

CompSci 101

Introduction to Computer Science

0	'Susan'	0	['Smith', 'Brandt', 'Rodger', 'Crackers']
1	'Jackie'	1	['Long', 'Johnson']
2	'Mary'	2	['White', 'Rodger', 'Velios']
3	'Eric'	3	['Long', 'Lund']
4	'Jack'	4	['Frost']

October 23, 2014

Prof. Rodger

Events of Interest coming up

- ACM International Programming Contest
 - Sat. Nov. 1 – looking for volunteers to help
 - Top 3 teams – go to finals in Morocco
- Hacking and Hackathons Demystified – Ladies in Tech Unite
 - Thursday, Oct 23, Allen Bldg (TONIGHT) 6:30pm
- HackDuke.com - Hackathon Nov. 15-16

Announcements

- Reading for next time TBA
 - RQ 12 to be posted
- Hangman due next Thursday
- APT 6 is due on Tuesday
- Finish lecture notes from last time

Problem: Longest Name

www.bit.ly/101fall14-1023-01

Given a **list of names** (one word only) and a **letter** (assume names start with capital letter, and letter is capital)

names = ['Helen', 'Bob', 'Bart', 'Hugh']

- 1) Find the **longest name** that **starts with** that letter
- 2) Find the **position** of the longest name that starts with that letter

See longestName.py, DO NOT use enumerate

Enumerate

- An iterator, generates a sequence
- Generates **tuples** of (index, item)
- Used with **for** loop to get both **index** and **item**
- for (index,item) in somelist:
 - You get both at the same time!
- Redo find position of longest name with iterator

Problem: Popular Name

- Given a list of names, determine the **most popular first name** and print that name with all of its last names.
- Input: Names are always two words, names are in a file. If multiple names are on the same line they are separated by a “:”
- Output: Most popular first name, followed by a “:”, followed by corresponding last names separated by a blank

Example Input File with 5 lines

```
Susan Smith:Jackie Long:Mary White  
Susan Brandt  
Jackie Johnson:Susan Rodger:Mary Rodger  
Eric Long:Susan Crackers:Mary Velios  
Jack Frost:Eric Lund
```

Corresponding Output

```
Susan: Smith Brandt Rodger Crackers
```

One way to solve

- Create a list of unique first names
- Create a list of lists of last names that are associated with each first name

Example – two lists

Unique First names	Corresponding Last names
0 'Susan'	0 ['Smith', 'Brandt', 'Rodger', 'Crackers']
1 'Jackie'	1 ['Long', 'Johnson']
2 'Mary'	2 ['White', 'Rodger', 'Velios']
3 'Eric'	3 ['Long', 'Lund']
4 'Jack'	4 ['Frost']

Now can we solve the problem?

- Compute those two lists that are associated with each other
 - List of unique first names
 - List of corresponding last names
- Compute the max list of last names
- Now easy to print the answer.
- See popular.py

Expanding the Problem

- Suppose we want to read from multiple data files
names1.txt, names2.txt, names3.txt

See popular.py