CSE 101: Introduction to Computational and Algorithmic Thinking

Stony Brook University

Lab Assignment #9

Spring 2018

Assignment Due: April 6, 2018 by 11:59 pm

Assignment Objectives

This lab assignment will give you some practice with string processing.

Getting Started

Visit Piazza and download the "bare bones" file `lab9.py` onto your computer. Open `lab9.py` in PyCharm and fill in the following information at the top:

1. your first and last name as they appear in Blackboard
2. your Net ID (e.g., jsmith)
3. your Stony Brook ID # (e.g., 111999999)
4. the course number (CSE 101)
5. the assignment name and number (Lab #9)

Submit your final `lab9.py` file to Blackboard by the due date and time. Late work will not be graded. Code that crashes and cannot be graded will earn no credit.

Part I: Search the System Logs (20 points)

Computers generate many diagnostic, error and other informational messages as users login and logout, run programs, and perform other tasks on the computer. This collection of messages is known as a "system log" or simply, "log". For this problem you will write a function that takes a list of strings that represent log messages and extract information from them.

In our fictional computer system, log messages are formatted like this:

```
COMPUTER NAME;USER NAME;MESSAGE TYPE;MESSAGE
```

For example, consider the following log message:

```
SuperComp1;poohbear;Error;Failed to login successfully
```

We see that the log message consists of four parts, separated by semicolons:

- the name of the computer that generated the log messsage

- the user whose activity generated the log message
- the type of the log message
- the message itself

Complete the function `search_log`, which takes arguments in the following order:

1. `log_messages`: A list of messages from a fictional computer system, formatted as described above.
2. `user`: A user (string) whose name might appear in one or more log messages.
3. `message_type`: A type of message (string) that might appear in one or more log messages.

The function iterates over the list of log messages and extracts those messages generated by `user` and that are of type `message_type`. For example, if `user` were "poohbear" and `message_type` were "Warning", the function would search through the `log_messages` for strings wherein the user name is "poohbear" and the message type is "Warning". When the function finds such a log message, it extracts the user name and the actual message, inserting a > in between them, and appending the concatenated string to a result list. Continuing our example, suppose the function finds the following two *relevant* strings in the `log_messages` list:

```
'SuperComp1;poohbear;Warning;Failed to login'
'DingbatServer;poohbear;Warning;Maximum login attempts'
```

The returned list in this case would be:

```
['poohbear>Failed to login', 'poohbear>Maximum login attempts']
```

Example:

Due to the lengthy nature of the `log_messages` lists, only one example is provided here. The driver file will contain more test cases.

Arguments:

```
log_messages = [
   'Fiorelli;parentaudio;Warning;This domain controller cannot replicate the
        directory partition',
   'Adalbert;paddleballtony;Info;Btrfs loaded',
   'Holbein;parentaudio;Failure;Key type big_key registered',
   'Oroitz;paddleballtony;Warning;Windows is removing the remaining objects',
   'Oroitz;paddleballtony;Failure;Starting Flush Journal to Persistent Storage',
   'Lambin;parentaudio;Service;Windows is removing the remaining objects',
   'Fiorelli;ridingchamaeleon;Service;Low system resources',
   'Oroitz;parentaudio;Failure;Additional objects will be removed',
   'Oroitz;ridingchamaeleon;Info;This domain controller cannot replicate the
        directory partition',
   'Lambin;cabotagepride;Info;Low system resources',
   'Fiorelli;ridingchamaeleon;Info;Btrfs loaded',
   'Ferrari;cabotagepride;Warning;Starting Flush Journal to Persistent Storage',
   'Donus;paddleballtony;Service;Low system resources',
   'Ferrari;cabotagepride;Info;Low system resources',
   'Oroitz;parentaudio;Warning;Started LVM2 metadata daemon',
   'Holbein;ridingchamaeleon;Failure;Additional objects will be removed',
   'Holbein;cabotagepride;Failure;This domain controller cannot replicate the
        directory partition',
   'Lambin;parentaudio;Failure;Btrfs loaded',
   'Holbein;paddleballtony;Info;Starting Flush Journal to Persistent Storage',
   'Oroitz;ridingchamaeleon;Info;Reached target Local File Systems',
   'Donus;parentaudio;Info;Reached target Local File Systems',
   'Lambin;paddleballtony;Info;Additional objects will be removed',
   'Ferrari;cabotagepride;Warning;Started LVM2 metadata daemon',
   'Lambin;parentaudio;Failure;Windows is removing the remaining objects',
   'Morandi;ridingchamaeleon;Info;Low system resources',
   'Oroitz;parentaudio;Info;Btrfs loaded',
   'Adalbert;paddleballtony;Info;Additional objects will be removed',
   'Ferrari;parentaudio;Warning;Btrfs loaded',
   'Fiorelli;paddleballtony;Info;Btrfs loaded',
   'Donus;paddleballtony;Warning;This domain controller cannot replicate the
        directory partition',
   'Morandi;ridingchamaeleon;Warning;Low system resources',
   'Oroitz;cabotagepride;Failure;Reached target Local File Systems',
   'Ferrari;parentaudio;Failure;Reached target Local File Systems'
]

user = 'paddleballtony'

message_Type = 'Warning'
```

Return value:

```
[
  'paddleballtony>Windows is removing the remaining objects',
  'paddleballtony>This domain controller cannot replicate the directory partition'
]
```

Part II: Encode Letter Differences (20 points)

While solving this problem you will need to use the `ord()` function, which takes a character as its argument and returns an integer called its *Unicode value*. The Unicode value is the number used inside the computer's memory to store the character. For example, the uppercase letter A has the Unicode value 65, so `ord('A')` returns `65`. As another example, the lowercase letter z has the Unicode value 122, so `ord('z')` returns `122`.

Complete the function `encode_steps`, which takes one argument, `message`, a non-empty string of uppercase and lowercase letters. The function iterates over the `message` string one character at a time, computing the difference in Unicode code values between adjacent characters. More specifically, the code of the "earlier" character is subtracted from the code of the "later" character. For example, suppose we have the string "Worf". The respective Unicode values of these characters are 87, 111, 114 and 102. Therefore, the differences are 24 ($24 = 111 - 87$), 3 ($3 = 114 - 111$) and -12 ($-12 = 102 - 114$). The returned string is assembled by inserting the Unicode differences between adjacent characters. Continuing this example, for the argument string `'Worf'`, the returned value would be the string `'W24o3r-12f'`.

Note that the number of differences is one less than the number of characters. Therefore, if `message` contains only one character, the function should simply return `message`.

Examples:

Function Call	**Return Value**
`encode_steps('StonyBrook')`	`'S33t-5o-1n11y-55B48r-3o0o-4k'`
`encode_steps('US')`	`'U-2S'`
`encode_steps('Z')`	`'Z'`
`encode_steps('VudcLg')`	`'V31u-17d-1c-23L27g'`
`encode_steps('RqEHTLW')`	`'R31q-44E3H12T-8L11W'`
`encode_steps('gTlEbN')`	`'g-19T24l-39E29b-20N'`
`encode_steps('baTNdul')`	`'b-1a-13T-6N22d17u-9l'`
`encode_steps('OBcIVZub')`	`'O-13B33c-26I13V4Z27u-19b'`
`encode_steps('ismdbpkV')`	`'i10s-6m-9d-2b14p-5k-21V'`
`encode_steps('DXLey')`	`'D20X-12L25e20y'`

Note: The quotation marks displayed in the example return values are there to emphasize that the return values are strings. You should not add quotation marks to your return values.

How to Submit Your Work for Grading

To submit your `.py` file for grading:

1. Login to Blackboard and locate the course account for CSE 101.
2. Click on “Assignments” in the left-hand menu and find the link for this assignment.
3. Click on the link for this assignment.
4. Click the “Browse My Computer” button and locate the `.py` file you wish to submit. Submit only that one `.py` file.
5. Click the “Submit” button to submit your work for grading.

Oops, I messed up and I need to resubmit a file!

No worries! Just follow the above directions again. We will grade only your last submission.