Write a program in PHP to read lines from a file called `data.txt` with employee record data in the following format:

Example Input File:

```
999551241,baker,22-Feb-84,4,105000,good
999551252,chu,28-Mar-01,2,64000,good
999551255,anderson,15-Jan-05,1,42000,excellent
```

The fields are, in order: employee ID number, employee last name, start date, level, base pay, and evaluation rating. In the input, there will be one record per line, and the fields for each record will be separated by commas. You should hard-code your program to read data from a file called `data.txt` that will be in the same directory as your PHP program. A sample `data.txt` file will be posted on the course website for you to use in testing. However, I may use a file with the same format, but different data when grading your programs.

Your program should read in all the input records and then produce output as described below. Each line of the output should contain information from one record: the employee last name, a tab character, and then an expanded form of the start date. The records should be output in alphabetical order by employee last name. Only output records for employees whose evaluation rating is excellent or good AND whose salary is greater than 60000.

Example Output Format:

```
baker February 22, 1984
chu   March 28, 2001
```

Your program output should be displayed to the screen, not written to a file.

You should NOT use MySQL for this project – do all the processing within PHP.

Hint: Look at the PHP documentation on associative arrays and on the following functions: `ksort, strtotime, date`.

Your program should be contained in a single file and be entirely code that you write yourself. Name your file according to the following convention:

```
youronyen_p1.php
```

Replace *youronyen* with your actual Onyen (e.g. my assignment would be `rcapra_p1.php`). The character between *youronyen* and the "p1.php" part should be a single underscore. There should be no spaces or other characters in the filename. Files with names that do not follow this convention may receive a substantial grade deduction.

I strongly recommend using the PHP `file` command to read in the data file. We saw an example of this in class:

```
$lines = file("data.txt");
foreach ($lines as $line) {....}
```

The `files` command reads all the data from the file you specify into an array, with each array element getting one line from the file. In the example above, `$lines[0]` has the first line of `data.txt`, `$lines[1]` has the second line, etc.

I will test your program using the following command, with the file `data.txt` in the same directory:

```
php –qn youronyen_p1.php
```

IMPORTANT NOTE
While you are testing your program, you probably will want to run PHP WITHOUT the "-qn" arguments that I showed above. The "-qn" suppresses some output that will be useful to you in debugging.

Submit your file electronically through the Sakai by going to the Assignments area and finding the "P1" assignment. After you think you have submitted the assignment, I strongly recommend checking to be sure the file was uploaded correctly by clicking on it from within Sakai. Keep in mind that if I cannot access your file, I cannot grade it.

If for some reason you need to re-submit your homework file, you must add a version number to your filename. Sakai is configured so that it will accept up to 3 total submissions. Use the following file naming convention if you need to re-submit:

Your first submission:	`youronyen_p1.php`
Your second submission:	`youronyen_p1_v2.php`
Your third submission:	`youronyen_p1_v3.php`

Sakai is also configured with a due date and an "accept until" date. Submissions received after the due date (even just 1 minute!) will receive a 10% penalty per day. The "accept until" date is 5 days after the due date. Unless you have made arrangements with me in advance, submissions will not be accepted after the "accept until" date and will have a score of zero recorded.