**(a)Create a dictionary of 5 students with their names as keys and marks as values. Calculate the average marks, Highest scorer, and print the name of students who scored above average.**

In [ ]:

```
#dictionary of 5 students with their names as keys and marks as values
studentDict= {
    "Prateek_Badola":8.56,
    "Raj_Sharma": 7.89,
    "Rishi_Kumar" : 9.64,
    "Satyam_Bhardwaj" : 8.80,
    "Mayank_Shaw" : 8.64
}
```

In [ ]:

```
#average marks
avg=0.0
for k in studentDict.keys():
    avg=avg+studentDict[k]
avg=avg/len(studentDict)
print("Average Marks = ",avg)
```

```
Average Marks =  8.706
```

In [ ]:

```
#Highest scorer
high=0.0
high_key=""
for k in studentDict.keys():
    if len(high_key)==0 or studentDict[k]>high:
        high=studentDict[k]
        high_key=k
print("Highest Scorer = ",high_key)
print("Marks = ",high)
```

```
Highest Scorer =  Rishi_Kumar
Marks =  9.64
```

In [ ]:

```
#name of students who scored above average.
print("name of students who scored above average.")
for k in studentDict.keys():
    if studentDict[k]>avg:
        print(k)
```

```
name of students who scored above average.
Rishi_Kumar
Satyam_Bhardwaj
```

In [ ]:

In [ ]:

In [ ]:

**(b)You have a list of temperatures for 7 days: temps = [30.5, 32.0, 29.8, 31.2, 30.1, 28.9, 33.0] Write a function that returns: 1. The maximum and minimum temperature, 2. The average temperature, 3. A list of temperatures above the average.**

In [ ]:

```
#function that returns The maximum and minimum temperature
def MaxMinTemp(list_temp):
    min_temp=list_temp[0]
    max_temp=list_temp[0]
    for i in range(1,len(list_temp)):
        if list_temp[i]>max_temp:
            max_temp=list_temp[i]
        elif list_temp[i]<min_temp:
            min_temp=list_temp[i]
    return max_temp,min_temp
```

In [ ]:

```
#function that return The average temperature
def AvgTemp(list_temp):
    avg_temp=0.0
    for i in range(len(list_temp)):
        avg_temp=avg_temp+list_temp[i]
    avg_temp=avg_temp/len(list_temp)
    return avg_temp
```

In [ ]:

```
# function that returns A list of temperatures above the average.
def ListAvgTemp(list_temp):
    avg_temp=AvgTemp(list_temp)
    res=[]
    for i in range(len(list_temp)):
        if list_temp[i]>avg_temp:
            res.append(list_temp[i])
    return res
```

In [ ]:

```
temps = [30.5, 32.0, 29.8, 31.2, 30.1, 28.9, 33.0]
```

In [ ]:

```
maxt,mint=MaxMinTemp(temps)
print("Maximum Tempraature = ",maxt)
print("Minimum Temprature = ",mint)
```

```
Maximum Tempraature =  33.0
Minimum Temprature =  28.9
```

In [ ]:

```
avgt=AvgTemp(temps)
print("Average Tempraature = ",avgt)
```

```
Average Tempraature =  30.785714285714285
```

In [ ]:

In [ ]:

```
list_avg=ListAvgTemp(temps)
print("list of temperatures above the average. : ",list_avg)
```

```
list of temperatures above the average. :  [32.0, 31.2, 33.0]
```

---

**(c)Given: student = {"name": "Alice", "score": 92, "subject": "Math"}. Print a formatted message: "Alice scored 92 in Math."**

---

In [ ]:

```
#Given
student = {"name": "Alice", "score": 92, "subject": "Math"}
#Print a formatted message: "Alice scored 92 in Math."
print(" {} scored {} in {}.".format(student["name"],student["score"],student["
```

```
 Alice scored 92 in Math.
```

---

**(d)Create a CSV file data.csv with the following content: name,score Alice,91 Bob,85 Charlie,78 Note: You have to add at least 20 random values in this csv file. Write code to read the file, print each name and score, count how many students scored above 80.**

---

In [ ]:

```
import csv
```

In [ ]:

```
#data
cols=["name","score"]
rows = [
    ["Alice", 91],
    ["Bob", 85],
    ["Charlie", 78],
    ["Aarav", 88],
    ["Priya", 92],
    ["Rahul", 81],
    ["Ananya", 95],
    ["Vikram", 76],
    ["Sneha", 89],
    ["Rohit", 84],
    ["Kavya", 90],
    ["Amit", 73],
    ["Neha", 87],
    ["Suresh", 69],
    ["Pooja", 93],
    ["Manish", 80],
    ["Riya", 86],
    ["Arjun", 82],
    ["Meera", 94],
    ["Kunal", 77]
]
```

In [ ]:

```
#writing to csv file
filename="./data.csv"
with open(filename,'w') as csvfile :
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(cols)
    csvwriter.writerows(rows)
```

In [ ]:

```
#reading from csv file and printing
rcol=[]
rrow=[]
count=0
with open(filename,'r') as csvfile :
    csvreader = csv.reader(csvfile)
    rcol=next(csvreader)
    for col in rcol:
        print(col,end="\t")
    print("\n")
    for row in csvreader:
        i=0
        for col in row:
            i=i+1
            print(col,end="\t")
            if i%2==0:
                if int(col) >80:
                    count=count+1
        print("\n")
```

```
name        score

Alice        91

Bob        85

Charlie        78

Aarav        88

Priya        92

Rahul        81

Ananya        95

Vikram        76

Sneha        89

Rohit        84

Kavya        90

Amit        73

Neha        87

Suresh        69

Pooja        93

Manish        80

Riya        86

Arjun        82

Meera        94

Kunal        77
```

```
In [ ]: print("Number of students scored above 80 : ",count)
```

```
Number of students scored above 80 :  14
```

---

**(e)Write a function grade_score that returns • 'A' for 85 and above • 'B' for 70–84 • 'C' for 50–69 • 'F' below 50**

---

```
In [ ]:
```

In [ ]:

```python

```

In [ ]:

```python
#function grade_score that returns • 'A' for 85 and above • 'B' for 70–84 • 'C
def grade_score(marks):
    if marks>=85 :
        return "A"
    elif marks >=70 and marks <=84:
        return "B"
    elif marks>=50 and marks<=69:
        return "C"
    else:
        return "F"
```

In [ ]:

```python
print("marks : 99 , grade : ",grade_score(99))
```

```
marks : 99 , grade :  A
```

---

**(f)You are given a list of students, where each student record contains: students = [ ("Bob", [88, 82, 80, 80, 85]), ("Charlie", [90, 92, 88, 94, 91]), ("Alice", [80, 85, 90, 75, 85]), ("David", [70, 75, 72, 68, 70]), ("Eve", [88, 82, 80, 80, 85]) ] Use a lambda function to: Calculate the average marks for each student. Sort students: First by highest average marks (descending) If averages are equal, sort by name alphabetically**

---

In [ ]:

```python
#student records
students = [
("Bob", [88, 82, 80, 80, 85]),
("Charlie", [90, 92, 88, 94, 91]),
("Alice", [80, 85, 90, 75, 85]),
("David", [70, 75, 72, 68, 70]),
("Eve", [88, 82, 80, 80, 85])
]
```

In [ ]:

```python
#lambda function to: Calculate the average marks for each student
acg_students=dict(map(lambda x : (x[0] , float(sum(x[1]))/len(x[1])),students)
print("Average Marks For each student : ",acg_students)
```

```
Average Marks For each student :  {'Bob': 83.0, 'Charlie': 91.0, 'Alice': 83.0,
'David': 71.0, 'Eve': 83.0}
```

In [ ]:

```python
#sorting students First by highest average marks (descending)
#If averages are equal, sort by name alphabetically
sortedStudents = sorted(acg_students,key= lambda x:(-(acg_students[x]),x))
print("Students in Sorted order : ",sortedStudents)
```

```
Students in Sorted order :  ['Charlie', 'Alice', 'Bob', 'Eve', 'David']
```

---

**(g)Create a tuple of employees:employees = (**

**("E001", "Alice", 45000),**

**("E002", "Bob", 52000),**

**("E003", "Charlie", 38000),**

**("E004", "David", 60000)**

)

**•Use map() to apply: 10% bonus if salary ≥ 50,000, 5% bonus otherwise**

**• Store updated salaries as a new tuple.**

**• Convert salaries into a NumPy array.**

**• Find: Mean salary, Standard deviation**

---

In [ ]:

```
#tuple of employees
employees = (

("E001", "Alice", 45000),

("E002", "Bob", 52000),

("E003", "Charlie", 38000),

("E004", "David", 60000)

)
```

In [ ]:

```
#Use map() to apply: 10% bonus if salary ≥ 50,000, 5% bonus otherwise
updated_employees = tuple(map(lambda x : (x[0],x[1],
                                          int(1.1*x[2]) if x[2]>= 50000 else i
#Store updated salaries as a new tuple.
print("Updated Salaries : ")
for t in updated_employees:
    print(t)
```

```
Updated Salaries :
('E001', 'Alice', 47250)
('E002', 'Bob', 57200)
('E003', 'Charlie', 39900)
('E004', 'David', 66000)
```

In [ ]:

```
import numpy as np
#Convert salaries into a NumPy array.
salaries=np.array(list(map(lambda x : x[2],updated_employees)))
print("Salaries numpy array = ",salaries)
```

```python
#Find: Mean salary, Standard deviation
mean=salaries.mean()
stadev=salaries.std()
print("Mean : ",mean)
print("Standard deviation : ",stadev)
```

```
Salaries numpy array =  [47250 57200 39900 66000]
Mean :  52587.5
Standard deviation :  9882.204650279207
```

---

---

## Learning Outcome

---

Learned to use Python data structures (dictionary, list, tuple) to store and analyze student and employee records.

Implemented functions, lambda expressions, and conditional logic to compute averages, grades, sorting, and filtering.

Gained practical experience in file handling (CSV) and NumPy operations for statistical calculations like mean and standard deviation.

---