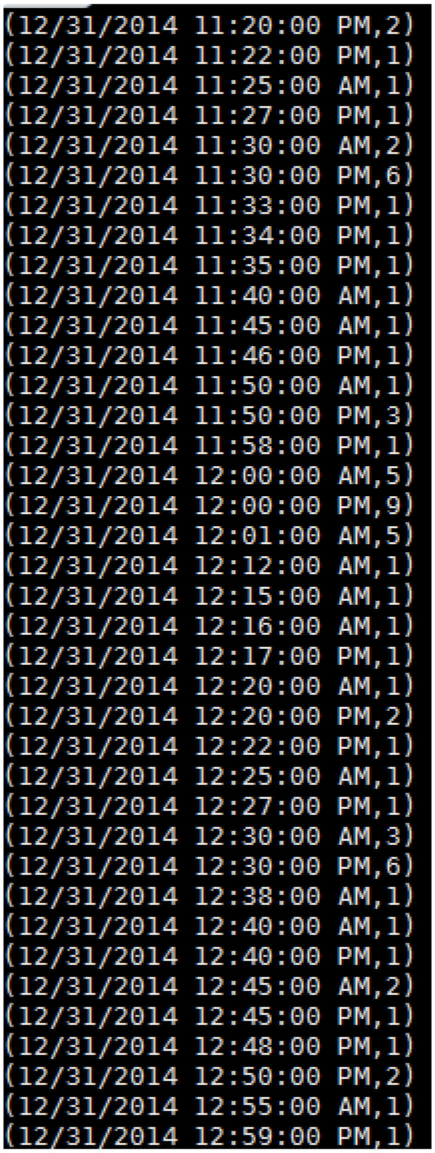
```
(01/01/2015 08:15:00 PM,3)
(01/01/2015 08:19:00 AM,1)
(01/01/2015 08:20:00 PM,2)
(01/01/2015 08:25:00 PM,2)
(01/01/2015 08:27:00 PM,1)
(01/01/2015 08:30:00 AM,1)
(01/01/2015 08:30:00 PM,6)
(01/01/2015 08:35:00 PM,2)
(01/01/2015 08:38:00 PM,1)
(01/01/2015 08:40:00 PM,1)
(01/01/2015 08:45:00 AM,1)
(01/01/2015 08:45:00 PM,3)
(01/01/2015 08:46:00 PM,1)
(01/01/2015 08:52:00 PM,1)
(01/01/2015 08:53:00 PM,1)
(01/01/2015 08:56:00 PM,1)
(01/01/2015 09:00:00 AM,43)
(01/01/2015 09:00:00 PM,12)
(01/01/2015 09:10:00 PM,1)
(01/01/2015 09:11:00 PM,1)
(01/01/2015 09:15:00 AM,1)
(01/01/2015 09:19:00 PM,1)
(01/01/2015 09:20:00 AM,1)
(01/01/2015 09:20:00 PM,4)
(01/01/2015 09:25:00 AM,2)
(01/01/2015 09:25:00 PM,1)
(01/01/2015 09:27:00 AM,1)
(01/01/2015 09:30:00 AM,2)
(01/01/2015 09:30:00 PM,3)
(01/01/2015 09:35:00 AM,1)
(01/01/2015 09:35:00 PM,3)
(01/01/2015 09:38:00 AM,1)
(01/01/2015 09:39:00 PM,1)
(01/01/2015 09:40:00 PM,1)
(01/01/2015 09:45:00 AM,1)
(01/01/2015 09:45:00 PM,2)
(01/01/2015 09:50:00 AM,1)
(01/01/2015 09:53:00 PM,1)
(01/01/2015 09:55:00 AM,1)
```



```
(01/01/2015 10:00:00 AM,10)
(01/01/2015 10:00:00 PM,9)
(01/01/2015 10:03:00 PM,1)
(01/01/2015 10:05:00 AM,1)
(01/01/2015 10:05:00 PM,2)
(01/01/2015 10:10:00 AM,1)
(01/01/2015 10:10:00 PM,1)
(01/01/2015 10:15:00 PM,1)
(01/01/2015 10:20:00 AM,1)
(01/01/2015 10:23:00 AM,1)
(01/01/2015 10:30:00 AM,3)
(01/01/2015 10:30:00 PM,3)
(01/01/2015 10:35:00 PM,2)
(01/01/2015 10:39:00 AM,2)
(01/01/2015 10:40:00 AM,1)
(01/01/2015 10:45:00 PM,1)
(01/01/2015 10:46:00 PM,1)
(01/01/2015 10:47:00 PM,2)
(01/01/2015 10:50:00 AM,1)
(01/01/2015 10:53:00 PM,1)
(01/01/2015 11:00:00 AM,7)
(01/01/2015 11:00:00 PM,7)
(01/01/2015 11:05:00 PM,1)
(01/01/2015 11:10:00 AM,1)
(01/01/2015 11:10:00 PM,1)
(01/01/2015 11:13:00 PM,1)
(01/01/2015 11:15:00 PM,1)
(01/01/2015 11:20:00 AM,1)
(01/01/2015 11:22:00 AM,1)
(01/01/2015 11:28:00 PM,1)
(01/01/2015 11:30:00 AM,1)
(01/01/2015 11:30:00 PM,3)
(01/01/2015 11:32:00 AM,1)
(01/01/2015 11:34:00 PM,1)
(01/01/2015 11:35:00 AM,2)
(01/01/2015 11:40:00 AM,1)
(01/01/2015 11:45:00 AM,1)
(01/01/2015 11:45:00 PM,1)
(01/01/2015 11:49:00 PM,1)
```



```
(09/09/2015 12:15:00 PM,1)
(09/09/2015 12:16:00 AM,1)
(09/09/2015 12:16:00 PM,1)
(09/09/2015 12:17:00 PM,1)
(09/09/2015 12:20:00 AM,2)
(09/09/2015 12:21:00 PM,1)
(09/09/2015 12:24:00 PM,1)
(09/09/2015 12:25:00 AM,1)
(09/09/2015 12:30:00 AM,2)
(09/09/2015 12:30:00 PM,3)
(09/09/2015 12:34:00 AM,1)
(09/09/2015 12:35:00 AM,1)
(09/09/2015 12:35:00 PM,2)
(09/09/2015 12:37:00 PM,2)
(09/09/2015 12:38:00 PM,1)
(09/09/2015 12:39:00 PM,1)
(09/09/2015 12:40:00 AM,2)
(09/09/2015 12:45:00 AM,1)
(09/09/2015 12:45:00 PM,3)
(09/09/2015 12:47:00 PM,1)
(09/09/2015 12:55:00 AM,1)
(09/10/2014 01:00:00 AM,6)
(09/10/2014 01:00:00 PM,7)
(09/10/2014 01:03:00 PM,1)
(09/10/2014 01:05:00 PM,1)
(09/10/2014 01:06:00 AM,1)
(09/10/2014 01:10:00 AM,1)
(09/10/2014 01:10:00 PM,1)
(09/10/2014 01:14:00 AM,1)
(09/10/2014 01:15:00 AM,1)
(09/10/2014 01:15:00 PM,4)
(09/10/2014 01:19:00 AM,1)
(09/10/2014 01:20:00 AM,1)
(09/10/2014 01:20:00 PM,3)
(09/10/2014 01:25:00 AM,1)
(09/10/2014 01:26:00 AM,1)
(09/10/2014 01:30:00 AM,2)
(09/10/2014 01:30:00 PM,8)
(09/10/2014 01:36:00 PM,1)
```



```
(12/31/2014 11:20:00 PM,2)
(12/31/2014 11:22:00 PM,1)
(12/31/2014 11:25:00 AM,1)
(12/31/2014 11:27:00 PM,1)
(12/31/2014 11:30:00 AM,2)
(12/31/2014 11:30:00 PM,6)
(12/31/2014 11:33:00 PM,1)
(12/31/2014 11:34:00 PM,1)
(12/31/2014 11:35:00 PM,1)
(12/31/2014 11:40:00 AM,1)
(12/31/2014 11:45:00 AM,1)
(12/31/2014 11:46:00 PM,1)
(12/31/2014 11:50:00 AM,1)
(12/31/2014 11:50:00 PM,3)
(12/31/2014 11:58:00 PM,1)
(12/31/2014 12:00:00 AM,5)
(12/31/2014 12:00:00 PM,9)
(12/31/2014 12:01:00 AM,5)
(12/31/2014 12:12:00 AM,1)
(12/31/2014 12:15:00 AM,1)
(12/31/2014 12:16:00 AM,1)
(12/31/2014 12:17:00 PM,1)
(12/31/2014 12:20:00 AM,1)
(12/31/2014 12:20:00 PM,2)
(12/31/2014 12:22:00 PM,1)
(12/31/2014 12:25:00 AM,1)
(12/31/2014 12:27:00 PM,1)
(12/31/2014 12:30:00 AM,3)
(12/31/2014 12:30:00 PM,6)
(12/31/2014 12:38:00 AM,1)
(12/31/2014 12:40:00 AM,1)
(12/31/2014 12:40:00 PM,1)
(12/31/2014 12:45:00 AM,2)
(12/31/2014 12:45:00 PM,1)
(12/31/2014 12:48:00 PM,1)
(12/31/2014 12:50:00 PM,2)
(12/31/2014 12:55:00 AM,1)
(12/31/2014 12:59:00 PM,1)
```

The output has many values and hence we clipped few values and pasted here.

*******************End of the document********************