**Python Journal Template**

**Directions:** Follow the directions for each part of the journal template. Include in your response all the elements listed under the Requirements section. Prompts in the Inspiration section are not required; however, they may help you to fully think through your response.

Remember to review the Touchstone page for entry requirements, examples, and grading specifics.

**Name:**

**Date:**

**Final Replit Program Share Link:**

Complete the following template. Fill out all entries using complete sentences.

PART 1: Defining Your Problem

|  |
| --- |
| **Task**  State the problem you are planning to solve.  **Requirements**   * Describe the problem you are trying to solve for. * Describe any input data you expect to use. * Describe what the program will do to solve the problem. * Describe any outputs or results the program will provide.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Why do you want to solve this particular problem? * What source(s) of data do you believe you will need? Will the user need to supply that data, or will you get it from an external file or another source? * Will you need to interact with the user throughout the program? Will users continually need to enter data in and see something to continue? * What are your expected results or what will be the end product? What will you need to tell a user of your program when it is complete? |
| <Write your journal entry response here>  Restock is an application that allows users to keep track of common ingredients or food in a household refrigerator and pantry, add in ingredients/food that are unique to their household. Many times, there is frustration when one is at the store and they forgot if they still have certain items in stock and this is especially true for food and groceries. Restock will alleviate that stress by keeping track of common ingredients that are stored in the pantry or fridge. There will already be set default categories and ingredients and the user just has to choose which ingredient they want to add and the quantity they want to add. Users will also be able to add in specific ingredients unique to their household or their taste or whatever ingredients that aren't already in the program by default. Likewise, the user can also remove any ingredient they want whether that be for dietary reasons or simply because they don't eat certain foods. Restock will output the stock of certain items when prompted so the user can also see how much they have left of each ingredient or they can view all the ingredients at once if they wanted to. Once started, the application will run in an endless loop. The code is written in such a way that allows for the automatic updating of data. |

PART 2: Working Through Specific Examples

|  |
| --- |
| **Task**  Write down clear and specific steps to solve a simple version of your problem you identified in Part 1.  **Requirements**  Complete the three steps below **for at least two distinct examples/scenarios**.   * State any necessary input data for your simplified problem. * Write clear and specific steps in English (not Python) detailing what the program will do to solve the problem. * Describe the specific result of your example/scenario.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Are there any steps that you don’t fully understand? These are places to spend more time working out the details. Consider adding additional smaller steps in these spots. * Remember that a computer program is very literal. Are there any steps that are unclear? Try giving the steps of your example/scenario to a friend or family member to read through and ask you questions about parts they don’t understand. Rewrite these parts as clearly as you can. * Are there interesting edge cases for your program? Try to start one of your examples/scenarios with input that matches this edge case. How does it change how your program might work? |
| <Write your journal entry response here>  First, the program will prompt the user with five options. To add quantity, remove quantity, view ingredient stock, add unique ingredients, and remove an ingredient entirely. With both add and remove, the user can then choose between adding or removing from the pantry or the refrigerator. Then depending on the whether the user chooses fridge or pantry, there will be specific categories within the pantry and fridge. Each category will have their own ingredients. For example, grains will be in the pantry and will have certain items like rice, oats, etc. The meat category will be in the fridge section and will have beef, pork, etc. So on, and so forth. Within pantry or refrigerator, the user can add quantities to multiple categories and ingredients at once so they don't have to repeat the process for each and every category or ingredient. Within the view option, the user will be able to view all the ingredients at once along with their quantity or they can view each category separately. Edge cases will be taken care of since this program will allow customization of unique or uncommon ingredients. In cases where the user inputs in the wrong data, for example when prompted to choose between pantry or refrigerator and they choose something other than those two options, then the program will prompt the user to try again. |

PART 3: Generalizing Into Pseudocode

|  |
| --- |
| **Task**  Write out the general sequence your program will use, including all specific examples/scenarios you provided in Part 2.  **Requirements**   * Write pseudocode for the program in English but refer to Python program elements where they are appropriate. The pseudocode should represent the full functionality of the program, not just a simplified version. Pseudocode is broken down enough that the details of the program are no longer in any paragraph form. One statement per line is ideal.   **Help with writing pseudocode**   * Here are a few links that can help you write pseudocode with examples. Remember to check out part 3 of the Example Journal Template Submission if you have not already. Note: everyone will write pseudocode differently. There is no right or wrong way to write it other than to make sure you write it clearly and in as much detail as you can so that it should be easy to convert it to code later. * <https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/> * <https://www.wikihow.com/Write-Pseudocode>   **Inspiration**  When writing your entry below ask yourself the following questions:   * Do you see common program elements and patterns in your specific examples/scenarios in Part 2, like variables, conditionals, functions, loops, and classes? These should be part of your pseudocode for the general sequence as well. * Are there places where the steps for your examples/scenarios in Part 2 diverged? These may be places where errors may occur later in the project. Make note of them. * When you are finished with your pseudocode, does it make sense, even to a person that does not know Python? Aim for the clearest description of the steps, as this will make it easier to convert into program code later. |
| <Write your pseudocode here>  Global Variables  Pantry Categories |

PART 4: Testing Your Program

|  |
| --- |
| **Task**  While writing and testing your program code, describe your tests, record any errors, and state your approach to fixing the errors.  **Requirements**   * For at least one of your test cases, describe how your choices for the test helped you understand whether the program was running correctly or not.   For each error that occurs while writing and testing your code:   * Record the details of the error from Replit. A screenshot or copy-and-paste of the text into the journal entry is acceptable. * Describe what you attempted in order to fix the error. Clearly identify what approach was the one that worked.   **Inspiration**  When writing your entry below ask yourself the following questions:   * Have you tested edge cases and special cases for the inputs of your program code? Often these unexpected values can cause errors in the operation of your program. * Have you tested opportunities for user error? If a user is asked to provide an input, what happens when they give the wrong type of input, like a letter instead of a number, or vice versa? * Did the outcome look the way you expected? Was it formatted correctly? * Does your output align with the solution to the problem you coded for? |
| <Record your errors and fixes here> |

PART 5: Commenting Your Program

|  |
| --- |
| **Task**  Submit your full program code, including thorough comments describing what each portion of the program should do when working correctly.  **Requirements**   * The purpose of the program and each of its parts should be clear to a reader that does not know the Python programming language.   **Inspiration**  When writing your entry, you are encouraged to consider the following:   * Is each section or sub-section of your code commented to describe what the code is doing? * Give your code with comments to a friend or family member to review. Add additional comments to spots that confuse them to make it clearer. |
| from math import \*  from datetime import \*  #create a list for pantry and refigerator categories  pantryCategories = ['Canned or Jarred','Grains','Oils and Vinegar', 'Spices, Herbs, and Seasoning','Vegetables','Baking Ingredients','Sweeteners','Snacks and Cereal','Condiments',]  refrigeratorCategories = ['Dairy and Eggs', 'Vegetables','Fruits', 'Meat and Fish',]  #create parent lists per category in the pantry list above, and then a list within that list per common ingredients in the pantry. Within that list are the name of the ingredients and the amount  pantry = [  [['Tomatoes', 0],['Beans', 0],['Salsa', 0],['Tuna', 0],['Chicken', 0],['Sardines', 0],['Salmon', 0],['Spam', 0],['Chili', 0],['Soup', 0],['Corn', 0],['Cheese', 0],['Milk', 0],],  [['Long-Grain White Rice', 0],['Brown Rice', 0],['Pasta', 0],['Polenta', 0],['Plain Breadcrubs', 0],['Quinoa', 0],['Coucous', 0],['Farro', 0],['Rice Noodles', 0],['Egg Noodles', 0],['Flat Bread', 0]],  [['Extra Virgin Olive Oil', 0],['Vegetable Oil', 0],['Apple Cider Vinegar', 0],['Red Wine Vinegar', 0],['Balsamic Vinegar', 0],['Rice Vinegar', 0],['Sesame Oil', 0]],  [['Cayenne Pepper', 0],['Chili Powder', 0],['Cinnamon', 0],['Crushed Red Pepper', 0],['Cumin', 0],['Garlic Powder', 0],['Salt', 0],['Mutmeg', 0],['Onion Powder', 0],['Ground Ginger', 0],['Oregano', 0],['Paprika', 0],[' Rosemary', 0],[' Sesame Seeds', 0],['Thyme', 0],['Tumeric', 0],['Vanilla Extract', 0],],  [['Potatoes', 0],['Onions', 0],['Shallots', 0],['Garlic', 0],['Yams', 0],['Sweet Potatoes', 0],],  [['All Purpose Flour', 0],['Baking Soda', 0],['Baking Powder', 0],['Pure vanilla extract', 0],],  [['Sugar', 0],['Brown Sugar', 0],['Maple Syrup', 0],['Honey', 0],],  [['Crackers', 0],['Chips', 0],['Cookies', 0],['Biscuits', 0],['Marshmellows', 0],['Popcorn', 0],['Seeds', 0],['Peanut Butter', 0],['Breakfast Cereal', 0],['Oats', 0],],  [['Ketchup', 0],['Mayo', 0],['Mustard', 0],['Hot Sauce', 0],['Worcestershire', 0],['Soy Sauce', 0],['Fish Sauce', 0],['Peanut Butter', 0],['Breakfast Cereal', 0],['Oats', 0],]  ]  #create parent lists per category in the refrigerator list above, and then a list within that list per common ingredients in the refrigerator. Within that list are the name of the ingredients and the amount  refrigerator = [  [['Milk', 0],['Almond or Soy Milk',0],['Eggs', 0],['Shredded Cheese', 0],['Butter', 0],['Sour Cream', 0],['Cheese', 0],],  [['Lettuce', 0],['Spinach', 0],['Kale', 0],['Pepper', 0],['Button Mushrooms', 0],['Portobello Mushrooms', 0],['Shiitake Mushrooms', 0],['Green Onions', 0],],  [['Apples', 0],['Bananas', 0],['Oranges', 0],['Grapes', 0],],  [['Chicken Breast', 0],['Ground Beef',0],['Ground Turkey',0],['Salmon',0],['Shrimp',0],['Ham', 0],['Bacon', 0],['Turkey', 0]]  ]  #create functions to automatically count how many unique ingredients are in each category  pantryCategoryListCount = len(pantryCategories)  canOrJarIngredientCount = len(pantry[0])  grainsIngredientCount = len(pantry[1])  oilsAndvinegarIngredientCount = len(pantry[2])  spicesHerbsSeasoningIngredientCount = len(pantry[3])  pantryVegetablesIngredientCount = len(pantry[4])  bakingIngredientCount = len(pantry[5])  sweetenersIngredientCount = len(pantry[6])  snacksCerealIngredientCount = len(pantry[7])  condimentsIngredientCount = len(pantry[8])  refrigeratorCategoryListCount = len(refrigeratorCategories)  dairyEggsIngredientCount = len(refrigerator[0])  refrigeratorVegetablesIngredientCount = len(refrigerator[1])  fruitsIngredientCount = len(refrigerator[2])  meatFishIngredientCount = len(refrigerator[3])  #function to prompt the user to try again if they made a mistake in their input  def tryAgain():  print('Please try again.')  #option to choose between pantry or refrigerator  def pantryOrRefrigerator():  print('Pantry or Refrigerator?')  print('1. Pantry\n')  print('2. Refrigerator\n')  return ''  #print out each category in the pantry  def pantryCategoriesList():  number = 1  for x in pantryCategories:  print('{}.'.format(number), x)  number += 1  print('Enter a number for the category you want to add.')  print('Press ENTER once to add another category.')  print('Press ENTER twice to continue.')  return number  #function to print out a list of each ingredient in a category and their quantity plus the common unit of measure if needed.  def canOrJarredIngredientsList(pantry):  number = 1  for x in pantry[0]:  print('{}.'.format(number), x[0], ': ', x[1])  number += 1  def grainsIngredientsList(pantry):  number = 1  for x in pantry[1]:  print('{}.'.format(number), x[0], ': ', x[1],'oz')  number += 1  def oilsAndVinegarList(pantry):  number = 1  for x in pantry[2]:  print('{}.'.format(number), x[0], ': ', x[1],'oz')  number += 1  def spicesList(pantry):  number = 1  for x in pantry[3]:  print('{}.'.format(number), x[0], ': ', x[1],'oz')  number += 1  def pantryVegetablesList(pantry):  number = 1  for x in pantry[4]:  print('{}.'.format(number), x[0], ': ', x[1])  number += 1  def bakingList(pantry):  number = 1  for x in pantry[5]:  print('{}.'.format(number), x[0], ': ', x[1],'oz')  number += 1  def sweetenerList(pantry):  number = 1  for x in pantry[6]:  print('{}.'.format(number), x[0], ': ', x[1],'oz')  number += 1  def snacksCerealList(pantry):  number = 1  for x in pantry[7]:  print('{}.'.format(number), x[0], ': ', x[1])  number += 1  def condimentList(pantry):  number = 1  for x in pantry[8]:  print('{}.'.format(number), x[0], ': ', x[1])  number += 1  #print out each category in the refigerator  def refrigeratorCategoriesList():  number = 1  for x in refrigeratorCategories:  print('{}.'.format(number), x)  number += 1  print('Enter a number for the category you want to add.')  print('Press ENTER once to add another category.')  print('Press ENTER twice to continue.')  return number  def dairyAndEggsList(refrigerator):  number = 1  for x in refrigerator[0]:  if 'cheese' in str(x).lower():  print('{}.'.format(number), x[0], ': ', x[1], 'oz')  number += 1  elif 'butter' in str(x).lower():  print('{}.'.format(number), x[0], ': ', x[1], 'oz')  number += 1  elif 'sour cream' in str(x).lower():  print('{}.'.format(number), x[0], ': ', x[1], 'oz')  number += 1  else:  print('{}.'.format(number), x[0], ': ', x[1], 'cartons')  number += 1  def refrigeratorVegetablesList(refrigerator):  number = 1  for x in refrigerator[1]:  print('{}.'.format(number), x[0], ': ', x[1], 'grams')  number += 1  def fruitsList(refrigerator):  number = 1  for x in refrigerator[2]:  print('{}.'.format(number), x[0], ': ', x[1], 'grams')  number += 1  def meatAndFishList(refrigerator):  number = 1  for x in refrigerator[3]:  print('{}.'.format(number), x[0], ': ', x[1], 'lbs')  number += 1  #function to choose one or multiple categories in the pantry or refrigerator (depending on what user asked for) to add ingredient quantities to  def chooseCategories(panOrRef):  categoriesList = []  Stop = False  AddCount = 0  if panOrRef == 1:  typeCount = pantryCategoriesList()  typeCount = typeCount - 1  elif panOrRef == 2:  typeCount = refrigeratorCategoriesList()  typeCount = typeCount - 1  while Stop == False:  categories = input()  if categories.isnumeric():  categories = int(categories)  if categories > 0 and categories <= typeCount:  categoriesList.append(categories)  AddCount += 1  else:  tryAgain()  elif AddCount == 0 and categories == '':  tryAgain()  elif categories == '':  Stop = True  categoriesList = list(set(categoriesList))  categoriesList.sort()  return categoriesList  #function to view category specific quantities  def viewCategories(panOrRef):  viewList = []  viewCount = 0  stop = False  while viewCount == 0:  if panOrRef == 1:  typeCount = pantryCategoriesList()  typeCount -= 1  if panOrRef == 2:  typeCount = refrigeratorCategoriesList()  typeCount -= 1  while stop == False:  categories = input()  if categories.isnumeric():  categories = int(categories)  if categories > 0 and categories <= typeCount:  viewList.append(categories)  viewCount += 1  else:  tryAgain()  elif viewCount == 0 and categories == '':  tryAgain()  elif categories == '':  stop = True  viewList = list(set(viewList))  viewList.sort()  return viewList  #function to add quantities  def addQuantity(panOrRef, categoriesList, stop):  for ingredientType in categoriesList:  ingredientType -= 1  if ingredientType == 0 and panOrRef == 1:  canOrJarredIngredientsList(pantry)  elif ingredientType == 1 and panOrRef == 1:  grainsIngredientsList(pantry)  elif ingredientType == 2 and panOrRef == 1:  oilsAndVinegarList(pantry)  elif ingredientType == 3 and panOrRef == 1:  spicesList(pantry)  elif ingredientType == 4 and panOrRef == 1:  pantryVegetablesList(pantry)  elif ingredientType == 5 and panOrRef == 1:  bakingList(pantry)  elif ingredientType == 6 and panOrRef == 1:  sweetenerList(pantry)  elif ingredientType == 7 and panOrRef == 1:  snacksCerealList(pantry)  elif ingredientType == 8 and panOrRef == 1:  condimentList(pantry)  elif ingredientType == 0 and panOrRef == 2:  dairyAndEggsList(refrigerator)  elif ingredientType == 1 and panOrRef == 2:  refrigeratorVegetablesList(refrigerator)  elif ingredientType == 2 and panOrRef == 2:  fruitsList(refrigerator)  elif ingredientType == 3 and panOrRef == 2:  meatAndFishList(refrigerator)  count = 0  stop = False  while stop == False:  if ingredientType == 0 and panOrRef == 1:  ingredientCount = canOrJarIngredientCount  elif ingredientType == 1 and panOrRef == 1:  ingredientCount = grainsIngredientCount  elif ingredientType == 2 and panOrRef == 1:  ingredientCount = oilsAndvinegarIngredientCount  elif ingredientType == 3 and panOrRef == 1:  ingredientCount = spicesHerbsSeasoningIngredientCount  elif ingredientType == 4 and panOrRef == 1:  ingredientCount = pantryVegetablesIngredientCount  elif ingredientType == 5 and panOrRef == 1:  ingredientCount = bakingIngredientCount  elif ingredientType == 6 and panOrRef == 1:  ingredientCount = sweetenersIngredientCount  elif ingredientType == 7 and panOrRef == 1:  ingredientCount = snacksCerealIngredientCount  elif ingredientType == 8 and panOrRef == 1:  ingredientCount = condimentsIngredientCount  elif ingredientType == 0 and panOrRef == 2:  ingredientCount = dairyEggsIngredientCount  elif ingredientType == 1 and panOrRef == 2:  ingredientCount = refrigeratorVegetablesIngredientCount  elif ingredientType == 2 and panOrRef == 2:  ingredientCount = fruitsIngredientCount  elif ingredientType == 3 and panOrRef == 2:  ingredientCount = meatFishIngredientCount    ingredientName = input('What ingredient do you want to add (press ENTER for next or when done):\n')  if count == 0 and ingredientName == '':  tryAgain()  elif count == 0 and ingredientName == '0':  tryAgain()  elif count > 0 and ingredientName == '':  stop = True  elif ingredientName.isnumeric():  ingredientName = int(ingredientName)  quantityStop = False  if ingredientName > 0 and ingredientName <= ingredientCount:  ingredientName -= 1  count += 1  quantityStop = False  while quantityStop == False:  quantity = (input('How many?: '))  if quantity == '' and count < 1:  tryAgain()  elif quantity.isnumeric():  if panOrRef == 1:  pantry[ingredientType][ingredientName][1] = pantry[ingredientType][ingredientName][1] + int(quantity)  quantityStop = True  elif panOrRef == 2:  refrigerator[ingredientType][ingredientName][1] = refrigerator[ingredientType][ingredientName][1] + int(quantity)  quantityStop = True  elif quantity == '':  stop = True  #function to remove quantities  def removeQuantity(panOrRef, categoriesList, stop):  for ingredientType in categoriesList:  ingredientType -= 1  if ingredientType == 0 and panOrRef == 1:  canOrJarredIngredientsList(pantry)  elif ingredientType == 1 and panOrRef == 1:  grainsIngredientsList(pantry)  elif ingredientType == 2 and panOrRef == 1:  oilsAndVinegarList(pantry)  elif ingredientType == 3 and panOrRef == 1:  spicesList(pantry)  elif ingredientType == 4 and panOrRef == 1:  pantryVegetablesList(pantry)  elif ingredientType == 5 and panOrRef == 1:  bakingList(pantry)  elif ingredientType == 6 and panOrRef == 1:  sweetenerList(pantry)  elif ingredientType == 7 and panOrRef == 1:  snacksCerealList(pantry)  elif ingredientType == 8 and panOrRef == 1:  condimentList(pantry)  elif ingredientType == 0 and panOrRef == 2:  dairyAndEggsList(refrigerator)  elif ingredientType == 1 and panOrRef == 2:  refrigeratorVegetablesList(refrigerator)  elif ingredientType == 2 and panOrRef == 2:  fruitsList(refrigerator)  elif ingredientType == 3 and panOrRef == 2:  meatAndFishList(refrigerator)  count = 0  stop = False  while stop == False:  if ingredientType == 0 and panOrRef == 1:  ingredientCount = canOrJarIngredientCount  elif ingredientType == 1 and panOrRef == 1:  ingredientCount = grainsIngredientCount  elif ingredientType == 2 and panOrRef == 1:  ingredientCount = oilsAndvinegarIngredientCount  elif ingredientType == 3 and panOrRef == 1:  ingredientCount = spicesHerbsSeasoningIngredientCount  elif ingredientType == 4 and panOrRef == 1:  ingredientCount = pantryVegetablesIngredientCount  elif ingredientType == 5 and panOrRef == 1:  ingredientCount = bakingIngredientCount  elif ingredientType == 6 and panOrRef == 1:  ingredientCount = sweetenersIngredientCount  elif ingredientType == 7 and panOrRef == 1:  ingredientCount = snacksCerealIngredientCount  elif ingredientType == 8 and panOrRef == 1:  ingredientCount = condimentsIngredientCount  elif ingredientType == 0 and panOrRef == 2:  ingredientCount = dairyEggsIngredientCount  elif ingredientType == 1 and panOrRef == 2:  ingredientCount = refrigeratorVegetablesIngredientCount  elif ingredientType == 2 and panOrRef == 2:  ingredientCount = fruitsIngredientCount  elif ingredientType == 3 and panOrRef == 2:  ingredientCount = meatFishIngredientCount    ingredientName = input('What ingredient do you want to remove (press ENTER for next or when done):\n')  if count == 0 and ingredientName == '':  tryAgain()  elif count == 0 and ingredientName == '0':  tryAgain()  elif count > 0 and ingredientName == '':  stop = True  elif ingredientName.isnumeric():  ingredientName = int(ingredientName)  quantityStop = False  if ingredientName > 0 and ingredientName <= ingredientCount:  ingredientName -= 1  count += 1  quantityStop = False  while quantityStop == False:  quantity = (input('How many?: '))  if quantity == '' and count < 1:  tryAgain()  elif quantity > pantry[ingredientType][ingredientName][1]:  print('Please pick a number less than or equal to')  elif quantity.isnumeric() and quantity <= pantry[ingredientType][ingredientName][1]:  pantry[ingredientType][ingredientName][1] = pantry[ingredientType][ingredientName][1] - int(quantity)  quantityStop = True  elif quantity == '':  stop = True  #function to view quantities  def viewQuantity(panOrRef, viewList):  for x in viewList:  x -= 1  if x == 0 and panOrRef == 1:  canOrJarredIngredientsList(pantry)  elif x == 1 and panOrRef == 1:  grainsIngredientsList(pantry)  elif x == 2 and panOrRef == 1:  oilsAndVinegarList(pantry)  elif x == 3 and panOrRef == 1:  spicesList(pantry)  elif x == 4 and panOrRef == 1:  pantryVegetablesList(pantry)  elif x == 5 and panOrRef == 1:  bakingList(pantry)  elif x == 6 and panOrRef == 1:  sweetenerList(pantry)  elif x == 7 and panOrRef == 1:  snacksCerealList(pantry)  elif x == 8 and panOrRef == 1:  condimentList(pantry)  elif x == 0 and panOrRef == 2:  dairyAndEggsList(refrigerator)  elif x == 1 and panOrRef == 2:  refrigeratorVegetablesList(refrigerator)  elif x == 2 and panOrRef == 2:  fruitsList(refrigerator)  elif x == 3 and panOrRef == 2:  meatAndFishList(refrigerator)    #function to choose between pantry or refrigerator  def choosePanOrRef(answer):  addCount = 0  while addCount == 0:  panOrRef = input(pantryOrRefrigerator())  if panOrRef.isnumeric():  panOrRef = int(panOrRef)  if panOrRef == 1:  if answer == 1:  categoriesList = chooseCategories(panOrRef)  stop = False  addQuantity(panOrRef, categoriesList, stop)  addCount += 1  elif answer == 2:  categoriesList = chooseCategories(panOrRef)  stop = False  removeQuantity(panOrRef, categoriesList, stop)  addCount += 1  elif answer == 4:  categoriesList = chooseCategories(panOrRef)  stop = False  addUnique(panOrRef, categoriesList)  addCount += 1  elif answer == 5:  categoriesList = chooseCategories(panOrRef)  stop = False  removeUnique(panOrRef, categoriesList)  addCount += 1  elif panOrRef == 2:  if answer == 1:  categoriesList = chooseCategories(panOrRef)  stop = False  addQuantity(panOrRef, categoriesList, stop)  addCount += 1  elif answer == 2:  categoriesList = chooseCategories(panOrRef)  stop = False  removeQuantity(panOrRef, categoriesList, stop)  addCount += 1  elif answer == 4:  categoriesList = chooseCategories(panOrRef)  stop = False  addUnique(panOrRef, categoriesList)  addCount += 1  elif answer == 5:  categoriesList = chooseCategories(panOrRef)  stop = False  for x in categoriesList:  x -= 1  if x == 0:  dairyAndEggsList(refrigerator)  elif x == 1:  refrigeratorVegetablesList(refrigerator)  elif x == 2:  fruitsList(refrigerator)  elif x ==3:  meatAndFishList(refrigerator)  removeUnique(panOrRef, categoriesList)  addCount += 1  else:  print('Please try again')  else:  tryAgain()    #add a unique ingredient to the pantry or refrigerator  def addUnique(panOrRef, categoriesList,):  for x in categoriesList:  x -= 1  global pantry  global refrigerator  if panOrRef == 1:  newIngredient = input('What new ingredient do you want to add to Pantry:{}?:\n'.format(pantryCategories[x]))  elif panOrRef == 2:  newIngredient = input('What new ingredient do you want to add to Refrigerator:{}?:\n'.format(refrigeratorCategories[x]))  if panOrRef == 1:  if x == 0:  global canOrJarIngredientCount  pantry[x].append([newIngredient, 0])  canOrJarIngredientCount += 1  canOrJarredIngredientsList(pantry)  elif x == 1:  global grainsIngredientCount  pantry[x].append([newIngredient, 0])  grainsIngredientCount += 1  grainsIngredientsList(pantry)  elif x == 2:  global oilsAndvinegarIngredientCount  pantry[x].append([newIngredient, 0])  oilsAndvinegarIngredientCount += 1  oilsAndVinegarList(pantry)  elif x == 3:  global spicesHerbsSeasoningIngredientCount  pantry[x].append([newIngredient, 0])  spicesHerbsSeasoningIngredientCount += 1  spicesList(pantry)  elif x == 4:  global pantryVegetablesIngredientCount  pantry[x].append([newIngredient, 0])  pantryVegetablesIngredientCount += 1  pantryVegetablesList(pantry)  elif x == 5:  global bakingIngredientCount  pantry[x].append([newIngredient, 0])  bakingIngredientCount += 1  bakingList(pantry)  elif x == 6:  global sweetenersIngredientCount  pantry[x].append([newIngredient, 0])  sweetenersIngredientCount += 1  sweetenerList(pantry)  elif x == 7:  global snacksCerealIngredientCount  pantry[x].append([newIngredient, 0])  snacksCerealIngredientCount += 1  snacksCerealList(pantry)  elif x == 8:  global condimentsIngredientCount  pantry[x].append([newIngredient, 0])  condimentsIngredientCount += 1  condimentList(pantry)  elif panOrRef == 2:  if x == 0:  global dairyEggsIngredientCount  refrigerator[x].append([newIngredient, 0])  dairyEggsIngredientCount += 1  dairyAndEggsList(refrigerator)  elif x == 1:  global refrigeratorVegetablesIngredientCount  refrigerator[x].append([newIngredient, 0])  refrigeratorVegetablesIngredientCount += 1  refrigeratorVegetablesList(refrigerator)  elif x == 2:  global fruitsIngredientCount  refrigerator[x].append([newIngredient, 0])  fruitsIngredientCount += 1  fruitsList(refrigerator)  elif x == 3:  global meatFishIngredientCount  refrigerator[x].append([newIngredient, 0])  meatFishIngredientCount += 1  meatAndFishList(refrigerator)  else:  tryAgain()  #removes a unique ingredient from the pantry or refrigerator  def removeUnique(panOrRef, categoriesList):  for x in categoriesList:  x -= 1  global pantry  global refrigerator  if panOrRef == 1:  newIngredient = int(input('What new ingredient do you want to remove from Pantry:{}?:\n'.format(pantryCategories[x])))  newIngredient -= 1  if x == 0:  global canOrJarIngredientCount  del refrigerator[x][newIngredient]  canOrJarIngredientCount -= 1  canOrJarredIngredientsList(pantry)  elif x == 1:  global grainsIngredientCount  del refrigerator[x][newIngredient]  grainsIngredientCount -= 1  grainsIngredientsList(pantry)  elif x == 0:  global oilsAndvinegarIngredientCount  del refrigerator[x][newIngredient]  oilsAndvinegarIngredientCount -= 1  oilsAndVinegarList(pantry)  elif x == 0:  global spicesHerbsSeasoningIngredientCount  del refrigerator[x][newIngredient]  spicesHerbsSeasoningIngredientCount -= 1  spicesList(pantry)  elif x == 0:  global pantryVegetablesIngredientCount  del refrigerator[x][newIngredient]  pantryVegetablesIngredientCount -= 1  pantryVegetablesList(pantry)  elif x == 0:  global bakingIngredientCount  del refrigerator[x][newIngredient]  bakingIngredientCount -= 1  bakingList(pantry)  elif x == 0:  global sweetenersIngredientCount  del refrigerator[x][newIngredient]  sweetenersIngredientCount -= 1  sweetenerList(pantry)  elif x == 0:  global snacksCerealIngredientCount  del refrigerator[x][newIngredient]  snacksCerealIngredientCount -= 1  snacksCerealList(pantry)  elif x == 0:  global condimentsIngredientCount  del refrigerator[x][newIngredient]  condimentsIngredientCount -= 1  condimentList(pantry)  elif panOrRef == 2:  newIngredient = int(input('What new ingredient do you want to remove from Refrigerator:{}?:\n'.format(refrigeratorCategories[x])))  newIngredient -= 1  if x == 0:  global dairyEggsIngredientCount  del refrigerator[x][newIngredient]  dairyEggsIngredientCount -= 1  dairyAndEggsList(refrigerator)  elif x == 1:  global refrigeratorVegetablesIngredientCount  del refrigerator[x][newIngredient]  refrigeratorVegetablesIngredientCount -= 1  refrigeratorVegetablesList(refrigerator)  elif x == 2:  global fruitsIngredientCount  del refrigerator[x][newIngredient]  fruitsIngredientCount -= 1  fruitsList(refrigerator)  elif x == 3:  global meatFishIngredientCount  del refrigerator[x][newIngredient]  meatFishIngredientCount -= 1  meatAndFishList(refrigerator)  #function to ask user 3 options: add quantity, remove quantity, or view what they have in their fridge.  def AddRemoveView():  count = 0  while count == 0:  print('What would you like to do?\n')  print('1. Add Quantity\n')  print('2. Remove Quantity\n')  print('3. View Ingredients\n')  print('4. Add Unique Ingredient\n')  print('5. Remove An Ingredient\n')  answer = input()  if answer.isnumeric():  answer = int(answer)  if answer < 1 and answer > 3:  tryAgain()  elif answer == 1 or answer == 2 or answer == 4 or answer == 5:  choosePanOrRef(answer)  elif answer == 3:  print('Would you like to:')  print('1. View all ingredients.')  print('2. View from pantry or refrigerator.')  viewAnswer = input()  if viewAnswer == '1':  canOrJarredIngredientsList(pantry)  grainsIngredientsList(pantry)  oilsAndVinegarList(pantry)  spicesList(pantry)  pantryVegetablesList(pantry)  bakingList(pantry)  sweetenerList(pantry)  snacksCerealList(pantry)  condimentList(pantry)  dairyAndEggsList(refrigerator)  refrigeratorVegetablesList(refrigerator)  fruitsList(refrigerator)  meatAndFishList(refrigerator)  elif viewAnswer == '2':  viewCount = 0  while viewCount == 0:  panOrRef = input(pantryOrRefrigerator())  if panOrRef.isnumeric():  panOrRef = int(panOrRef)  if panOrRef == 1:  viewList = viewCategories(panOrRef)  viewQuantity(panOrRef, viewList)  viewCount += 1  elif panOrRef == 2:  viewList = viewCategories(panOrRef)  viewQuantity(panOrRef, viewList)  viewCount += 1  else:  tryAgain()  doneCount = 0  while doneCount == 0:  done = input("Press 'Enter' to go back to main menu.")  if done == '':  doneCount += 1  else:  continue  #starting the program on infinite loop  infiniteloop = 0  while infiniteloop < 1:  AddRemoveView() |

PART 6: Your Completed Program

|  |
| --- |
| **Task**  Provide the Replit link to your full program code.  **Requirements**   * The program must work correctly with all the comments included in the program.   **Inspiration**   * Check before submitting your touchstone that your final version of the program is running successfully. |
| <Provide the link to your program here> |