

Election Results

Objectives

- Read in command-line arguments
- Read in data from files
- Store data into a dictionary
- Search the internet for how to format
- Format output

Election Results

The presidential election for the student council of your local university is about to be held. The chair of the election committee wants to computerize the voting and has asked you to write a program to analyze the data and report the winner. The university has four major divisions, and each division has several departments.

For the election, the four divisions are labeled as region 1, region 2, region 3, and region 4. Each department in each division handles its own voting and reports the votes received by each candidate to the election committee. The voting is reported in the following form:

```
firstName lastName regionNumber numberOfVotes
```

The election committee wants the output in the following tabular form: ***Please note: when you print a dictionary, it may print the entries in any order. So your table of Candidates may be in a different order each time you run it!***

Candidates	Votes	Percent
=====	=====	=====
Peter Pammer	387	17 %
Jamie Johnson	349	15 %
Greg Gohime	243	10 %
Lee Fisher	268	12 %
Tracey Boose	493	22 %
Mila Miller	476	21 %

The winner is Tracey Boose with 493 votes!

Total votes polled: 2216

You will read in the name of the data file from command-line arguments. An example data file (expected output shown above):

```
Tracey Boose 23 70 133 267
Jamie Johnson 25 71 156 97
Lee Fisher 110 158 0 0
Greg Gohime 75 34 134 0
Peter Pammer 285 56 0 46
Mila Miller 112 141 156 67
```



Another example data file:

```
John Black 93 20 39 167
Zoe Lester 75 44 57 87
Lee Alder 99 158 9 20
Oriah Zhang 75 34 104 0
```

And its expected output: *Remember, the order of the Candidates may appear in a different order!*

Candidates	Votes	Percent
=====	=====	=====
Oriah Zhang	213	19 %
John Black	319	29 %
Lee Alder	286	26 %
Zoe Lester	263	24 %

The winner is John Black with 319 votes!

Total votes polled: 1081

Requirements

- The name of the file must be called **ElectionResults.py**
- Comments at the top of your program
 - The 1st line of code must be: **#!/usr/bin/python3** (this is called the “shebang”) - it allows the auto-grader to know it should use python3
 - Your name
 - Date
 - Homework #4
 - Brief description of the assignment (one or two lines max)
- The program takes one command-line argument, the name of the file to use
- When reading in from the file, you will need to set up a dictionary data structure with the full name of the candidate as the key and the total number of votes the candidate got as the value.
- When you are done reading from the file, close the file and then loop through the dictionary to print out the table.
- The output must match exactly to the examples provided
 - The list of candidates should be centered across 15 spaces
 - The votes should be right-aligned across 8 spaces
 - The percent should be right-aligned across 8 spaces (*Hint: use 10 spaces for the headings, 8 spaces for the numbers*)
 - The empty lines also need to be there
- You should test with other data files as well (*note: when you use a different file, you do NOT change your code at all!*)
- Submit your assignment to [web-COG](#) (like you did with Lab 4) as a zip file named **Firstname_Lastname_HW4.zip**. Select the assignment “HW #4 Submit”. Also submit your .zip file to Moodle.

