*Turn on the Navigation Pane in Word when viewing this document:* ***View tab > enable Navigation pane***

# Program Specification

* *Program title:* *List Processing*
* *File name:* **MIS3301-Ch7-HW-Travel.py**
* *Due Date:* see Canvas for due date.
* *Main objective:* Victorino Travel Tours is asking that you create an application for their customers to lookup information about their travel destinations.
* Refer to the sample *Execution Screenshot* below for additional details.
  + Red, bold text is a calculated or determined value.

# Read before starting

* The expectation is that students have read the *entire* chapter and attended *all* lectures.
* Reminder: This is an *individual* assignment. Students are expected to complete this assignment independently and without the assistance of others. If you need help, contact your instructor via email or visit during office hours.
  + When emailing your instructor, please provide as much information as possible:
    1. State what the issue is and provide line numbers.
    2. Attach your \*.py file *zipped* as outlook does not permit attaching these files.
    3. Also, copy & paste the snippet of code where the issue is occurring within the email. If I am not at a computer when I see your email, I may be able to respond by seeing the snippet of code.
* I recommend that you read the entire assignment before beginning.
* Periodically review the ***Assignment Tips & Updates*** page in our module looking for tips as well as any ***required*** updates to this assignment.
* **This code must be free of syntax errors, or it will not be graded. Thus, comment out any code that crashes, or for a better grade, improve the code and submit it one day late with a 10% deduction.**

# Execution Screenshot

|  |
| --- |
| 1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. Victorino Travel Tours 3. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. List of Available Destinations 5. ---------------------------------------- 6. 1: Empire State Building - New York, NY 7. 2: Everglades National Park - Florida, FL 8. 3: Fort Worth Stockyards - Fort Worth, TX 9. 4: Rocky Mountain National Park - Estes Park, CO 10. 5: Grand Canyon National Park - Arizona, AZ 11. 6: Pikes Peak - Colorado, CO 12. 7: River Walk - San Antonio, TX 13. X: Press to exit 14. Enter a destination #: **7** 15. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 16. Destination: River Walk - San Antonio, TX 17. The San Antonio River Walk (also known as Paseo del Río or simply as 18. The River Walk) is a city park and network of walkways along the banks 19. of the San Antonio River, one story beneath the streets of San 20. Antonio, Texas, United States. Lined by bars, shops, restaurants, 21. nature, public artwork, and the five historic missions, the River Walk 22. is an important part of the city's urban fabric and a tourist 23. attraction in its own right. [source:wikipedia.org] 24. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 25. Would you like to know more about the state (y/n)? **y** 26. ----------------------------------- 27. State: TX - Texas 28. Capital: Austin 29. Bird: Mockingbird 30. Flower: Bluebonnet 31. Join Order: 28 32. Join Date: 12/29/1845 33. Views: 0 34. ----------------------------------- 35. Press enter to continue... |

Please note that the Canvas File Preview does not render the alignment in this execution screenshot well. Open this document in MS Word.

# Lists

## Destinations

* You will create the ***DESTINATIONS*** list which is a basic (1-dimensional) list with the following elements.

|  |  |
| --- | --- |
| *0* | Empire State Building - New York, NY |
| *1* | Everglades National Park - Florida, FL |
| *2* | Fort Worth Stockyards - Fort Worth, TX |
| *3* | Rocky Mountain National Park - Estes Park, CO |
| *4* | Grand Canyon National Park - Arizona, AZ |
| *5* | Pikes Peak - Colorado, CO |
| *6* | River Walk - San Antonio, TX |

## Destination Details

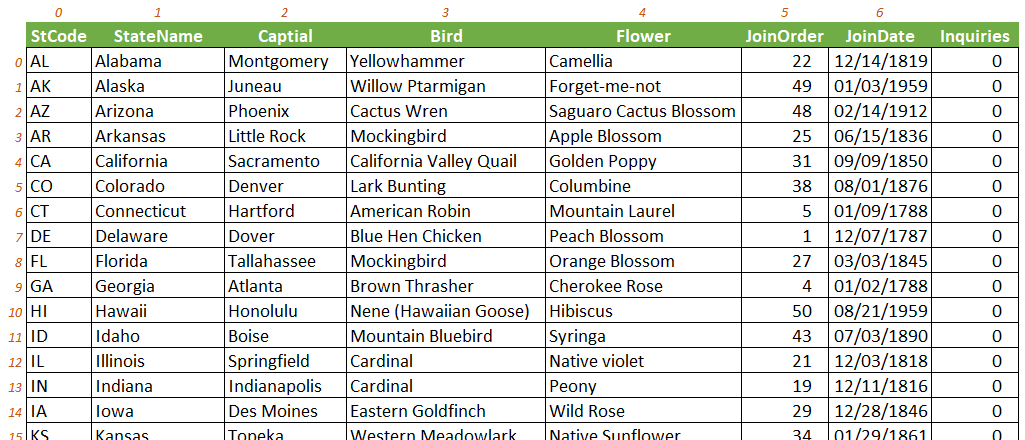
* The **DEST\_DETAILS** list is a basic (1-dimensional) list that is related to the *DESTINATIONS* list (i.e. it is a *parallel list*). That is, position 0 in both lists contains data about the same destination. This list is provided in the downloaded code.

A picture containing table

Description automatically generated

## States List

* The STATES list is a *2-dimensional list* with data for all 50 states. You will loop through this list looking for the state of interest. This list is provided in the downloaded code.



# Instructions

Note: with the exception of the main() function, you are not required to have any additional functions as that would add additional complexity.

1. Download the **starting file** for this assignment and write your code there.
2. Create the *DESTINATIONS list* – copy the data provided above. **Do not include the element numbers in the list.** For readability, place each destination on a different line. The code for the other 2 lists are provided for you.
3. Initialize variables
   * Within main(), create a 1D list called **state\_views** that will have 50 elements all preinitialized to 0. This requires only 1 line of code. Do not manually type 50 zeroes – see class notes. *Try* not to hardcode the 50 and determine it dynamically based on the numbers of rows in the STATES list.
4. Display the company header.
5. Display **Destinations Menu** - estimated lines of code: 7
   * Loop through the *destinations* list to display each destination and preface each destination with a number. These choices must not be hardcoded and must come from the list.
   * After the loop, display the exit choice.
6. Handle **menu selection**.

This section involves: 1) prompting for a menu selection, 2) bonus: validating the input, and 3) exiting, if requested. TIP: use 2 variable names here for clarity: **menu selection** and **destination number**.

* + *Prompt the user for a menu selection*. Since the **menu selection** can be a number or ‘X’, make sure that you do not yet convert the menu selection to an int() or your code will crash. For now, you may skip the validation & exiting (as this can be tricky) but do not forget to return to this step and handle the validation and exit.
    - *Validation (BONUS: +5 pts): w*rite an input validation loop to validate the menu selection (as you learned in Ch 4). If the menu selection is invalid, let the user know and keep looping until they enter a valid menu selection. String testing methods which return a Boolean might be helpful – any\_str.isalpha(), any\_str.isdigit(), any\_str.isspace(). Test values: 1 through 7, X, 9, a, user entered a space, user simply hit enter and left it blank (i.e. ‘’), etc.
    - *Exit (REQUIRED): o*nce the validation loop is exited, you can then test if they entered an ‘X’ and, if so, exit the application (as you learned in the Ch 3 homework). If the user exits, make sure to thank then for using the application.
  + After this point (i.e. you are left with a valid number), convert the **menu selection** which is a string to a **destination number** (i.e. an int). Destination number is the variable to be used in the rest of your code.

1. **Display Destinations Details** - estimated lines of code: 4-6
   * Using the destination number that the user provided, look up the destination name in the *destinations* list. Since the destinations list and the destination details list are parallel lists, you can just go **directly** to the *destination details* and read the correct element. IF statements are not allowed here.
   * You will only need one print statement to display the details; however, since the details are long, do the following to invoke text wrapping:
     + e.g. **textwrap**.**fill**(any\_str, 25) 🡪 this wraps a string after 25 characters, adjust accordingly.
     + *note: textwrap* is a module that is imported, and *fill* is a function in that module.
2. **Display the State Details** - estimated lines of code: 15-18
   * Prompt the user if they want to view the destination’s state details.
   * You will need to *extract the state code* **from the selected destination**. Make sure you are referencing only the *selected* destination (and not the entire destinations list).
     + Tip: Use string *slicing* to do extraction. For example, assume:  
       any\_str = 'baylor.edu' the code any\_str[-3:] will yield 'edu'.
   * Loop through the *states list* looking for the state of interest. I recommend that you use a ***for*** *loop*.
     + Know what you are working with in a *for* loop – i.e., **each iteration variable is looking at only one row** at a time (i.e. a one-dimensional list as shown below); thus, you only need one subscript to access a value in it.

Chart

Description automatically generated with low confidence

* + - Look for **the correct state**. When found, display its details properly formatted.
    - ***Views*** – this field is not in the state details list. Instead create a simple 1D list with 50 elements preinitialized to zero – do this at the top of main(). The purpose of the list is to hold a count of how many times each state’s details have been *previously* viewed. Thus, the first time a state is displayed, the number of *views* is 0; however, once the state is viewed (i.e. in this loop), the value ***in the state views list*** should be immediately increased by one – you will have to update the correct element (i.e. position 0 would be the number of views for Alabama).

1. **Pause processing - wait for user keypress**
   * See line 42 in the execution screenshot. Simply use an input() function to display “Press enter to continue…” so that it halts execution until the user presses any key. You will not need to assign a variable.
   * After the user presses enter, the company header and menu should be re-displayed; thus, you need all of your code inside of an outer while loop (e.g. menu\_choice != ‘X’) – note: this loop will actually never end as the exit is handled later after a menu choice of ‘x’.

# Submitting the Assignment

* First, review the ***Assignment Tips & Updates*** page to see if there are any additional requirements to this assignment – denoted with the tag: **[required].**
* To submit the assignment, go to Canvas and open the assignment link.
* Upload your \*.py file**.** You may upload more than once as long as it is before the due date & time. Only the final submission will be graded.