

# ALA\_Sessional\_1\_Pattern\_Students

September 25, 2023

## 1 ALA Lab Sessional 1 Pattern Problem

### 1.0.1 1. Pre-requisites for lab in VS Code

1. Install PDF export support <https://saturncloud.io/blog/how-to-export-jupyter-notebook-by-vscode-in-pdf-format/>

### 1.0.2 2. At the end, export your notebook as pdf

1. Open VS Code command palette Shift + Ctrl + P
2. Type “Export Jupyter Notebook” in the search bar and select “Export Jupyter Notebook to PDF”
3. Save the pdf and send it to me in MS Teams

### 1.0.3 3. Lab Sessional - Dos and Don'ts - Summary

1. A code template will be given to you in actual lab sessional just like this pattern problem
2. The template will follow a linear sequence of TODOs appropriately labelled with question marks
3. You will have to fill the TODO question marks to compile those notebook cells and proceed to next cells
4. You can think of the linear sequence of TODO templates as a guided thought process.
5. Sessional is open book. Google, github, browse product documentation ChatGPT - Do anything you want, except copying from your classmates (Sending questions to your seniors and seeking answers is prohibited). If you are caught carrying out these illegal activities, you will be reported to MSIS Director for immediate action.
6. You CANNOT replace the TODO code template with some other code copied from stack overflow, ChatGPT etc. All your browsing and search should give you insights into finally how you can fit that into the framework I provide for the thought process of solving the problem in this notebook. You will have to mandatorily fill the question marks and proceed with the lab problem.

```
[ ]: import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
```

### 1.0.4 Pattern Problem

The food that we eat contains primarily three items - fats, proteins, and carbohydrates. The calorific data for the three is as tabulated below

Nutrients	Cal/gm
Fats	9
Proteins	4
Carbohydrates	4

It was found that a typical person from Karnataka consumes the following on a daily basis 1. two servings of Idli for breakfast 2. one serving of rice in the lunch 3. one serving of bisibelebath for supper(night) 4. two servings of snacks during the day

The fats, proteins and carbs breakdown in *1 serving* each of the above food items is tabulated below

	Fat (gm)	Protein (gm)	Carbohydrate (gm)
Idli	0.8	30	25
Rice	0.9	5	50
Snacks	20	5	10
Bisibelebath	10	20	50

### 1.0.5 Your task

1. Use numpy to create vectors (and matrices if needed) to hold the data
2. Use simple linear algebra operations (such as linear combinations, dot product, hadamard product, matrix-vector multiplication etc.) on these vectors and matrices to calculate the total calories consumed by the typical person from Karnataka over the span of day and night.

```
[ ]: food_nutrients_in_grams = np.array([
    [0.8, 30, 25],
    [0.9, 5, 50],
    [20, 5, 10],
    [10, 20, 50]
])
food_nutrients_in_grams
```

```
[ ]: array([[ 0.8, 30. , 25. ],
           [ 0.9,  5. , 50. ],
           [20. ,  5. , 10. ],
           [10. , 20. , 50. ]])
```

```
[ ]: calories_per_grams = np.array([9, 4, 4])
calories_per_grams
```

```
[ ]: array([9, 4, 4])
```

```
[ ]: # TODO: 1
# Fill the array to hold the servings per day
# Caution: Order of food items and their corresponding servings should be lined
# up correctly
servings_per_day = np.array([[ 2., 1., 1., 2.]])
```

```
[ ]: # TODO: 2
# using the food_nutrients_in_grams matrix and cal_per_grams vector,
# calculate the calories in each food items
# For e.g. Idli has Fat-Protein-Carbohydrate = [0.8, 30, 25]
# cal_per_grams of Fat-Protein-Carbohydrate = [9, 4, 4]
# idli_calories = idli_fat * fat_calories + idli_protein * protein_calories +
# idli_carb * carb_calories
idli_calories = np.sum(food_nutrients_in_grams[0]* calories_per_grams)
print(idli_calories)
```

227.2

```
[ ]: # TODO: 3

# Similarly calculate rice calories
rice_calories = np.sum(food_nutrients_in_grams[1]* calories_per_grams)
print(rice_calories)

# Calculate snack calories
snack_calories = np.sum(food_nutrients_in_grams[2]* calories_per_grams)
print(snack_calories)

# Calculate bisibelebath calories
bisibelebath_calories = np.sum(food_nutrients_in_grams[3]* calories_per_grams)
print(bisibelebath_calories)
```

228.1

240.0

370.0

```
[ ]: # TODO: 4
# In the previous cells you calculated calories in individual food items
# as individual numpy operation
# All the 4 numpy operations can be combined into a single numpy operation
# between a matrix and vector
fooditem_calories = np.matmul(food_nutrients_in_grams,calories_per_grams )
print(fooditem_calories)
```

[227.2 228.1 240. 370. ]

```
[ ]: # TODO: 5
```

```

# In previous cell you calculated the calories in each food item i.e. Idli,
  ↳Rice etc. as a vector
# Now you should calculate the daily calories of idli, rice, etc.
# For e.g. daily_idli_calories = idli_calories * num_idli_servings_per_day
# daily_rice_calories = rice_calories * num_rice_servings_per_day
# But you should calculate it with a operation on fooditem_calories vector and
  ↳another _____ vector

# You might get a compilation error. Follow instructions in next cells to fix
  ↳the error
total_daily_calories = np.multiply(fooditem_calories,servings_per_day )
print(total_daily_calories)

```

```
[[454.4 228.1 240. 740. ]]
```

```

[ ]: # TODO: 6
# Write your debug code here to print the shape of the two vectors you used in
  ↳above cell
print(f"shape of fooditem_calories vector is {fooditem_calories.shape}")
print(f"shape of servings_per_day vector is {servings_per_day.shape}") #This
  ↳shape will be odd. hence error

```

```

shape of fooditem_calories vector is (4,)
shape of servings_per_day vector is (1, 4)

```

```

[ ]: # TODO: 7
# You might want to flatten the second vector before using it
servings_per_day = servings_per_day.flatten()

```

```

[ ]: print(f"shape of fooditem_calories vector is {fooditem_calories.shape}")
print(f"shape of servings_per_day vector is {servings_per_day.shape}")

```

```

shape of fooditem_calories vector is (4,)
shape of servings_per_day vector is (4,)

```

```

[ ]: # TODO: 8
# Now you should be able to use the flattened vector here
# Calculate total daily calories with a operation on two vectors whose output
  ↳is scalar
# The two vectors are fooditem_calories vector and another _____ vector
total_daily_calories = np.multiply(fooditem_calories, servings_per_day)
print(total_daily_calories)

```

```
[[454.4 228.1 240. 740. ]]
```

```

[ ]: # TODO: 9
# Line below is not printing the total daily calories
# Fix this
print(f"Total daily calories = {total_daily_calories}")

```

Total daily calories = [454.4 228.1 240. 740. ]