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']
```