

Assignment 3 – Python

Software System Development – Monsoon 2023

Due Date: 04th Nov, 2023, 11:59 PM

Instructions:

- This assignment is an individual submission.
- Total Marks of 100 Marks with duration of 3 weeks.
- All script submissions should be submitted via moodle
- Inputs/output should fit the criteria mentioned in the respective question.
- All other conditions are open to your interpretations.
- Evaluation will be conducted based on a fixed grading rubric (syntax, logic, input and output) and the marks are divided as per prescribed weightage in respective questions.
- The assignment must be made in Python only.
- You are free to use any libraries that you deem fit, except when mentioned in the particular question.
- For queries, reach out to TAs via Moodle

Submission Criteria:

- Please create one ZIP file as <rollnumber>_A3.ZIP
- Include all your scripts inside the ZIP as follows <rollnumber>_A3_Q1.py, <rollnumber>_A3_Q2.py, <rollnumber>_A3_Q3.py and README.md (to provide any instructions to evaluator or your notes)
- For Example:

2022201079_A3.zip

```
|__2022201079_A3
    |__2022201079_A3_Q1.py
    |__2022201079_A3_Q2.py
    |__2022201079_A3_Q3.py
    |__README.md
```

Question 1: (30 Marks)

Expense Tracker

You have to design a Menu based Expense Tracker. The menu will be of the following format:

Menu:

- Add participant(s)
- Add expense
- Show all participants
- Show expenses
- Exit/Export.

Explanation

- **Add participant(s):** Enter 1 or more than 1 unique names who will participate in the transactions.
- **Add expense:** On entering this option you, have to take 3 further inputs:
 - **Paid by:** <Name of the person who paid the money>
 - **Amount:** <The amount paid by the person>
 - **Distributed amongst:** <All the participants that are involved in this payment.>
 - You can take the number of participants first and then take the name of each person or you can take one string and tokenize it. It's your choice.
 - The split is always equal among the mentioned names.
- **Show Participants:** Just display all participants that are currently there.
- **Show expenses:** A table should be displayed showing each person's expenses. The table must be divided into 2 columns for each participant.

Participant's Name	
Amount	"Owes" or "Gets Back"

- Note that the table must contain each participant. The above table is a subset of the table showing only one participant.
- **Exit/Export:** On exiting, you have to write back the data in a .csv file. The name of the file should be "expenses.csv". The data arrangement in the csv file is up to you.

Note: Write the program covering all the logical edge cases that you can identify. Rest all details are up to you and should be mentioned in the Readme.md.

Question 2: (40 Marks)

You have been made the analyst for the IPL team “Gachibowli Gorillas”. You have to build a “Cricketer’s Directory” in python. The directory consists of a list of entries. Each entry can store details such as:

1. First Name
2. Last Name
3. Age
4. Nationality
5. Role (Batsmen, Bowler, All-rounder, Wk-Batsmen)
6. Runs
7. Balls
8. Wickets
9. Strike Rate

For Calculating Strike Rate:

If the player is a batsmen, then Batting Strike rate = Runs/Balls

If the player is a bowler, then Bowling Strike rate = Wickets/Balls

If the player is an All-rounder, then Strike rate = $\max(\text{Batting Strike rate, Bowling Strike rate})$

In order to maintain the cricket directory, make a menu driven program and implement the following functionalities:

- Consider reading/loading entries from .csv file
- Display the directory on the terminal (in a table-like format)
- A user can add a new entry (Strike rate may or may not be given)
- Allow removing and updating entries in the directory
- Search for entries in the directory based on some attribute(s)
- Write back the data in a CSV file

Write the program covering all edge cases that you can identify. Rest all details are up to you and should be mentioned in the Readme.

Question 3: (30 Marks)

Visualize Data

1.

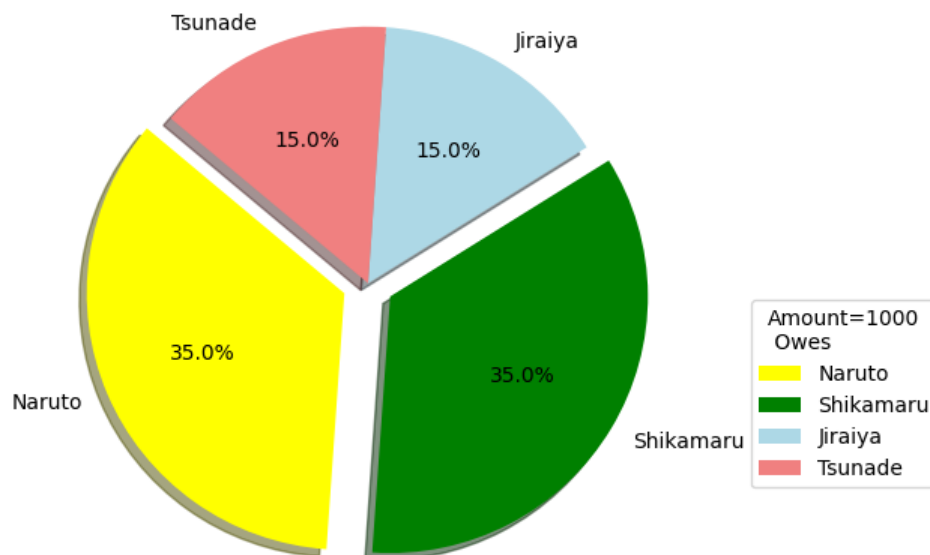
For this question it is mandatory that you use the library matplotlib and no other data visualization library. This is an extension of the first 2 questions. The expenses.csv you generated in the first question will be used as the base for creating pie charts here.

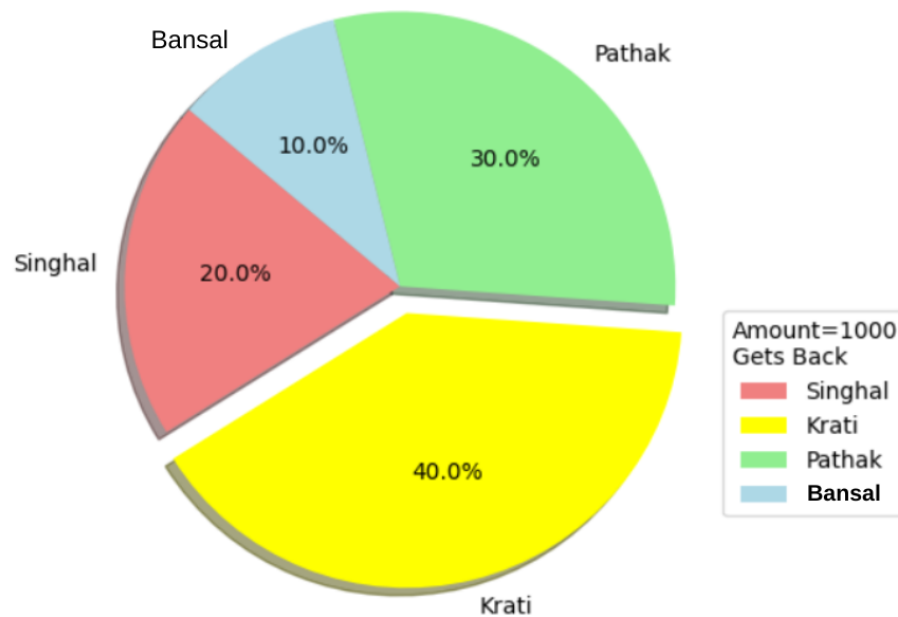
A. Pie Chart 1

- The name of this Pie Chart is "Owes". You have to create a pie chart showing the % of money owed by each person.
- The persons who owe the money (those who have to give the money) should be the only ones to be involved in the pie chart.

B. Pie Chart 2

- The name of this Pie Chart is "Gets Back". You have to create a pie chart showing the % of money each person will get back.
- The persons who get back the money (those who are owed the money) should be the only ones to be involved in the pie chart.



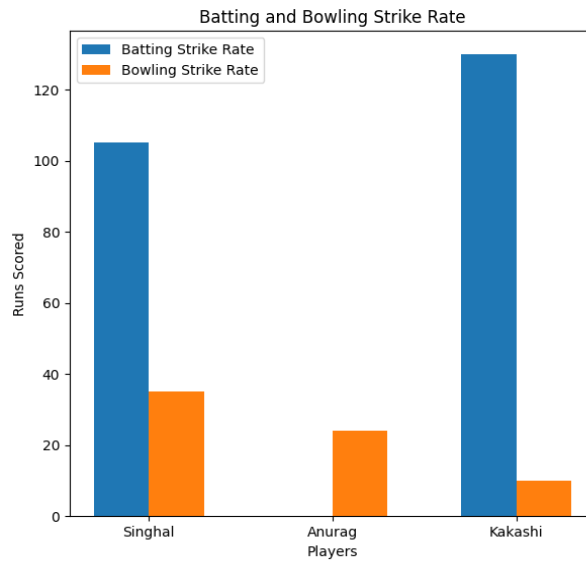


General instructions for the pie chart:

- The legend must show the name and amount of the respective pie chart.
- The pie which has the highest % in the pie chart must "explode". (See the Naruto's and Shikamaru's pie in the "Owes" chart and Krati's pie in "Gets Back" chart)

2.

This part is an extension of the data created in Q-2. You have to create a comparison histogram for all the players. For example:



For this particular example, there were only 3 players, where Singhal and Kakashi are all rounders, but Anurag is a bowler only.

Note:

- You have to create histograms based on the player's role.
- Just show the relevant strike rate and no other data.