

CSCI 1100 — Computer Science 1 Homework 6

Files and Sets

Homework Overview



Exterminate!

This homework is worth **90 points** toward your overall homework grade and is due **Friday April 8, 2016 at 4:59:59 pm**. It consists of one single program. Please review the rules about excess collaboration from HW 3 to make sure you are turning in a program that is your own.

We strongly suggest that you study the example in file `example_list_of_lists.py` before you get started. Put in print statements and step through it in the debugger to see how it works.

As always, you are expected to match your output to the one given on the submission server. While your grade is going to depend primarily on code correctness and structure, the more closely you match the submission server output, the easier it is for the **TAs**, the **instructors**, and **you** to determine that the code is working correctly. Be sure that you follow the rules we lay out for numbering, sorting, and organization and you will get maximum points on the homework!

You are not expected to use dictionaries to answer this homework, but you can, You must use **sets**. It will become quite obvious why as you start working on the program.

As mentioned in previous homeworks, your program must have the following basic structure, and adherence to program structure and organization will be graded:

```
'''
    A short comment detailing the purpose of the program and major
    assumptions that you are making. Your name and a date may help you later.
'''

## all imports
import math ## shouldn't be necessary, but here is where it would go

## all functions first
def function1(x):
    return x + 1

def function2(y):
    return y -1

## all your main code after this if statement
if __name__ == '__main__':
    z = 10
    print function1(z)
    print function2(z)
```

This homework brings together many little things you learned in different parts of this class. But, we will not tell you how to do the whole thing. We have a solution that is a total of 78 lines including blank lines for readability. It is not a long program if you plan it properly. We recommend constructing it in little pieces, similar to the way we've constructed examples in lecture.

A final word: The input file is from 2013, so it will not have the most recent information.

Problem: Villains and stories

You are given a file called `DrWhoVillains.tsv`. It is a TAB separated file containing villains in a science fiction show called Dr. Who. The show is actually known for its extremely low budget and silly creature effects. Although the modern version is slickly produced and has a higher budget, it still pays homage to the original look and feel of the old villains. Look at the picture of a Dalek (one of the main villains) above to see this for yourself. Each line of this file contains information on a single, unique villain. No villain is repeated on more than one line. Each line contains the following information (separated by TABs):

1. Villain: name of the villain (e.g. Dalek)
2. Year first: the first year this villain was introduced
3. Year last: the last year this villain was shown
4. Doc. no.: the list of doctor ids with this villain, separated by commas
5. Doctor actor: the list of actors for each doctor above
6. Episodes: number of episodes featuring this villain
7. Motivation (invasion earth, kill humans, etc): the description of the main motivation of this villain
8. Story titles: the titles of the stories that the villain was involved with, a list of strings separated by commas.

Note that some of the names of the villains are very long!

Your program must do the following:

- Read through the file to find the villain names the (set of) stories for each villain and the set of Doctors associated with this villain. Be careful, the file is manually created and may have repeated names of Doctors and Story titles, extra spaces before after names, etc. We will guarantee that each villain only appears on one line and that there are no misspellings, but you must use sets to find unique stories and Doctors.
- Find the top 10 most popular villains: those with top 10 highest number of stories listed in the file and display them to the user, numbered 1 to 10, in the decreasing order of popularity. Then, ask the user for a number.
- If the user enters -1, the program exits.
- If the user enters something other than a number between 1 to 10, or -1, the program displays the top 10 list again and asks for another input.
- If the user enters a valid option, numbered between 1 to 10, then you find the corresponding villain and display the following:
 - The number of stories with this villain
 - The names of all other villains that were in a story with this chosen villain, in alphabetical order.
 - The names of all story lines that featured only this villain, and no other villain. If none, you should display there is no such story line.
 - The number and names of all the Doctors (actors) who foiled this villain over the years.

Once you are done displaying this, print the top 10 again and ask for the next input. Stop when the user enters -1.

Display the names of matching villains and stories in a reasonable looking list. You should print each villain on a separate, numbered line. For very long villain names, you should truncate the string to 50 characters. We will not take points off for breaking lines mid-word.

Some Hints

We are not going to tell you exactly how to structure your code, but we are going to give you a few hints.

1. Parsing the villain file is an important of this HW. We suggest you create a function that takes a single string parameter corresponding to a single line of the input file, parses that line, and then returns a list containing:

```
[number of shows, villain name, set of show titles, set of doctors]
```

Think about this a little. It is not overly hard if you realize that we only need information from 3 of the 8 fields contained in the file. The remaining 5 fields – Year first, Year last, Doc. no., Episodes, and Motivation – are not used in this lab. Take the line, split it at the TABS. `Item[0]` is the villain name, `item[4]` is a comma separated string that you can split and turn into a set of Doctors, and `item[7]` is a comma separated string that you can split and turn into a set of story titles.

2. If you maintain a list of lists, for example if you make a list of the output from item 1 above for every line, it is simple to sort. Check out `example_list_of_lists.py` for code that will help you with this. Put in print statements and step through it in the debugger to see how it works.
3. Start small and build up. Make sure you can parse the lines correctly into a list first and test it. Then build up the list of lists. Then sort it. Then figure out how to generate the list of villains, etc. Once your program is working correctly, get the formatting right.
4. Lab 8 touches on some of the same topics and requires many of the same skills required here. If you are having problems, work through the material in Lab 8. It will help.

Deliverables – Final Check

Here is a summary of all the requirements for this homework. Double check this before you submit:

- Your program must use **sets**, this will also help you quite a lot.
- Submit a single file called `hw6.py` that assumes the existence of a text file called `DrWhoVillains.tsv`.
- The main loop of the program should be able to handle incorrect entry, repeated calls to display the top 10 villains, and exit when -1 is entered.
- Print all input that you read immediately.
- The outputs containing multiple items like villains, stories, or the list of Doctors should be a reasonable looking numbered list, wrapping long lines into multiple lines.
- Your program should print the villain names, story names, and Doctor's names in alphabetical order.
- Your program should have the required structure.

Expected Output

Below you can see the expected functioning of this program with the file we gave you (note: we might change the file in the homework submission server) (Note that the separators are strings of 50 '='s):

1. Daleks
2. Cybermen
3. Master (the)
4. Dalek Supreme (Supreme Dalek), inc Progenitor
5. Sontarans
6. Ice Warriors (inc benevolent appearances)
7. Davros
8. Silurians (exc Sea Devils)
9. Kovarian, Madame
10. Cyber-leader

```
Please enter a number between 1 and 10, or -1 to end
Enter a villain ==> Stop
Stop
```

1. Daleks
2. Cybermen
3. Master (the)
4. Dalek Supreme (Supreme Dalek), inc Progenitor
5. Sontarans
6. Ice Warriors (inc benevolent appearances)
7. Davros
8. Silurians (exc Sea Devils)
9. Kovarian, Madame
10. Cyber-leader

```
Please enter a number between 1 and 10, or -1 to end
Enter a villain ==> 5
5
```

Sontarans in 9 stories, with the following other villains:

- ```
=====
```
1. Alliance - onscreen - Daleks, Cybermen, Sontarans
  2. Atraxi
  3. Borusa
  4. Chessene of the Franzine Grig
  5. Cybermen
  6. Dalek Supreme (Supreme Dalek), inc Progenitor
  7. Daleks
  8. Dastari, Joinson
  9. Gray, Steve
  10. Harris, Private Carl
  11. Headless Monks
  12. Irongron
  13. Kovarian, Madame
  14. Manton, Colonel

15. Martha Jones- Clone
16. Master (the)
17. Naismith, Joshua
18. Rassilon
19. Rattigan, Luke
20. Red Carnivorous Maw
21. Shockeye of the Quawncing Grig
22. Silurians (exc Sea Devils)
23. Staal
24. Stike
25. Vardan
26. Varl
27. Varla

The stories that only features Sontarans are:

=====

1. The Sontaran Experiment

This villain was foiled by 6 doctor(s):

=====

1. Colin Baker
2. David Tennant
3. Jon Pertwee
4. Matt Smith
5. Patrick Troughton
6. Tom Baker

1. Daleks
2. Cybermen
3. Master (the)
4. Dalek Supreme (Supreme Dalek), inc Progenitor
5. Sontarans
6. Ice Warriors (inc benevolent appearances)
7. Davros
8. Silurians (exc Sea Devils)
9. Kovarian, Madame
10. Cyber-leader

Please enter a number between 1 and 10, or -1 to end

Enter a villain ==> 7

7

Davros in 6 stories, with the following other villains:

=====

1. Dalek Supreme (Supreme Dalek), inc Progenitor
2. Daleks
3. Lytton, Commander Gustav
4. Movellans
5. Nyder
6. Sharrel

There are no stories with only this villain

=====  
This villain was foiled by 5 doctor(s):  
=====

1. Colin Baker
2. David Tennant
3. Peter Davison
4. Sylvester McCoy
5. Tom Baker

1. Daleks
2. Cybermen
3. Master (the)
4. Dalek Supreme (Supreme Dalek), inc Progenitor
5. Sontarans
6. Ice Warriors (inc benevolent appearances)
7. Davros
8. Silurians (exc Sea Devils)
9. Kovarian, Madame
10. Cyber-leader

Please enter a number between 1 and 10, or -1 to end  
Enter a villain ==> 11  
11

1. Daleks
2. Cybermen
3. Master (the)
4. Dalek Supreme (Supreme Dalek), inc Progenitor
5. Sontarans
6. Ice Warriors (inc benevolent appearances)
7. Davros
8. Silurians (exc Sea Devils)
9. Kovarian, Madame
10. Cyber-leader

Please enter a number between 1 and 10, or -1 to end  
Enter a villain ==> -1  
-1  
Exiting