Counting Unique Users

For this exercise, you may use the language of your choice. Also, feel free to use any of the language's standard features or libraries.

The provided CSV file documents user activity for an application. Each line in the file indicates the time of activity and the user associated with the activity. Determine how addictive the application is by computing how many **consecutive** days users that started playing the game on a given day ended up playing for. In this exercise we will consider the date range 7/1/2010 to 7/14/2010 and you can assume that nobody used the application on 6/30/2010 or 7/15/2010. Your output should consist of one line of output for each day in the date range, with each line having the format

```
DAY,USERS_PLAYING_1_DAY,USERS_PLAYING_2_DAYS,.....,USERS_PLAYING_14_DAYS
```

where DAY is an integer in the range from 1 to 14 and USERS_PLAYING_N_DAY(S) is a count of the people who **started** playing that day and then played for N consecutive days in a row. If they only played the day they started, and did not return the following day, this should be considered one day of consecutive play. If a users returns to the application after a gap in playing they should be counted the same way as a user who is seen for the first time.

The activity data provided covers the time range 7/1/2010 to 7/14/2010. The time in seconds from start of epoch to 7/1/2010 00:00:00 is 1277942400.

Input file format:

- CSV file
- each line ends with a '\n'
- the timestamp is field 1 and the UID is field 2
- timestamp is 32-bit int, indicating the number of seconds from start of epoch (1/1/1970 00:00) in UTC
- UID is 64-bit int
- the data in the file is sorted in ascending order of timestamps

Example input and output can be found in sample_input.csv and sample_output.csv, respectively.

Please provide your solution source code (including build/usage instructions) and the output your code produces from the test_input.csv input. Note that the test dataset is ~10x larger than the sample. The ideal solution should be able to handle much larger data sets.