Lists Challenge 6: Grade Sorter App

Description:
You are responsible for writing a program that will collect four grades from a user. Your program will then sort these grades from highest to lowest. Then, your program will simulate dropping the lowest two grades the user entered. Lastly, it will comment on the users highest grade.

Step By Step Guide:

- Print a welcome message.
- Create a blank list called grades.
- Get user input to add 4 grades to the list.
    - Be aware of the data type you are using.

- Print the list as formatted below.
    - You may have to cast the list to a string.

- Permanently sort the list from highest to lowest.
- Print the list as formatted below.

- Inform the user that their lowest two grades are being dropped.
- Remove the lowest two grades from the list.
- Print a message informing the user that the grades were dropped.

- Print the remaining grades.
- Print the users highest grade with a message.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Grade Sorter App

What is your first grade (0-100): 82
What is your second grade (0-100): 95
What is your third grade (0-100): 100
What is your fourth grade (0-100): 61

Your grades are: [82, 95, 100, 61]

Your grades from highest to lowest are: [100, 95, 82, 61]

The lowest two grades will now be dropped.
Removed grade: 61
Removed grade: 82

Your remaining grades are: [100, 95]
Nice work! Your highest grade is a 100.

Lists Challenge 7: Different Types of Lists Program

Description:
You are responsible for writing a program that will highlight the similarities and differences between four different types of lists: a list of strings, a list of integers, a list of floats, and a list of lists. For each list, your program will describe the data type of the list, the elements of the list, and the data type of the first element in the list. Your program will then highlight the similarities and differences between sorting a list numerically and alphabetically.

Step By Step Guide:

- Define a list using a variable num_strings and "hard code" the following four numerical strings: "15", "100", "55", "42".
- Define a list using a variable num_ints and hard code the following four numerical integers: 15, 100, 55, 42.
- Define a list using a variable num_floats and hard code any four floats you want.
- Define a list using a variable num_lists. This is going to be a lists of lists or a nested list! Use the following syntax: num_lists = [[1,2,3], [4,5,6], [7,8,9]]

- Print a summary of each variable (or list). The summary should contain:
    - A statement about the variable's type.
    - A statement about the elements of the variable.
    - A statement about the first element and its type.
    - Use formatting below.

- Permanently sort num_strings and num_ints.
- Print each list.
- Print a statement about what you discover when sorting these two lists.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Summary Table

The variable num_strings is a <class 'list'>.
It contains the elements: ['15', '100', '55', '42']
The element 15 is a <class 'str'>.

The variable num_ints is a <class 'list'>.
It contains the elements: [15, 100, 55, 42]
The element 15 is a <class 'int'>.

The variable num_floats is a <class 'list'>.
It contains the elements: [2.2, 5.0, 1.245, 0.142857]
The element 2.2 is a <class 'float'>.

The variable num_lists is a <class 'list'>.
It contains the elements: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
The element [1, 2, 3] is a <class 'list'>.

Now sorting num_strings and num_ints...
Sorted num_strings: ['100', '15', '42', '55']
Sorted num_ints: [15, 42, 55, 100]

Strings are sorted alphabetically while integers are sorted numerically!

Lists Challenge 8: Grocery List App

Description:
You are responsible for writing a program that will simulate a grocery shopping list. Your program will start with two items on the shopping list, meat and cheese, and then allow a user to add three new items to the list. To simulate shopping, your program will ask the user what item they just purchased and then remove the item from the shopping list. Upon having only two items in the shopping list, your program will inform the user that the store is out of a particular item and prompt the user to replace the item with a new item. You will use the datetime library to display the current date and time the shopping is taking place in mm/dd hh:mm format.

Step By Step Guide:

- Define a list that will hold the foods you need to get at the grocery store.
    - Start by populating the list with two foods, Meat and Cheese.
- Print a welcome message.
- Print the current date and time.
    - This functionality is outside the scope of basic Python. We will have to import the datetime library to gain access to code that can perform this function.
    - Type import datetime as the first line of code in your program.
    - Create a datetime object using the datetime library and store the pertinent information in appropriately named variables using the following code:
        - time = datetime.datetime.now()
        - month = str(time.month)
        - day = str(time.day)
        - hour = str(time.hour)
        - minute = str(time.minute)
    - Use these new variables to print the current date and time.
- Print a message informing the user of the two foods in their grocery list.
- Get user input and append three new foods to the grocery list.
    - Make sure to title case any user input.
- Print the grocery list.
- Permanently sort the grocery list.
- Print the sorted grocery list.

- Simulate shopping by doing the following:
    - Print the current list length.
    - Print the current list.
    - Get user input for the food purchased.
        - The input should be case insensitive. The program should recognize Meat, meat, and MEAT all the same.
    - Remove the appropriate food from the list.

    - Do this three times.

- When there is only 2 foods left in the list, print the list and inform the user that the store is out of the last item in the list, making sure to remove this item from the list.
- Ask user what food they would like instead and insert this food at the beginning of the list.
- Print a final version of the grocery list.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Grocery List App.
Current Date and Time: 11/3 2:31
You currently have Meat and Cheese in your list.

Type of food to add to the grocery list: bananas
Type of food to add to the grocery list: apples
Type of food to add to the grocery list: soup

Here is your grocery list:
['Meat', 'Cheese', 'Bananas', 'Apples', 'Soup']
Here is your grocery list sorted:
['Apples', 'Bananas', 'Cheese', 'Meat', 'Soup']

Simulating grocery shopping...

Current grocery list: 5 items
['Apples', 'Bananas', 'Cheese', 'Meat', 'Soup']
What food did you just buy: apples
Removing Apples from list...

Current grocery list: 4 items
['Bananas', 'Cheese', 'Meat', 'Soup']
What food did you just buy: Cheese
Removing Cheese from list...

Current grocery list: 3 items
['Bananas', 'Meat', 'Soup']
What food did you just buy: soup
Removing Soup from list...

Current grocery list: 2 items

['Bananas', 'Meat']

Sorry, the store is out of Meat.
What food would you like instead: fish

Here is what remains on your grocery list:
['Fish', 'Bananas']

Lists Challenge 9: Basketball Roster Program

Description:
You are responsible for writing a program that will build and display a basketball roster based off user input. Your program will then simulate an injury to a specific player in the roster and prompt the user to update the roster. Upon updating the roster, your program will display the final roster and wish the newly add player good luck.

Step By Step Guide:

- Print a welcome message.
- Create a blank list called roster.
- Get user input for the names of the starting roster for a basketball team.
    - The starting roster includes a point guard, shooting guard, small forward, power forward, and center. Use an input statement for each position.
    - Make sure to always display the name capitalized.
    - Add each position to your list roster.
    - Your point guard should be index 0, shooting guard at index 1, ect...
- Print the starting 5 as formatted below.

- Remove the small forward from the list and store it in a variable called injured_player.
- Print a message to the user informing them that this player is injured.
- Get the length of the current roster.
- Print a message to the user informing them of the length of the current roster.

- Get user input for who will take the injured players spot and store this in a variable called added_player.
- Add this player to the roster at the correct position.

- Print the updated starting 5 as formatted below.
- Print a good luck message to the new player.
- Get the length of the current roster.
- Print a message to the user informing them of the length of the current roster.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

```
Welcome to the Basketball Roster Program

Who is your point guard:  mike eramo
Who is your shooting guard:  Klay Thompson
Who is your small forward:  LEBRON JAMES
```

Who is your power forward: Anthony DaVis
Who is your center: kevin getman

Your starting 5 for the upcoming basketball season

Point Guard:	Mike Eramo
Shooting Guard:	Klay Thompson
Small Forward:	Lebron James
Power Forward:	Anthony Davis
Center:	Kevin Getman

Oh no, Lebron James is injured.
Your roster only has 4 players.
Who will take Lebron James's spot: lucas eramo

Your starting 5 for the upcoming basketball season

Point Guard:	Mike Eramo
Shooting Guard:	Klay Thompson
Small Forward:	Lucas Eramo
Power Forward:	Anthony Davis
Center:	Kevin Getman

Good luck Lucas Eramo you will do great!
Your roster now has 5 players.

Lists Challenge 10: Favorite Teachers Program

Description:
You are responsible for writing a program that will create a list of a user's favorite teachers. It will display these teachers ranked (assuming the first teacher entered is the favorite, the second teacher entered is the next favorite, ect…), alphabetically, in reverse alphabetical order, the top two teachers, the next two teachers, the last favorite teacher, and the total number of favorite teachers in the list. Your program will then add and remove teachers from this list, each time displaying a similar summary.

Step By Step Guide:

- Print a welcome message.
- Create a blank list called fav_teachers.
- Prompt the user to enter in their four favorite teachers.
    - The user should only enter the last name of the teacher.
    - Make sure to title case all user input for sorting purposes.
- Add each of these teachers to the list fav_teachers.

- Print the favorite teachers ranked.
- Print the favorite teachers alphabetically.
- Print the favorite teachers in reverse alphabetical order.
- Print the top two teachers.
- Print the next two teachers.
- Print the last favorite teacher.
- Print the total number of favorite teachers.
- Follow the formatting below.

- Print a message to the user informing them that first favorite teacher is no longer their favorite teacher.
- Prompt the user to enter their new favorite teacher.
    - Store the name of this teacher in the first index of the list
    - Shift all other teachers down one index.

- Print the favorite teachers ranked.
- Print the favorite teachers alphabetically.
- Print the favorite teachers in reverse alphabetical order.
- Print the top two teachers.
- Print the next two teachers.
- Print the last favorite teacher.
- Print the total number of favorite teachers.
- Follow the formatting below.

- Print a message to the user informing them that they no longer like a teacher.
- Prompt the user to pick a teacher to remove from the list.

- Remove this teacher.

- Print the favorite teachers ranked.
- Print the favorite teachers alphabetically.
- Print the favorite teachers in reverse alphabetical order.
- Print the top two teachers.
- Print the next two teachers.
- Print the last favorite teacher.
- Print the total number of favorite teachers.
- Follow the formatting below.

- Use at least 2 comments to describe sections of your code
- “Chunk” your so that it is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output:

Welcome to the Favorite Teachers Program

Who is your first favorite teacher: Eramo
Who is your second favorite teacher: ricco
Who is your third favorite teacher: gates
Who is your fourth favorite teacher: foote

Your favorite teachers ranked are: ['Eramo', 'Ricco', 'Gates', 'Foote']
Your favorite teachers alphabetically are: ['Eramo', 'Foote', 'Gates', 'Ricco']
Your favorite teachers in reverse alphabetical order are: ['Ricco', 'Gates', 'Foote', 'Eramo']

Your top two teachers are: Eramo and Ricco.
Your next two favorite teachers are: Gates and Foote.
Your last favorite teacher is: Foote.
You have a total of 4 favorite teachers.

Oops, Eramo is no longer your first favorite teacher. Who is your new FAVORITE teacher: marley

Your favorite teachers ranked are: ['Marley', 'Eramo', 'Ricco', 'Gates', 'Foote']
Your favorite teachers alphabetically are: ['Eramo', 'Foote', 'Gates', 'Marley', 'Ricco']
Your favorite teachers in reverse alphabetical order are: ['Ricco', 'Marley', 'Gates', 'Foote', 'Eramo']

Your top two teachers are: Marley and Eramo.
Your next two favorite teachers are: Ricco and Gates.
Your last favorite teacher is: Foote.
You have a total of 5 favorite teachers.

You've decided you no longer like a teacher. Which teacher would you like to remove from you list: eramo

Your favorite teachers ranked are: ['Marley', 'Ricco', 'Gates', 'Foote']
Your favorite teachers alphabetically are: ['Foote', 'Gates', 'Marley', 'Ricco']
Your favorite teachers in reverse alphabetical order are: ['Ricco', 'Marley', 'Gates', 'Foote']

Your top two teachers are: Marley and Ricco.
Your next two favorite teachers are: Gates and Foote.
Your last favorite teacher is: Foote.
You have a total of 4 favorite teachers.

Example Screenshots: