CS417 Assignment 2: Frequently-asked questions

1. Q: How do I read a text file?

   A: Like this:

   ```
   in_handle = open(the_file_name, 'r')
      for line in in_handle.readlines():
          line = line.rstrip('\r\n')
          # do something with the line
   ```

2. Q: Given a line in a `.csv` file, how do I get the various fields?

   A: Use `date, precip, ... = line.split(",")`. The number of variables on the left of the `=` must match the number of fields in the data line.

3. Q: How do I know if the date is in ISO format or US format?

   A: Check if the date contains a `'-'` or a `'/'`. Here are examples:
   - `2020-01-29` : ISO format, contains a `'-'`
   - `1/29/20` : US format, contains a `'/'`
   - `23.0` : doesn't contain `'-'` or a `'/'`, you made a mistake somewhere.

4. Q: How do I get the parts of the date?

   A: It depends on the date format. If it's ISO, `date.split("-")`, and get the year, month, day. If it's US, `date.split("/")`, and get the month, day, year.

5. Q: How do I handle two-digit years?

   A: Ah, the Y2K bug. Since you know that the dates are increasing in time, you can keep track of the century. The century changes when:
   - the year has two digits (`year < 100`), and
   - the year is LESS than the previous line's year (you just went from `99` to `00`).

   So you need to know the previous year: at the bottom of the loop, save the year into `previous_year`.

   Anyway, if you detect a century change, you should increment the century, and use it to form a 4-digit year.

6. Q: How to I write to a text file?

A: Like this:

```
out_handle = open(the_file_name, 'w')
while ...:
    out_handle.write(some_string_goes_here + '\n')
```

7. Q: When I write a file, is there a convenient way to create a big string for the `.write()` call?

   A: Try `.format` Something like this:

```
  out_handle.write("{},{},{},...\n".format(date_in_year,
avg_precip,
                                          avg_low, ...))
```

8. Q: How do I convert `day` and `month` into the format `MM/DD`?

   A: Use `.format` Try this:

```
 date_in_year = "{:02d}/{:02d}".format(month, day)"
```

9. Q: How do I convert daily data into climate stats?

   A: Like this:

```
for line in in_handle.readlines():

     # get the year, month, day, precip, low, high

     # turn the month and day into a MM/DD date

     total_precip[date] += precip
     total_low[date] += low
     ...
     if low < record_low[date]:
         record_low[date] = low
     ...
```

10. Q: Does that mean that I need dictionaries?

    A: Yes. You need several dictionaries: for the sum of the precips, the sum of the lows, the sum of the hight, the record lows, the record highs, the record low year, and the record high year.

11. Q: Why am I getting a `KeyError`?

A: The *first* time that you do `total_precip[date] += precip`, you are adding to an entry that does not yet exist. Handle it separately:

```
if date in total_precip[date]:
    total_precip[date] += precip
else:
    # this is first time for this date
    total_precip[date] = precip
```

12. Q: How come the `plot.py` program doesn't recognize the data file? It thinks there is only one data line!

    A: It has to do with the newline chars at the end of the lines. If this happens to you, try opening your output file like as follows, to force it to use `\n` at the end of each line:

```
out_handle = open(output_filename, "w", newline="\n")
```