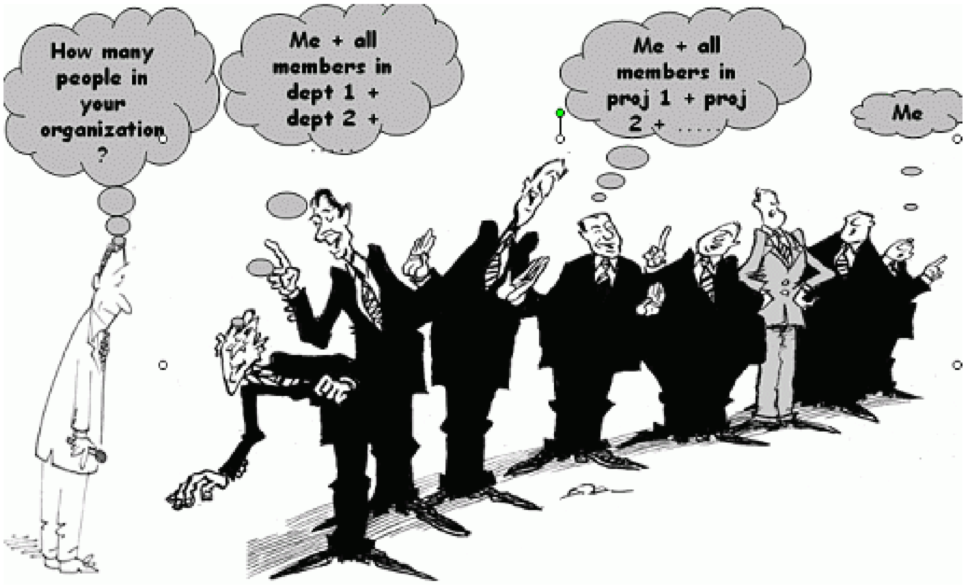
Recursive Lists


How many people in your organization ?
Me + all members in dept 1 + dept 2 + ...
Me + all members in proj 1 + proj 2 + ...
Me

Picture from: https://prezi.com/in5_yckckmjt/copy-of-recursion/

Objectives:

- Practice writing recursive functions

Lab Overview:

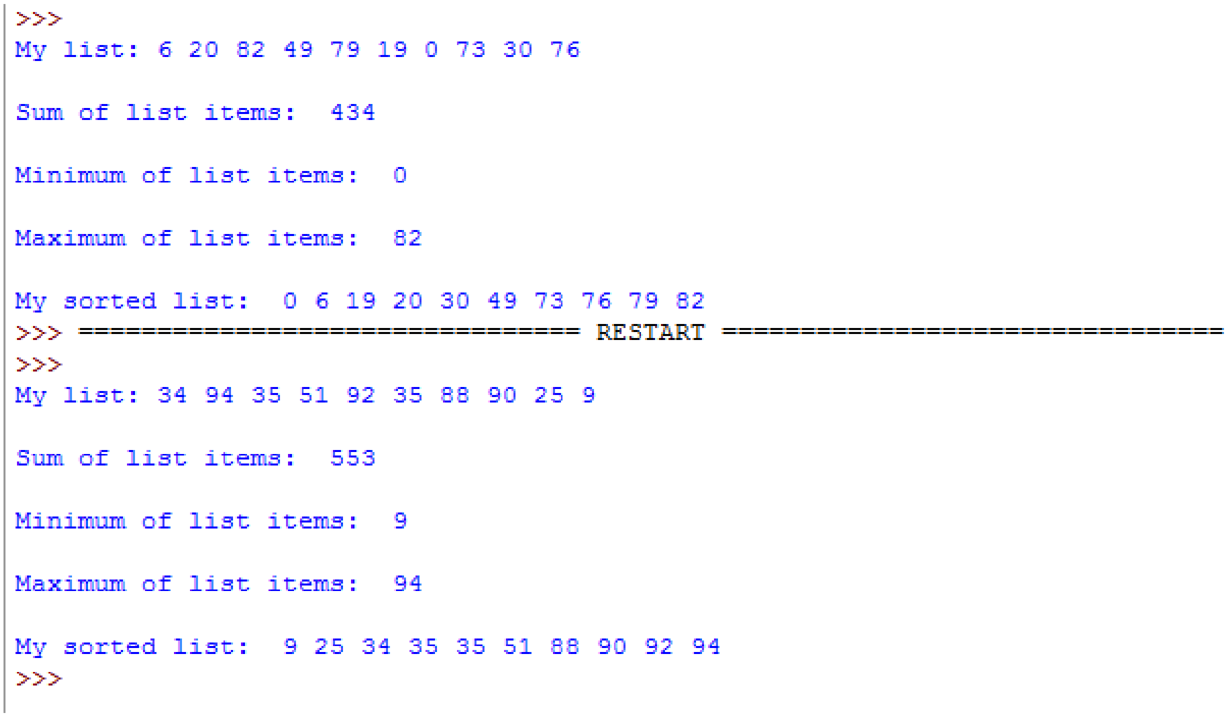
Quite a few functions that we’re used to using may be written so that they are performed recursively. We’ll be practicing recursion by making our own list functions for max, min, sum, sort, and print – and they will all be done recursively. You may not use the Python list functions for max, min, sum, or sort. You will probably want to use list slicing. For this practice, we’ll assume that the lists contain only integers.

Here’s the functions that you will need to write:

- `recursive_sum(a_list)`: This function should return the sum of the elements in the list.
- `recursive_max(a_list)`: This function should return the largest integer in the list. We discussed this one in class.
- `recursive_min(a_list)`: This function should return the smallest integer in the list.
- `recursive_print(a_list)`: This function should print the list on one line with a space between the numbers, and it should do it recursively. Please remember that you can override the default ending of a print statement with ‘\n’ character by including `end = ‘ ’` in your print statements.
- `quick_sort(a_list, first, last)`: This function sorts the list. You should find the code in the lecture slides.

You’ll need to place your functions in a file named `recursiveListFunctions.py` so that they will be imported properly into the driver file that is provided.

Sample Execution:



```
>>>
My list: 6 20 82 49 79 19 0 73 30 76

Sum of list items:  434

Minimum of list items:  0

Maximum of list items:  82

My sorted list:  0 6 19 20 30 49 73 76 79 82
>>> ================================ RESTART ================================
>>>
My list: 34 94 35 51 92 35 88 90 25 9

Sum of list items:  553

Minimum of list items:  9

Maximum of list items:  94

My sorted list:  9 25 34 35 35 51 88 90 92 94
>>>
```

Program Requirements:

For the code:

- Your program should have the correct comment block at the top (see last assignment)

- Use appropriate comments throughout the code. Be sure to include comments before each function definition explaining the interface (what must be sent, what is returned) and the purpose of the function. Also, don't forget to make appropriate comments within the method definition.

- Make good use of whitespace. Be sure to include a blank line of space before each comment. Two lines of space between functions is nice, but one will suffice – whichever you choose, be consistent. Follow the Python programming style guide: https://www.python.org/dev/peps/pep-0008/#introduction

For the lab report, follow the Lab Report Format Guide and complete the following sections:

- Title Page
- Analysis and Conclusions
- Appendix B - code (Be sure that you copy and paste the code into your word document, don't take a screen shot of it and paste that – it ends up being too hard to read.)

Deliverables:

Electronic submission in myCourses:

- Code (due by the end of the class period)
- Lab Report (due by the start of the next time that your lab meets) -- either a Microsoft Word document or a pdf

Paper submission in class:

- Lab Report (due at the start of lab the next time that it meets)

Grading:

Task	**Points**
Lab Report	30 points
Title Page	5 points
Analysis and Conclusions	10 points
Coding Style	15 points
Code	70 points
Recursive sum function	20 points
Recursive print function	20 points
Recursive max function	5 points
Recursive min function	15 points
Quicksort function	10 points