

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:

```
student1 mark is [100,10,98,20,45]
student2 mark is [90,10,96,15,93]
student3 mark is [88,10,86,84,83]
student4 mark is [10,10,20,10,40]
student5 mark is [30,40,50,90,80]
[10 20 45 10 15 10 10 10 20 10 40 30 40]
[100 98 96 93]
[93 83 80]
```

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:

[90, 80, 70, 50, 20, 10]

[94,7,25,13,84,52]

	Mark-1	Mark-2
Maths	A	A
Eng	B	F
Cs	C	F
Phy	F	F
Che	F	B
Bio	F	F