Note:

1. **Use try-except blocks for all programs to indicate errors in input format.**

2. Use built-in attributes/ methods/ functions wherever necessary.

3. All programs should be menu driven and looped. Include exit option for each to stop execution based on user’s choice.

1. Input the following: expenditure for any 2 months, in the form of a nested dictionary, **under \_\_main\_\_** segment of your program. The outer dictionary uses name of the month as key and an inner dictionary as values. The inner dictionary contains ‘Monthly’ and ‘Variable’ as the keys. The corresponding values are given below. Store them in a list. Store 0 to indicate no value for each.

Monthly:

* [Mortgage and/or rent](https://www.incharge.org/financial-literacy/budgeting-saving/how-to-reduce-mortgage-rent-payments/)
* [Auto expenses: car payments, insurance, gas and tolls](https://www.incharge.org/financial-literacy/budgeting-saving/save-money-transportation-car-expenses-gas-insurance/)

Variable:

* Food: groceries and eating out
* Prescriptions

**Create functions :**

1. **def TOTAL\_EXP( sal ):** to accept salary **(from main)** ,compute total expenditure per month . Store this in a **global dictionary exp. Compute total expenditure of all the months** and return “surplus” - If salary-expenditure> 0 or “deficit” - If salary-expenditure< 0.

Use **positional argument** during function call. Store the result in **main as res**.

1. **def SUGGEST(result):** Using the values returned in the task above (**res from main**), display suggestions as given below.
2. If res is surplus - display: invest “remaining amount”
3. If res is deficit): Compute the highest among values in ‘Variable’. Display “Change plan or stop spending …<max amount>”

If the deficit exists for all months, then display “work overtime or find a second job”.

**Use keyword argument during function call.**

**Sample input:**

{‘Jan’:{‘Monthly’:[15000, 2000] ,‘Variable’:[ 2000, 0],’tot’:19000 } ,

‘Feb’: {‘Monthly’:[ 15000, 800 ], ‘Variable’: [1500, 100 ]} }

1. Create a file/module Q2 and input details of ‘n’ countries in the form of a dictionary - **dct\_cntry**. The values include a list of: capital, population, President/Prime Minister/Head of the State. Perform the following tasks, from the file Q2, **using a module for each.** Display all results, in a tabular form with headers.
2. Module name: **FORMATSTR**

Create the function: **def change( )** - Convert all strings in **dct\_cntry** and store them with first letter in upper case. Return the dictionary.

1. Module name: **CHECK\_PM**

Create the function: **def remove\_cntry( )** - Move details of countries which do not have a Prime Minister/Head of the State (empty string). Store these in a separate dictionary. Return the dictionary.

1. Module name: **EDIT\_CNTRY**

Create the function : **def add( )** - Prompt the user to add more data to the list of a chosen country. Choices include: names of neighbouring countries, type of government, places to visit, food delicacies. Return the edited dictionary.

1. Write a menu based program to perform the tasks given below.
   * 1. Create a function: **def Med\_Cat( med\_name )** - Input names of ‘n’ medicines. Store them in a dictionary according to the categories: oral\_drugs, injectable, vaccines, antiseptics.

Using the string med\_name (sent from main), search for the category and print it.

* + 1. Create a function : **def med\_count(name\_list)** - Input a list of medicine names (lst\_med) and store them in a list. Remove elements from name\_list(list of medicine names sent from main), which are not present in the list: lst\_med. Display the original list and the modified list in main.
    2. Create a function: **def chk\_exp(usr\_date)** - Input a dictionary, containing medicine name(key) , date of manufacture and expiry(values, in a list). usr\_date is a tuple containing expiry date in the format : DD/MM/YYYY, and isinput in main. Using usr\_date, create a tuple of all medicines having this value. Return the tuple.

1. Write a program to connect to any two programming based websites. Print the website that contains the maximum number of programming language names. In addition, print the first 4 lines of meta data of each website.