Abhinav Srivatsa

21BDS0340

Computer Science Lab

Problem Set 4

**Question 1**

**Code:**

import numpy as np

marks = []

commerce = []

for i in range(5):

    inp = eval(input().strip().split(' ')[-1])

    commerce.append(inp[-1])

    for i in inp:

        marks.append(i)

marks = np.array(marks)

commerce = np.array(commerce)

fail = marks[marks < 50]

dist = marks[marks > 90]

commerce = commerce[commerce > 50]

print(fail)

print(dist)

print(commerce)

**Algorithm:**

Import numpy as NP

Initialise Marks as list

Initialise Commerce as list

Loop though from 0 till 4 as I

Read Inp as input evaluated, stripped, and split by space and taking the last index

Append the last index of Inp to Commerce

Loop through Inp as J

Append J to Marks

Reinitialise Marks as NP array of Marks

Reinitialise Commerce as NP array of Commerce

Calculate Fail as Marks < 50

Calculate Dist as Marks > 90

Calculate Commerce as Commerce > 50

Display Fail

Display Dist

Display Commerce

**Output:**

Text

Description automatically generated

**Question 2**

**Code:**

import pandas as pd

def score\_mapper(score):

    if score >= 90:

        return 'A'

    if score >= 80:

        return 'B'

    if score >= 60:

        return 'C'

    else:

        return 'F'

inp1 = eval(input())

inp2 = eval(input())

scores = {

    'Mark-1': inp1,

    'Mark-2': inp2

}

indices = ['Maths', 'Eng', 'Cs', 'Phy', 'Che', 'Bio']

scores\_df = pd.DataFrame(scores, index = indices)

grade\_df = scores\_df.applymap(score\_mapper)

print(grade\_df)

**Algorithm:**

Import pandas as PD

Score\_Mapper(Integer Score):

If Score >= 90, then return ‘A’

If Score >= 80, then return ‘B’

If Score >= 60, then return ‘C’

Else return ‘F’

Read Inp1 as evaluated input

Read Inp2 as evaluated input

Initialise dictionary Scores with keys ‘Mark-1’, ‘Mark-2’ and values as Inp1, Inp2 respectively

Initialise Indices as ['Maths', 'Eng', 'Cs', 'Phy', 'Che', 'Bio']

Initialise Scores\_DF as PD dataframe of Scores with index as Indices

Calculate Grade\_DF as Scores\_DF applying Score\_Mapper

Display Grade\_DF

**Output:**

A picture containing text

Description automatically generated