ITSE 1359 – Lab 3 Assignment (Refer to Ch. 3 as needed):

NOTE: Lab 3 requires you to work on and submit two programs. Read this document carefully and comply with submission and grading criteria. Your first problem (Southern States) is more prescriptive (use video provided), while the second program (Sales) is more open ended (solve as specified).

Problem #1 - Southern States (southern_states.py):

Problem Definition: Create a Python program that shows the southern states of the United States, first unsorted and then sorted. Save your program as southern_states.py. Do not forget general and specific comments. Keep your solution in a safe place because you will use it again in another lab assignment.

Use this list for your southern states: virginia, tennessee, arkansas, louisiana, north carolina, south carolina, mississippi, alabama, georgia, florida, texas. Make sure to store the states in the order specified above and in lower case. Name your list southern_states.

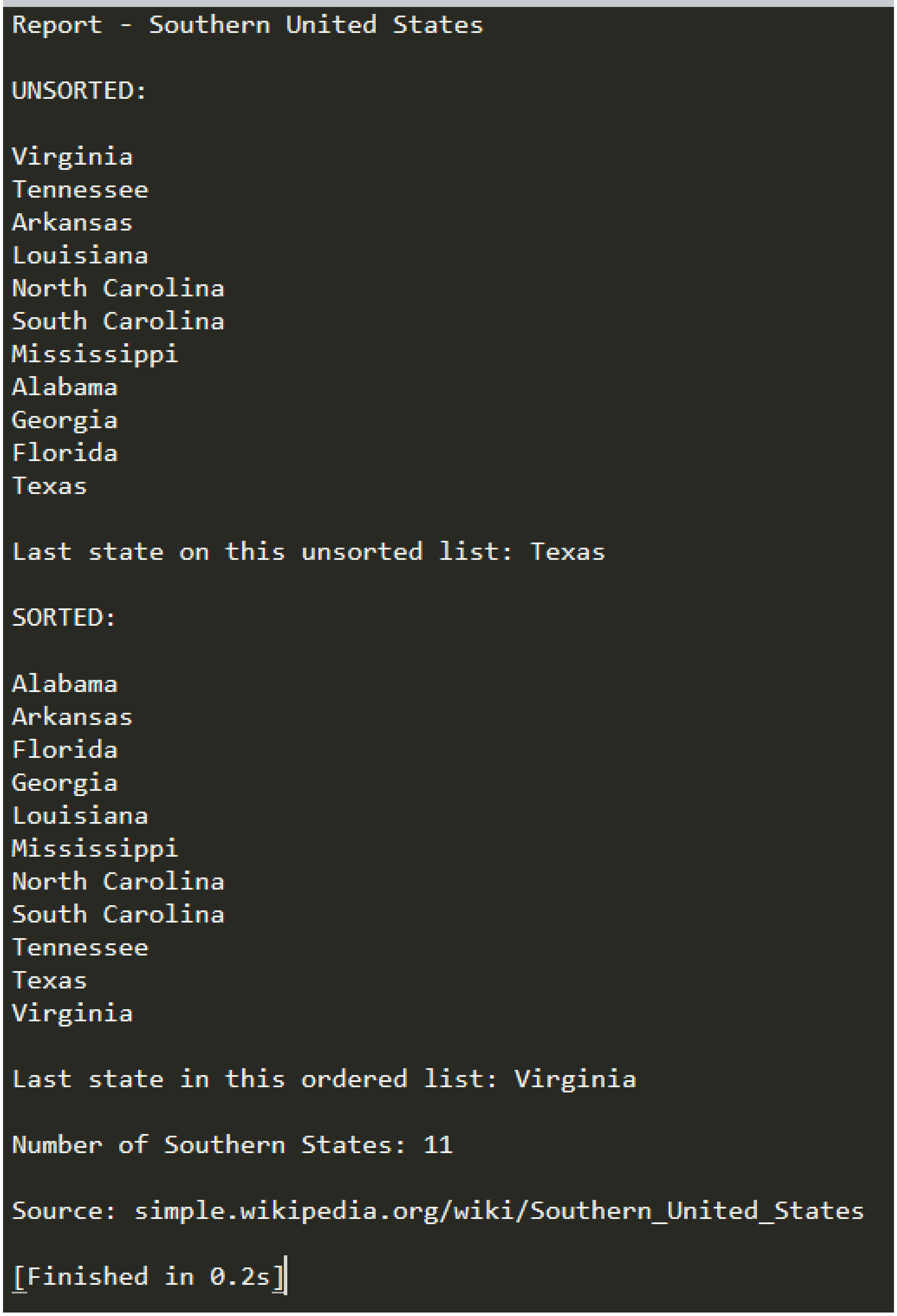
Coding: First show the list unsorted, followed by a printing of the last state on the list. Then sort the list and reshow the list. Finish by showing the last state on the list, the number of states in the list, and the source of the list.

Output: Use a formatted string to print out the states in title case. Use the index values of the list to access the elements in the list. Use a negative index to access the last element in the list. Use the len() function to return the number of states in the list. Sort your list with the sort() method. VIP: Your output should match screenshot provided.

Specific Comments / Pseudocode:

```
# initialize a list with names of the southern states all lowercase
# print name of report
# show unsorted list using index of each element in list - show title case
# use neg index to access last element in list and show it in title case
# sort the list using the sort() method and show list again in title case
# use neg index to access last element in list and show it in title case
# use len() function to show length of list (number of states)
# print credit line showing source of data
```

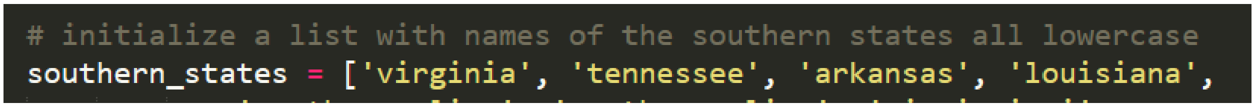
Screenshot of Output:



```
Report - Southern United States

UNSORTED:

Virginia
Tennessee
Arkansas
Louisiana
North Carolina
South Carolina
Mississippi
Alabama
Georgia
Florida
Texas

Last state on this unsorted list: Texas

SORTED:

Alabama
Arkansas
Florida
Georgia
Louisiana
Mississippi
North Carolina
South Carolina
Tennessee
Texas
Virginia

Last state in this ordered list: Virginia

Number of Southern States: 11

Source: simple.wikipedia.org/wiki/Southern_United_States

[Finished in 0.2s]
```

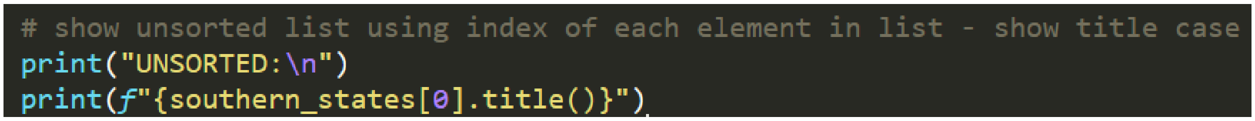
Code Examples (Southern States):

List of states (you complete):



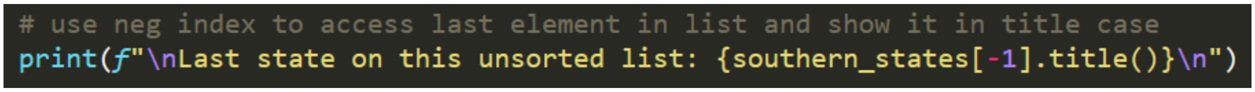
```
# initialize a list with names of the southern states all lowercase
southern_states = ['virginia', 'tennessee', 'arkansas', 'louisiana',
```

Printing the list (you complete):



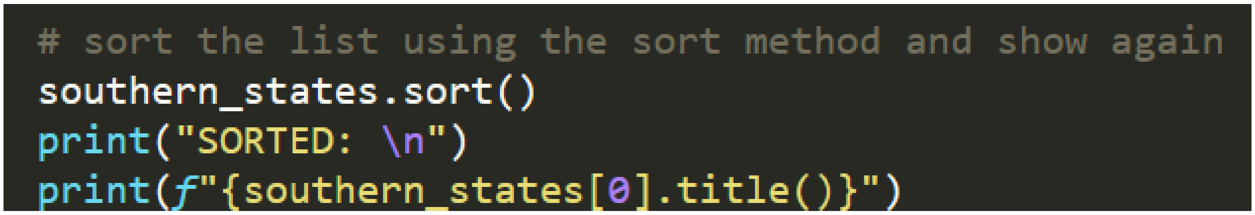
```
# show unsorted list using index of each element in list - show title case
print("UNSORTED:\n")
print(f"{southern_states[0].title()}")
```

Accessing the last item on the list using negative index value?



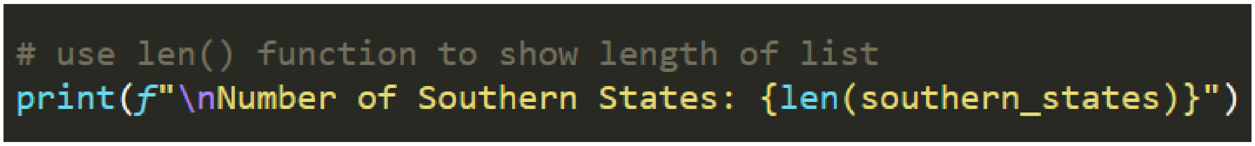
```
# use neg index to access last element in list and show it in title case
print(f"\nLast state on this unsorted list: {southern_states[-1].title()}\n")
```

How to sort the list?



```
# sort the list using the sort method and show again
southern_states.sort()
print("SORTED: \n")
print(f"{southern_states[0].title()}")
```

How many items are there in this list?



```
# use len() function to show length of list
print(f"\nNumber of Southern States: {len(southern_states)}")
```

Problem #2 - Sales (sales.py):

Problem Definition: Create a Python program that shows current sales for each day of the week as well as a projected 5% increase in sales. Name your program sales.py.

Coding: Create two parallel lists. The first list should hold the days of the week starting with Monday. This list should be in abbreviated format and in lowercase. The second list should hold the daily sales corresponding to the days of the week list. Initialize with this: 1000, 2000, 3000, 4000, 5000, 6000, 7000. Your fist list should contain string data and your second list should contain numeric data.

Output: First show the current sales data which should include the day of the week followed by its corresponding sales. Then, list the projected sales of 5% increase. Use an f string to print out the days of the week beginning with a capital letter and the sales should be formatted for currency. Output must match screenshot provided. Be sure to use tab escape codes to create the second column of output. In addition to general and specific comments, make sure you create the requested lists, access the items in the lists using index values, and show the output that formats the day of week and sales as specified.

Specific Comments (Sales):

# initialize list of days of week starting with Monday abbreviated lowercase

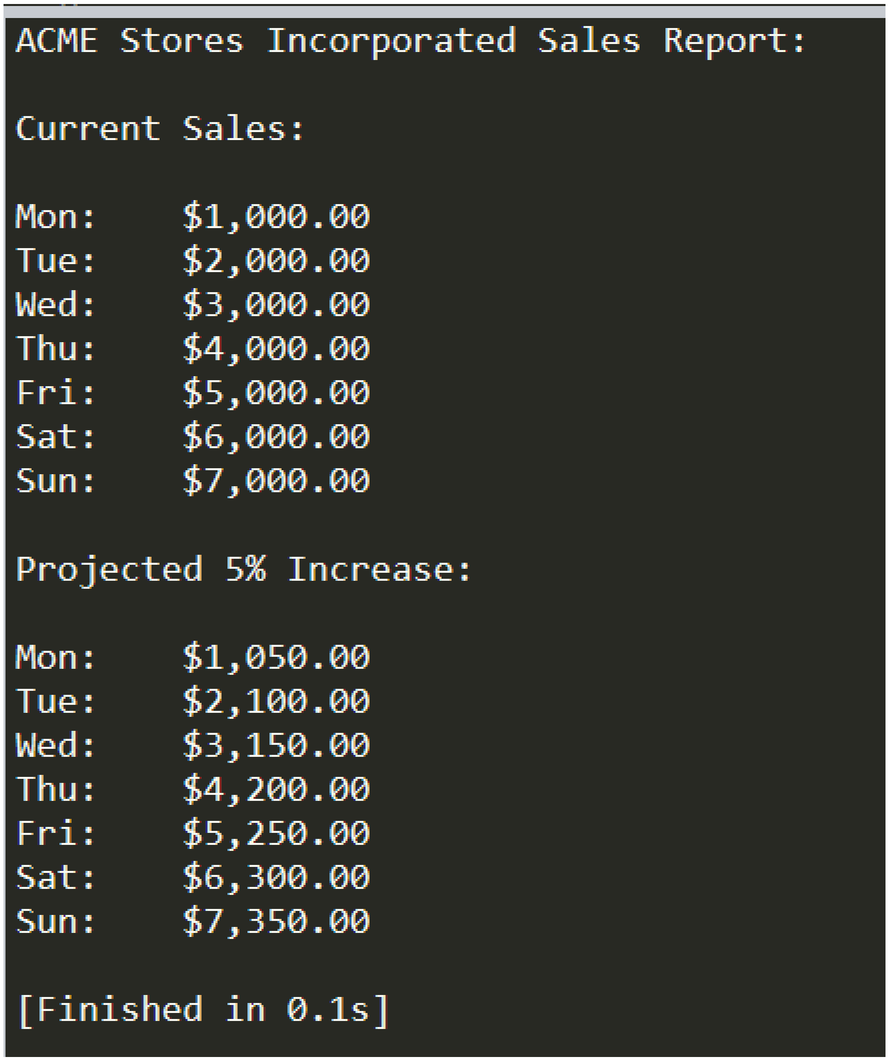
# initialize daily sales list corresponding to days of week

# print report header – Sales Report

# print current sales with formatted days and sales as currency

# print projected 5% increase in sales using same formatting as before

Screenshot of Output (Sales):



```
ACME Stores Incorporated Sales Report:

Current Sales:

Mon:    $1,000.00
Tue:    $2,000.00
Wed:    $3,000.00
Thu:    $4,000.00
Fri:    $5,000.00
Sat:    $6,000.00
Sun:    $7,000.00

Projected 5% Increase:

Mon:    $1,050.00
Tue:    $2,100.00
Wed:    $3,150.00
Thu:    $4,200.00
Fri:    $5,250.00
Sat:    $6,300.00
Sun:    $7,350.00

[Finished in 0.1s]
```

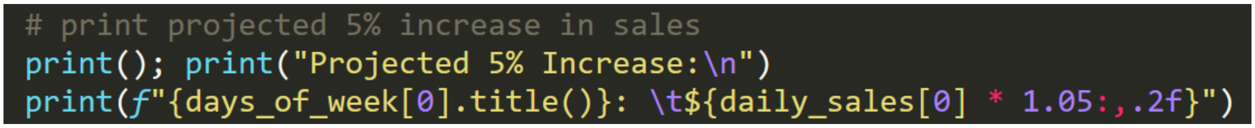
SPELLING: You should always use good grammar and spell correctly in the output of your programs. You are free to change the name of the company if you like or just say “Sales Report”.

FOCUS: For this program, I will zero in on creating lists and using the index value of a list to access the data. Do not use loops.

DATA TYPES: You are to create two list. One should contain the days of the week and the data should be string data type. The other list should contain the sales data for that day and it should be numeric. When you create your numeric list, do not put in anything other than numbers. You will format later.

Code Examples (Sales):

In sales, see how projected 5% increase in sales is done (note your list names may differ from mine):



```
# print projected 5% increase in sales
print(); print("Projected 5% Increase:\n")
print(f"{days_of_week[0].title()}: \t${daily_sales[0] * 1.05:,.2f}")
```

VIP: Ways to go wrong?

May sure your fist list is a string and your second list is numeric. So, if you make the daily_sales list a string, then you will not be able to do math on that list and you will get an error if you try.

From the code example above, you see my first list is days_of_week and my second list is daily_sales. So, if you name your lists something different, then use the names you created. Otherwise, you will get an error.

Food for Thought:

This lab asks you to show the items in the list by brute force – using the index values of the list. Don't you think a loop would be better? We will cover that soon – hang on! This lab is designed for you to master the basics of lists: how to create them, how to access them by means of their index values, and how to use some basic list methods and functions, like sort() and len(). Stay focused!

Submit your lab assignment:

Using the Canvas assignment tool, upload your completed work (2 files) to the lab 3 assignment. Attach the first file (southern_states.py) and then the second (sales.py) and then submit.

All lab assignments must be submitted using the CANVAS assignments tool. Lab Assignments will not be accepted any other way. Make sure you submit your work to the right lab assignment number otherwise you will not get credit.

VIP – Keep your Southern States Solution: Keep your solution to this program in a safe place because you will need it again for another lab assignment. `

Grading Criteria:

- ✓ You must use the provided video to help you do Southern States.
- ✓ Don't forget general comments.
- ✓ Specific comments are optional.
- ✓ Use white space to make your code easy to read.
- ✓ Comply with PEP-8 conventions for variable name, file names, etc.
- ✓ Satisfy the problem definition and other grading standards.
- ✓ Your work should not have syntax errors.
- ✓ Your work must be your own.
- ✓ Match your output screen to screenshot provided.
- ✓ Any deviations from lab specifications will result in points off.
- ✓ If you want to vary, do so on your own. Stay focused on specs!

VIP Videos:

How to do Southern States program (step-by-step): Lab 3 Southern States

I was a victim of copy and paste! In the above video, on time 19:20, I copy a line and forget to edit it. I copied the text that says "unsorted" when I need to say "ordered" or "sorted". Sorry! Prof. Benavides

Lectures and other videos: VIEW MY PYTHON PLAYLIST

Closing:

If you have questions about this lab send me a message using canvas inbox or attend zoom office hours. See Unit 0 for link for my office hours. Programming tutors are also available and listed in your Canvas class as announcements.

By-the-way, a great way to get ready for your lab assignments (and exams) is do the Try It Yourself problems in your book. Most of the solutions are on the authors website and I also discuss them in my YouTube video lectures.

Another way to get ready for your lab assignments is to review my lecture notes for each chapter. In Unit 0, find the link for my chapter notes.

Warning: Labs will increase in complexity with each lab assignment. A beginner may need 5 to 15 hours to complete each lab. Please start early because there will be no extensions.

Consider doing this lab over a three-day period:

- ✓ Day 1 – Get confused and run out of time.
- ✓ Day 2 – Research and debug errors.
- ✓ Day 3 – Polish, double-check everything, and submit.

Have an exception free day!

Prof. Benavides