Write a program that allows a user to check-in: supply their current location as a string. Your program must compute the current date and store the check-in data in a dictionary and allow the user to query about their check-in data.

**Getting the current date and time**

```python
from datetime import datetime

# Prints the components of the current date
current_date = datetime.now()
print(current_date.month)
print(current_date.day)
print(current_date.year)

print(current_date.hour)
print(current_date.minute)
print(current_date.second)
```

You must use the `datetime.now()` function and the above attributes of the `datetime` object to associate the date and location data. You must use the date and time information (as a key) to index the location (string).

*Note:* if we only used the date as the key, then we couldn't have more than one check-in per day. Using only the date, the snippet:

```python
checkins["4/5/2012"] = "New York"
checkins["4/5/2012"] = "NYU"
print(checkins["4/5/2012"])  
```

would yield an output of “NYU”. The original value “New York” would be lost.

Hence, to ensure that each key is unique, we'll use the time information. You can store the date and time string as a key of the form: “4/5/2012 11:35:16” which implies that a check-in occurred on 4/5/2012 at 11:36:16 am. *Note: Python’s time information is based on a 24 hour time system, so 11pm is actually 23:00:00.*

**Menu System**

Welcome to Foursquare
What would you like to do?
  1. Check-in
  2. View Check-ins for a date
  3. View Check-ins for a location
  4. Exit
Your choice:
**Feature Details**

**View Check-ins for a date**: allow users to supply a forward-slash separated date (month/day/year) as the input. Your program must use this input to find and print all of the check-ins (locations) for that date.

**Hint**: since your dictionary keys include time information, you can check if the user’s date is a substring of the key. You can use the `count()` function to determine if “4/5/2012” is a substring of “4/5/2012 11:35:16”

Example:
```python
dt = "4/5/2012 11:35:16"
print(dt.count("4/5/2012")) # Output = 1 implying the substring was found
```

**View Check-ins for a location**: allow users to supply a string representing a location (e.g., “NYU”) as the input. Your program must use this input to find and print all of the dates (format: month/day/year hour:minute:second) where the user checked in to that location.

**Exit**: your program must terminate when the user chooses this option. You may print a goodbye message.

**Implementation Notes**

You must use a dictionary for this implementation. Multiple, associated arrays are not allowed!

**Extra Credit: (10 points)**

Incorporate file I/O! When your program starts up, load the existing check-ins (if any) from a file that you create, checkins.txt. The textfile should have two columns, date and location.

After you load all of the check-ins, when the user adds a new check-in, be sure to save it to the file. You have a few options of how to implement this: either save to the file as soon as you get the new check-in, or wait until the user chooses the Exit option to save the new data.

You’ll have to research how to write to a file in Python.

**CAUTION**

If, when reading the file, you end up getting extra newline characters at the end of each line, you can use the `strip()` function to remove leading and trailing whitespace and newlines.

Example:
```python
a = "4/15/2012\n"
a_stripped = a.strip()
print(a_stripped)  # Output is 4/15/2012
```

**Submission Details**

Please submit all of your source code, a screenshot of the working program, and your textfile if you choose to do the extra credit assignment.