MIT-WPU

Dr. Vishwanath Karad
MIT WORLD PEACE
UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

## Third Year B. Tech (EL & CE)

**Semester: VI** **Subject:** Data Science for Engineering

**Name: Shreerang Mhatre** **Class: TY**

**Roll No: 52** **Batch: A2**

## Experiment No: 01

**Name of the Experiment: Data Science Fundamentals**

**Performed on: 15/1/2024**

**Submitted on: 15/1/2024**

**Problem Statement:**

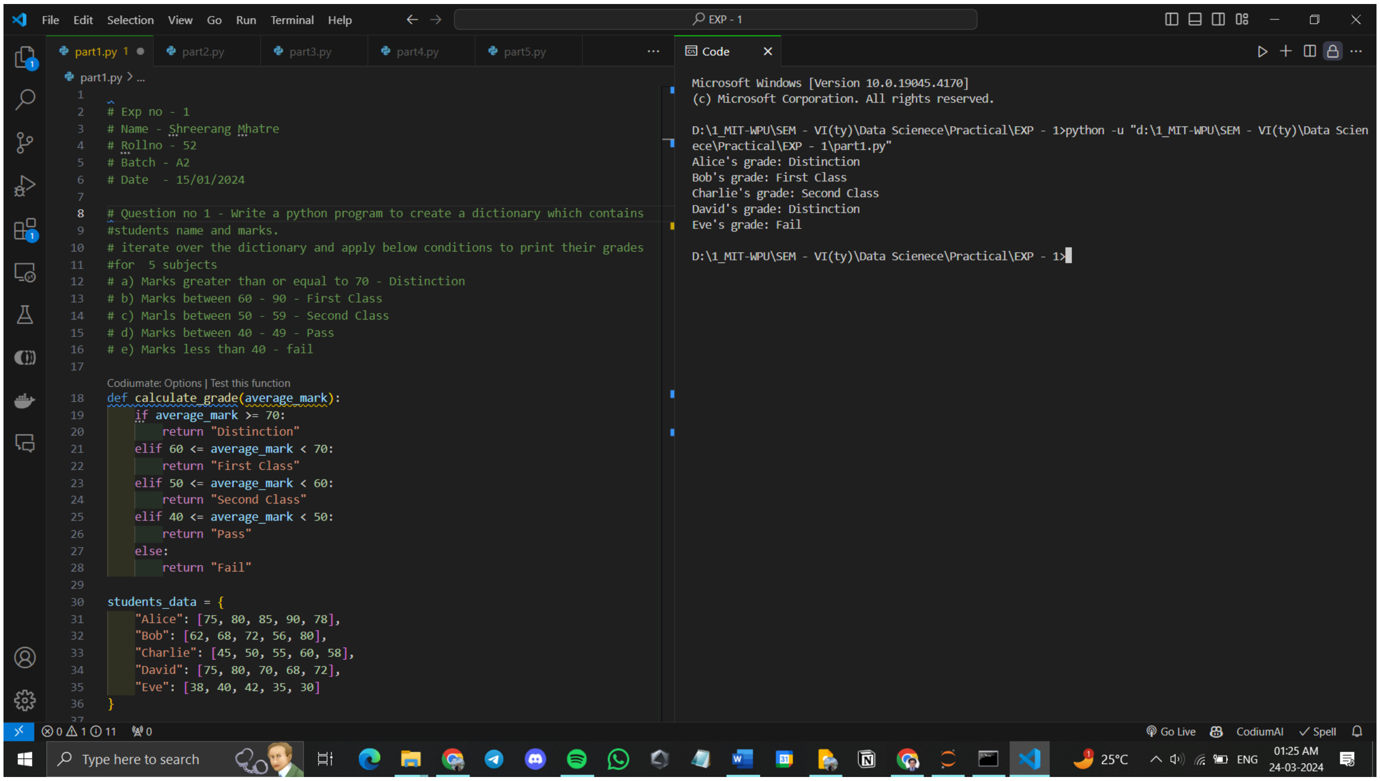
**Question no 1 -**
Write a python program to create a dictionary which contains students name and marks. iterate over the dictionary and apply below conditions to print their grades for 5 subjects
a) Marks greater than or equal to 70 - Distinction
b) Marks between 60 - 90 - First Class
c) Marls between 50 - 59 - Second Class
d) Marks between 40 - 49 - Pass
e) Marks less than 40 - fail

**Question no 2 -**
Write a python program to create a ID array of numbers from 0 to 9

**Question no 3 -**
Write a Numpy Python Program to create an array of all even integers from 30 to 70

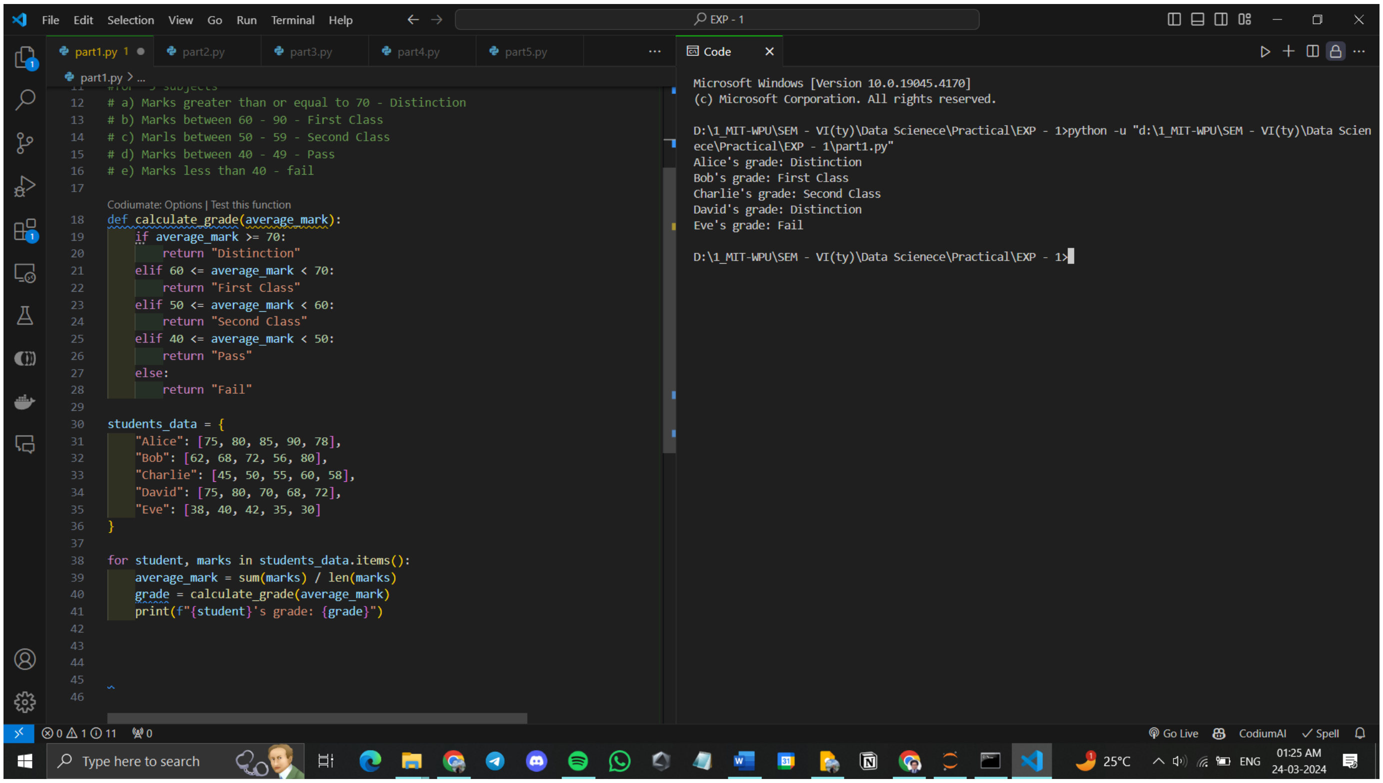
**Question no 4 -**
Write a Numpy program to create a 3x4 matrix filled with values from 10 to 21

**Question no 5 -**
Write a Numpy program to compute the sum of all elements, sum of each column and sum of each row of a given array.



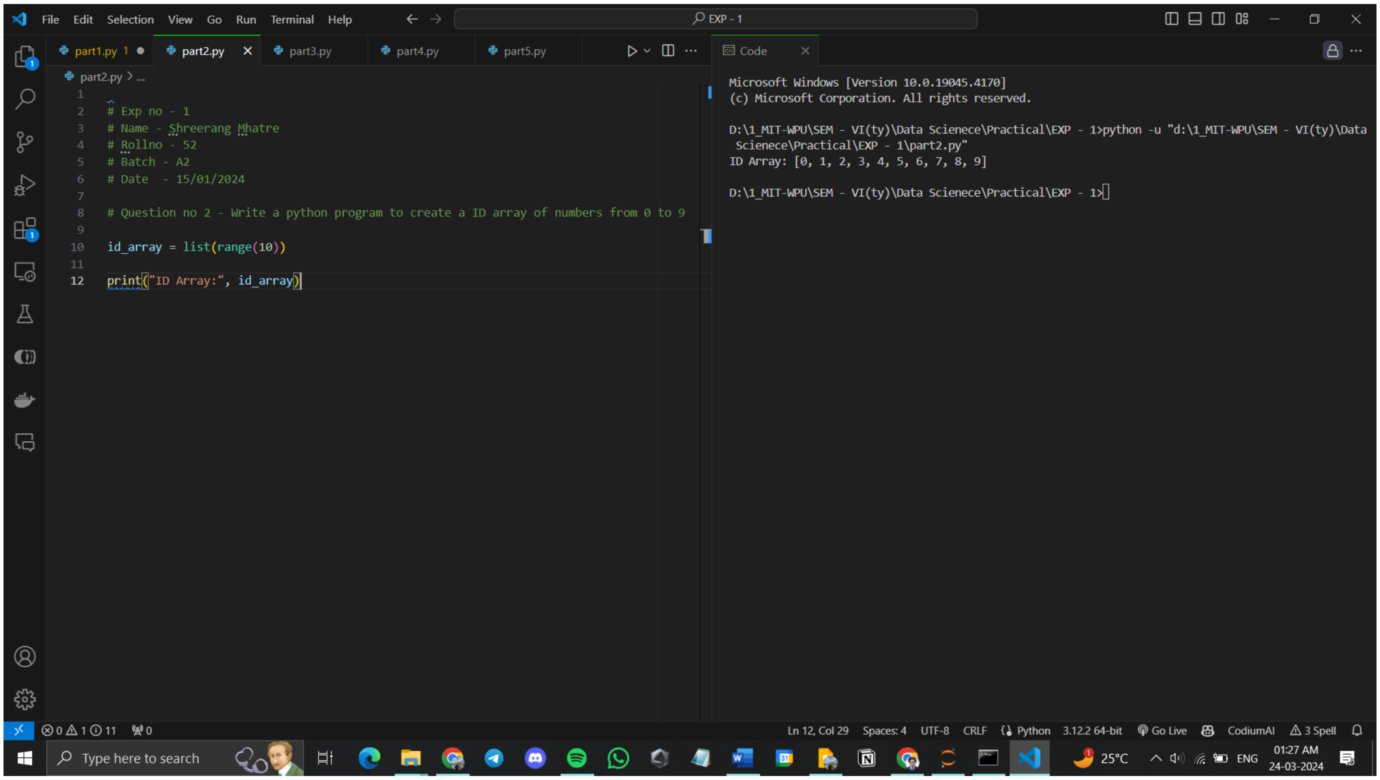
```python

# Exp no - 1
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Date  - 15/01/2024

# Question no 1 - Write a python program to create a dictionary which contains
#students name and marks.
# iterate over the dictionary and apply below conditions to print their grades
#for  5 subjects
# a) Marks greater than or equal to 70 - Distinction
# b) Marks between 60 - 90 - First Class
# c) Marls between 50 - 59 - Second Class
# d) Marks between 40 - 49 - Pass
# e) Marks less than 40 - fail

def calculate_grade(average_mark):
    if average_mark >= 70:
        return "Distinction"
    elif 60 <= average_mark < 70:
        return "First Class"
    elif 50 <= average_mark < 60:
        return "Second Class"
    elif 40 <= average_mark < 50:
        return "Pass"
    else:
        return "Fail"

students_data = {
    "Alice": [75, 80, 85, 90, 78],
    "Bob": [62, 68, 72, 56, 80],
    "Charlie": [45, 50, 55, 60, 58],
    "David": [75, 80, 70, 68, 72],
    "Eve": [38, 40, 42, 35, 30]
}
```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data Scien
ece\Practical\EXP - 1\part1.py"
Alice's grade: Distinction
Bob's grade: First Class
Charlie's grade: Second Class
David's grade: Distinction
Eve's grade: Fail

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```



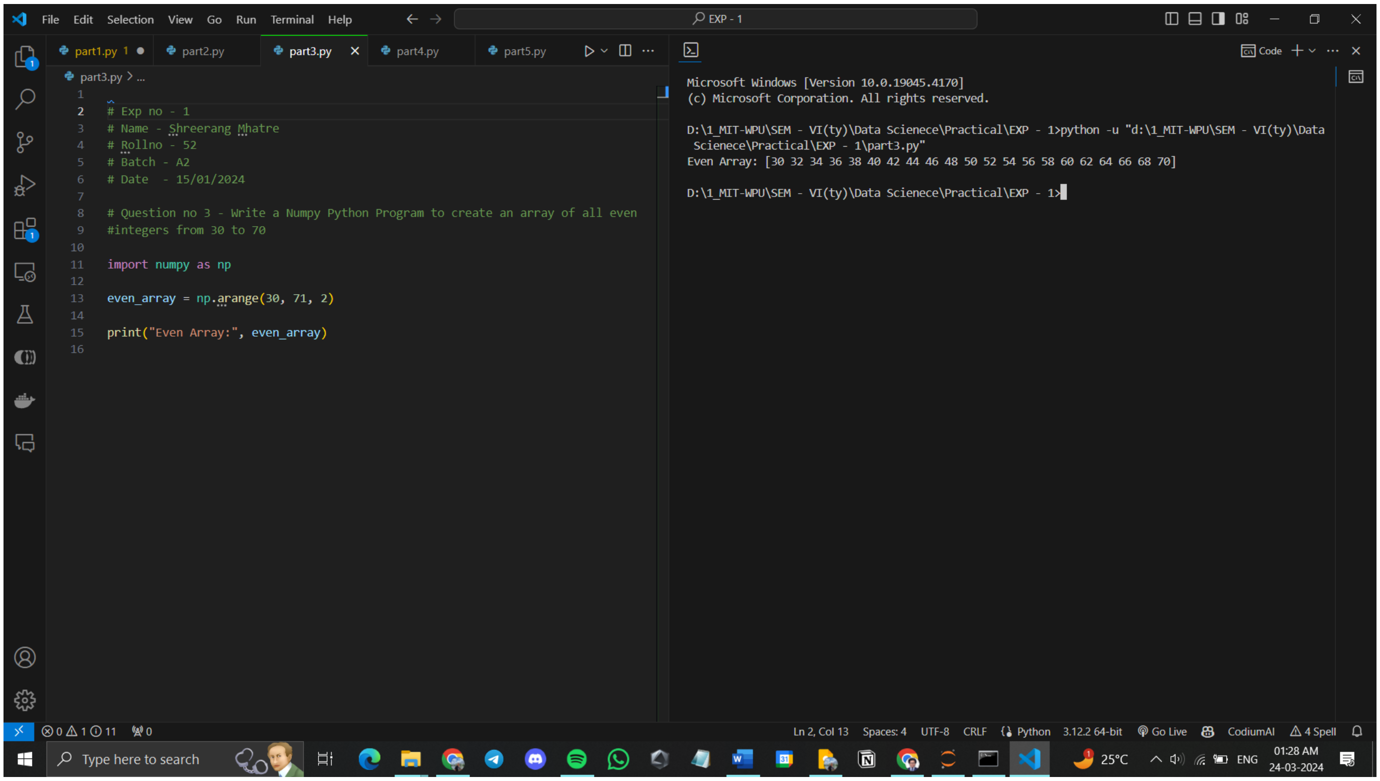
```python
# a) Marks greater than or equal to 70 - Distinction
# b) Marks between 60 - 90 - First Class
# c) Marls between 50 - 59 - Second Class
# d) Marks between 40 - 49 - Pass
# e) Marks less than 40 - fail

def calculate_grade(average_mark):
    if average_mark >= 70:
        return "Distinction"
    elif 60 <= average_mark < 70:
        return "First Class"
    elif 50 <= average_mark < 60:
        return "Second Class"
    elif 40 <= average_mark < 50:
        return "Pass"
    else:
        return "Fail"

students_data = {
    "Alice": [75, 80, 85, 90, 78],
    "Bob": [62, 68, 72, 56, 80],
    "Charlie": [45, 50, 55, 60, 58],
    "David": [75, 80, 70, 68, 72],
    "Eve": [38, 40, 42, 35, 30]
}

for student, marks in students_data.items():
    average_mark = sum(marks) / len(marks)
    grade = calculate_grade(average_mark)
    print(f"{student}'s grade: {grade}")
```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data Scien
ece\Practical\EXP - 1\part1.py"
Alice's grade: Distinction
Bob's grade: First Class
Charlie's grade: Second Class
David's grade: Distinction
Eve's grade: Fail

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```


File Edit Selection View Go Run Terminal Help
EXP - 1
part1.py 1
part2.py
part3.py
part4.py
part5.py
Code
part2.py > ...
Ln 12, Col 29 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live CodiumAI 3 Spell
Type here to search
25°C ENG 01:27 AM 24-03-2024

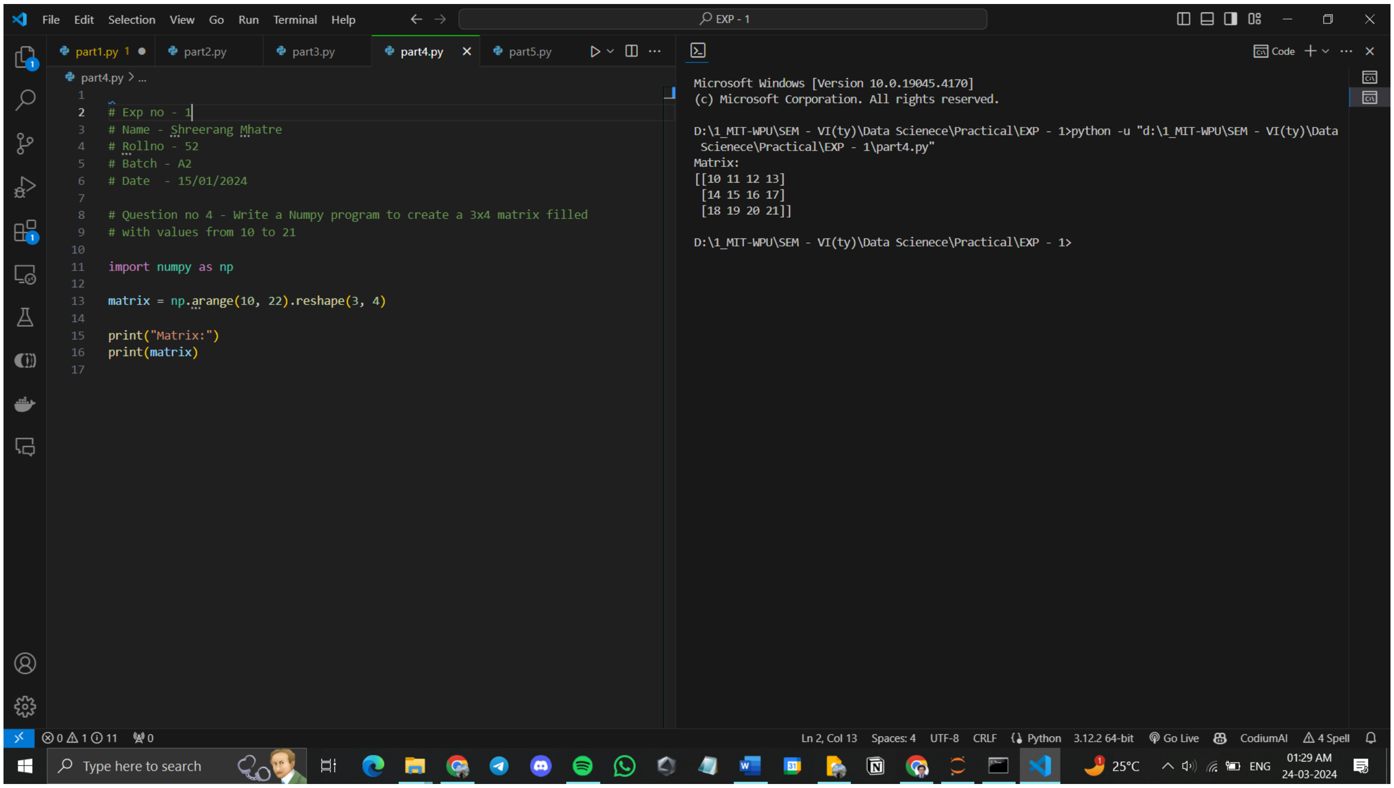
```python

# Exp no - 1
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Date  - 15/01/2024

# Question no 2 - Write a python program to create a ID array of numbers from 0 to 9

id_array = list(range(10))

print("ID Array:", id_array)
```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 1\part2.py"
ID Array: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```


File Edit Selection View Go Run Terminal Help
EXP - 1
part1.py 1
part2.py
part3.py
part4.py
part5.py
Code
part3.py > ...
Ln 2, Col 13 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit Go Live CodiumAI 4 Spell
Type here to search
25°C ENG 01:28 AM 24-03-2024

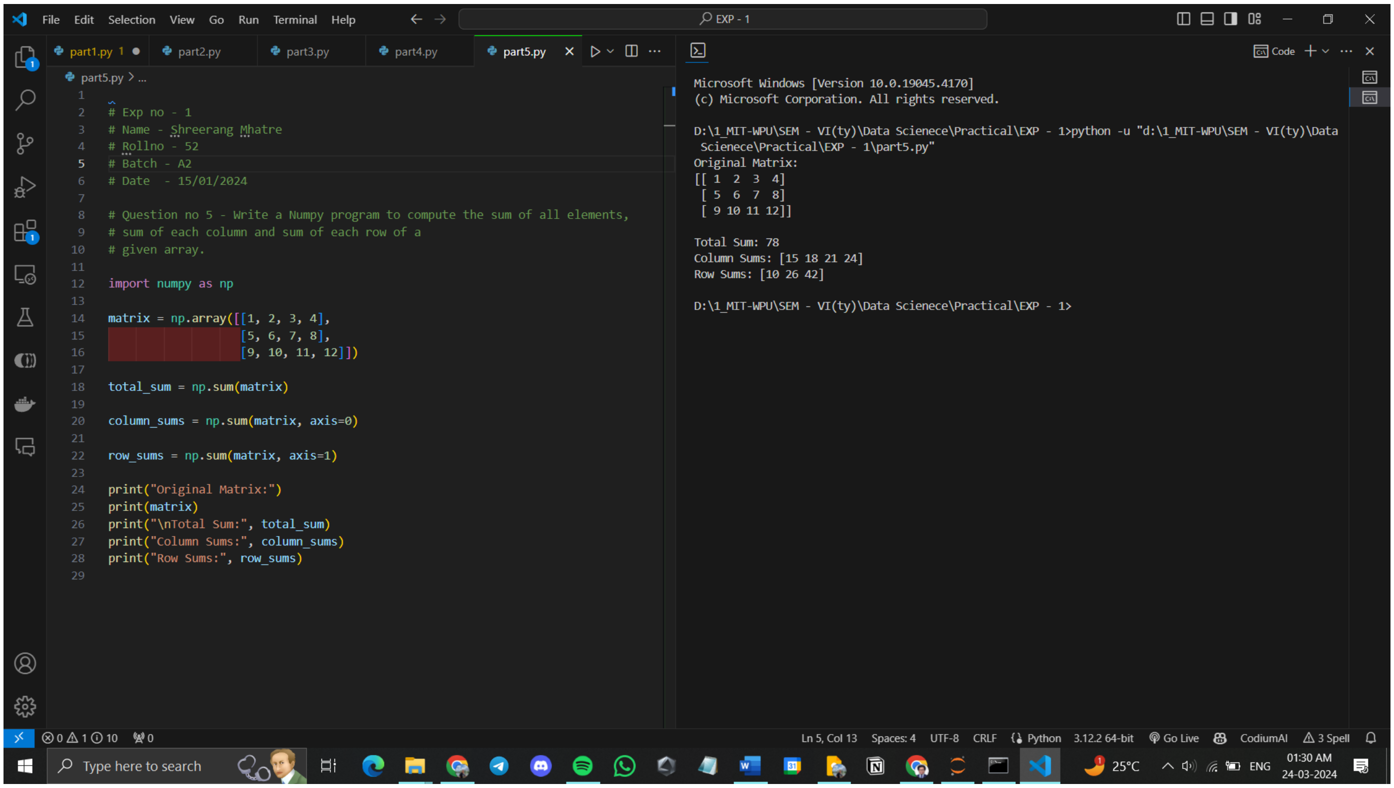
```python

# Exp no - 1
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Date  - 15/01/2024

# Question no 3 - Write a Numpy Python Program to create an array of all even
#integers from 30 to 70

import numpy as np

even_array = np.arange(30, 71, 2)

print("Even Array:", even_array)

```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 1\part3.py"
Even Array: [30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70]

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```



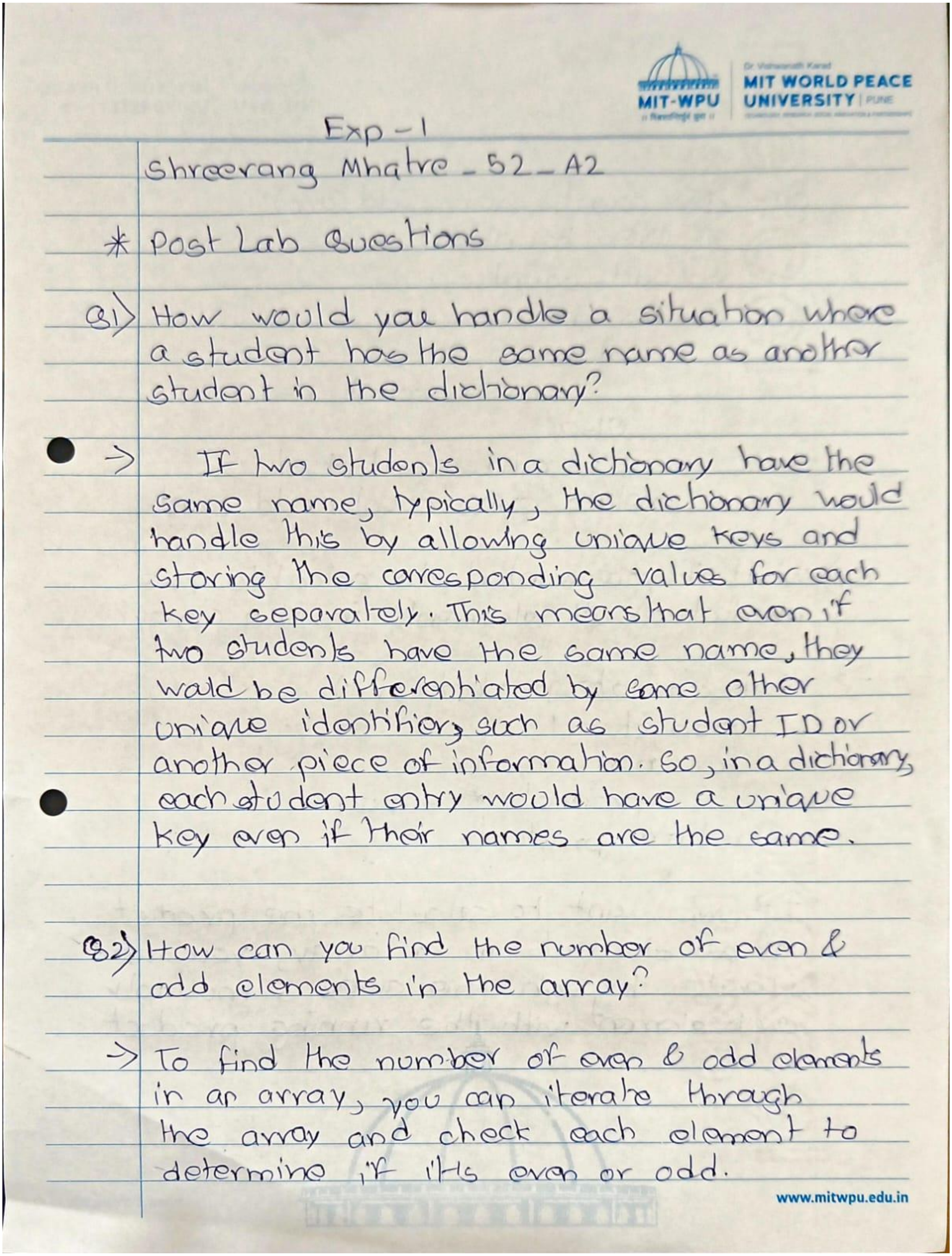
```python

# Exp no - 1
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Date  - 15/01/2024

# Question no 4 - Write a Numpy program to create a 3x4 matrix filled
# with values from 10 to 21

import numpy as np

matrix = np.arange(10, 22).reshape(3, 4)

print("Matrix:")
print(matrix)

```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 1\part4.py"
Matrix:
[[10 11 12 13]
 [14 15 16 17]
 [18 19 20 21]]

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```



```python

# Exp no - 1
# Name - Shreerang Mhatre
# Rollno - 52
# Batch - A2
# Date  - 15/01/2024

# Question no 5 - Write a Numpy program to compute the sum of all elements,
# sum of each column and sum of each row of a
# given array.

import numpy as np

matrix = np.array([[1, 2, 3, 4],
                   [5, 6, 7, 8],
                   [9, 10, 11, 12]])

total_sum = np.sum(matrix)

column_sums = np.sum(matrix, axis=0)

row_sums = np.sum(matrix, axis=1)

print("Original Matrix:")
print(matrix)
print("\nTotal Sum:", total_sum)
print("Column Sums:", column_sums)
print("Row Sums:", row_sums)

```

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>python -u "d:\1_MIT-WPU\SEM - VI(ty)\Data
 Scienece\Practical\EXP - 1\part5.py"
Original Matrix:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Total Sum: 78
Column Sums: [15 18 21 24]
Row Sums: [10 26 42]

D:\1_MIT-WPU\SEM - VI(ty)\Data Scienece\Practical\EXP - 1>
```



Exp - 1

Shreerang Mhatre _ 52 _ A2

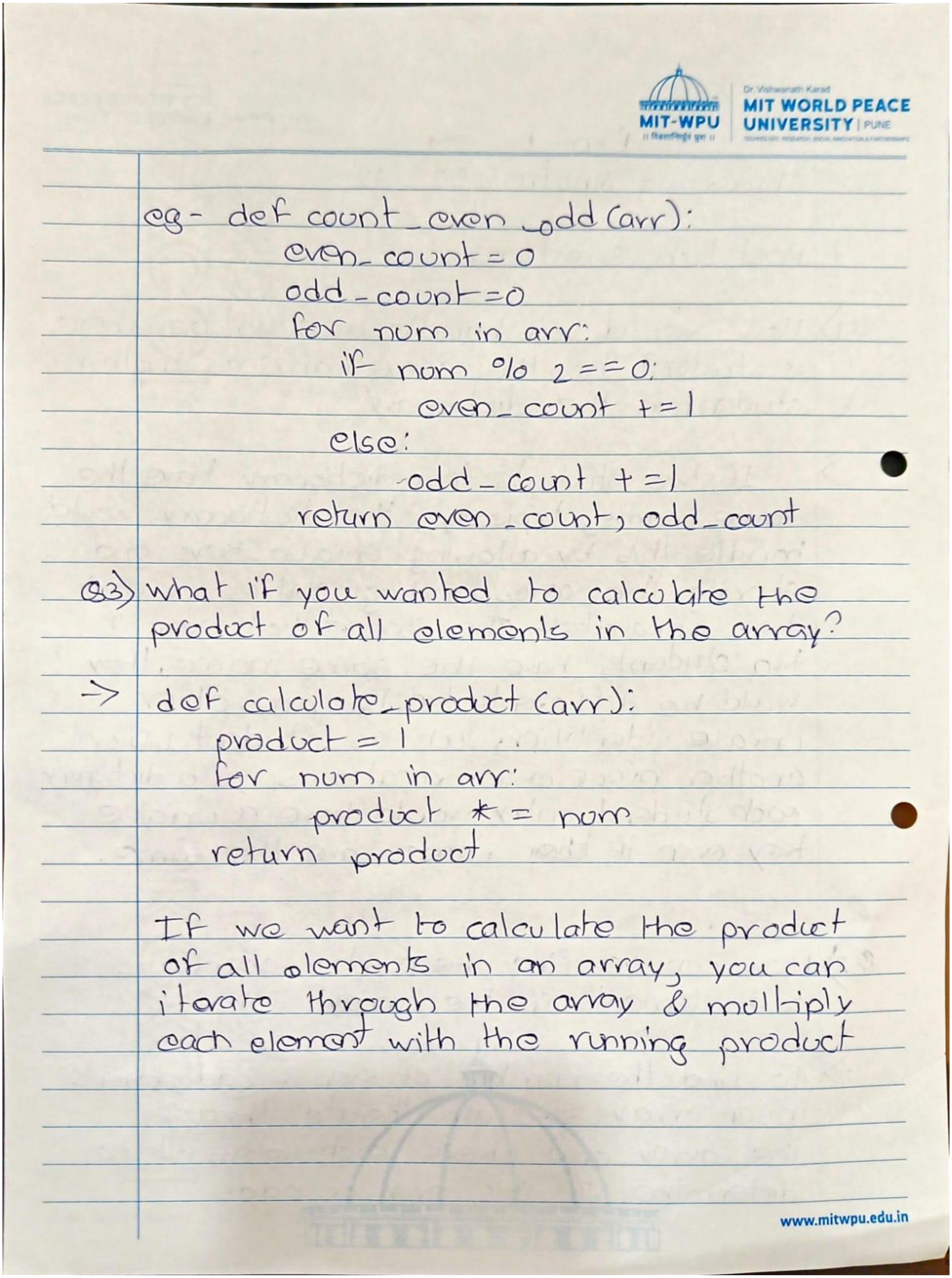
* Post Lab Questions

Q1) How would you handle a situation where a student has the same name as another student in the dictionary?

-> If two students in a dictionary have the same name, typically, the dictionary would handle this by allowing unique keys and storing the corresponding values for each key separately. This means that even if two students have the same name, they would be differentiated by some other unique identifier, such as student ID or another piece of information. So, in a dictionary, each student entry would have a unique key even if their names are the same.

Q2) How can you find the number of even & odd elements in the array?

-> To find the number of even & odd elements in an array, you can iterate through the array and check each element to determine if it's even or odd.



```
eg- def count_even_odd(arr):
        even_count = 0
        odd_count = 0
        for num in arr:
            if num % 2 == 0:
                even_count += 1
            else:
                odd_count += 1
        return even_count, odd_count
```

Q3) what if you wanted to calculate the product of all elements in the array?

```
-> def calculate_product(arr):
       product = 1
       for num in arr:
           product *= num
       return product
```

If we want to calculate the product of all elements in an array, you can iterate through the array & multiply each element with the running product