A

Lab Records of Programming with Problem Solving Lab

Bachelor of Computer Application -I Sem

RUNGTA INTERNATIONAL SKILLS UNIVERSITY

SESSION: 2025-26

Submitted By: -
Samiksha Singh
REF/2025/03908

Kavita Kanwar
(Assistant Professor)

Submitted To:

RUNGTA INTERNATIONAL SKILLS UNIVERSITY, CG

SCHOOL OF INFORMATION TECHNOLOGY

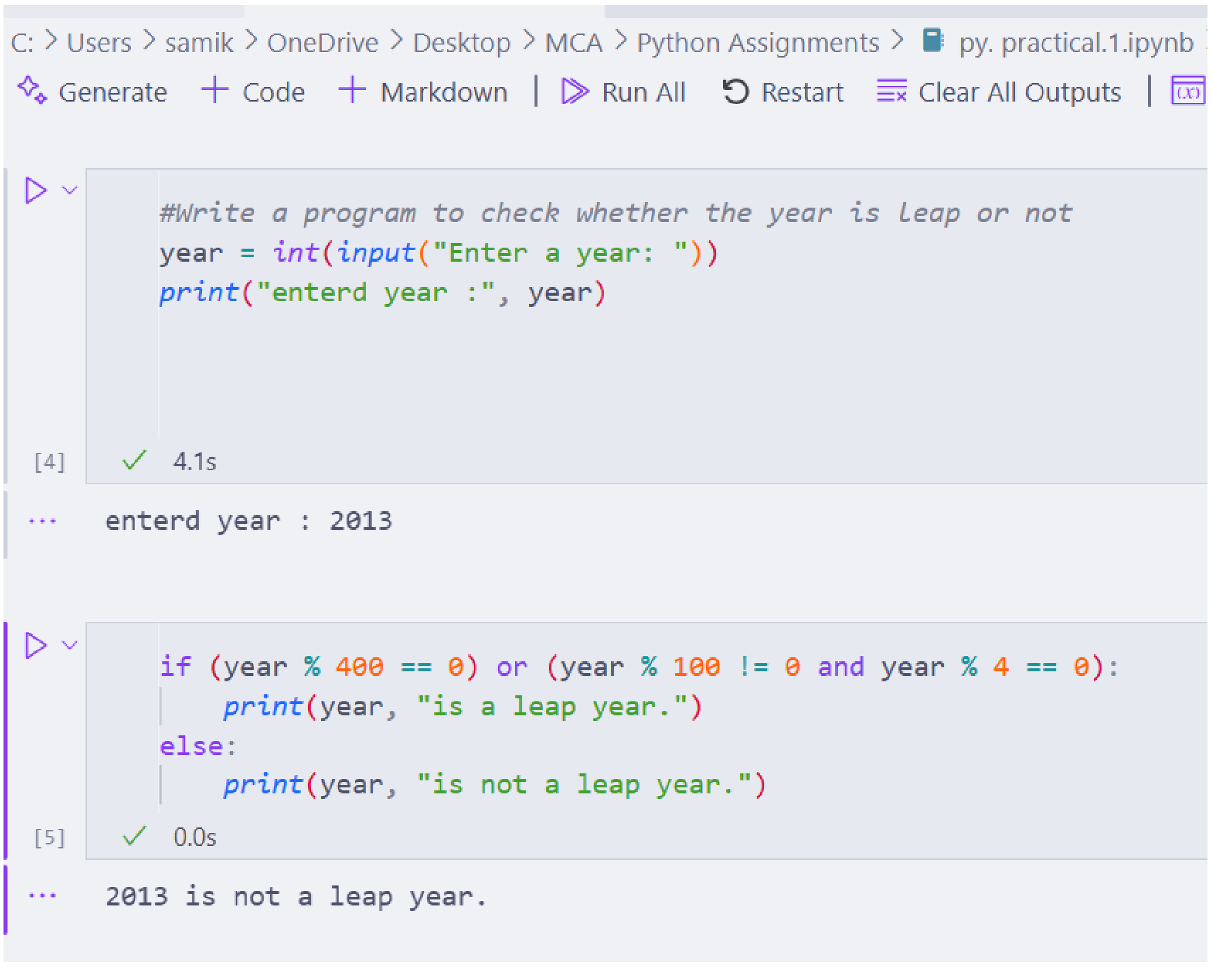
Index

Sno	Name of Practical	Submission Date	Remarks
1.	Write a program to check whether the year is a leap year or not		
2.	Write a program to count the number of vowels in a string.		
3.	Write a program to reverse a number.		
4.	Write a program to find mean, median and mode of a given number.		
5.	Write a Python program to reverse only the vowels in a given string, keeping other characters in their original positions.		
6.	Create a script that takes an integer and displays its binary, octal, and hexadecimal representations neatly formatted.		
7.	Given a list of items (possibly with duplicates), write a program that removes duplicates and displays the sorted list.		
8.	Accept a list of students and their marks as tuples. Display the name of the student with the highest marks.		

9.	Read data from a CSV file containing employee details (name, department, salary) and display the average salary by department.		

Practical-1

Aim: Write a program to check whether the year is leap or not


C: > Users > samik > OneDrive > Desktop > MCA > Python Assignments > py. practical.1.ipynb
Generate
Code
Markdown
Run All
Restart
Clear All Outputs

```python
#Write a program to check whether the year is leap or not
year = int(input("Enter a year: "))
print("enterd year :", year)
```

[4] ✓ 4.1s

```
enterd year : 2013
```

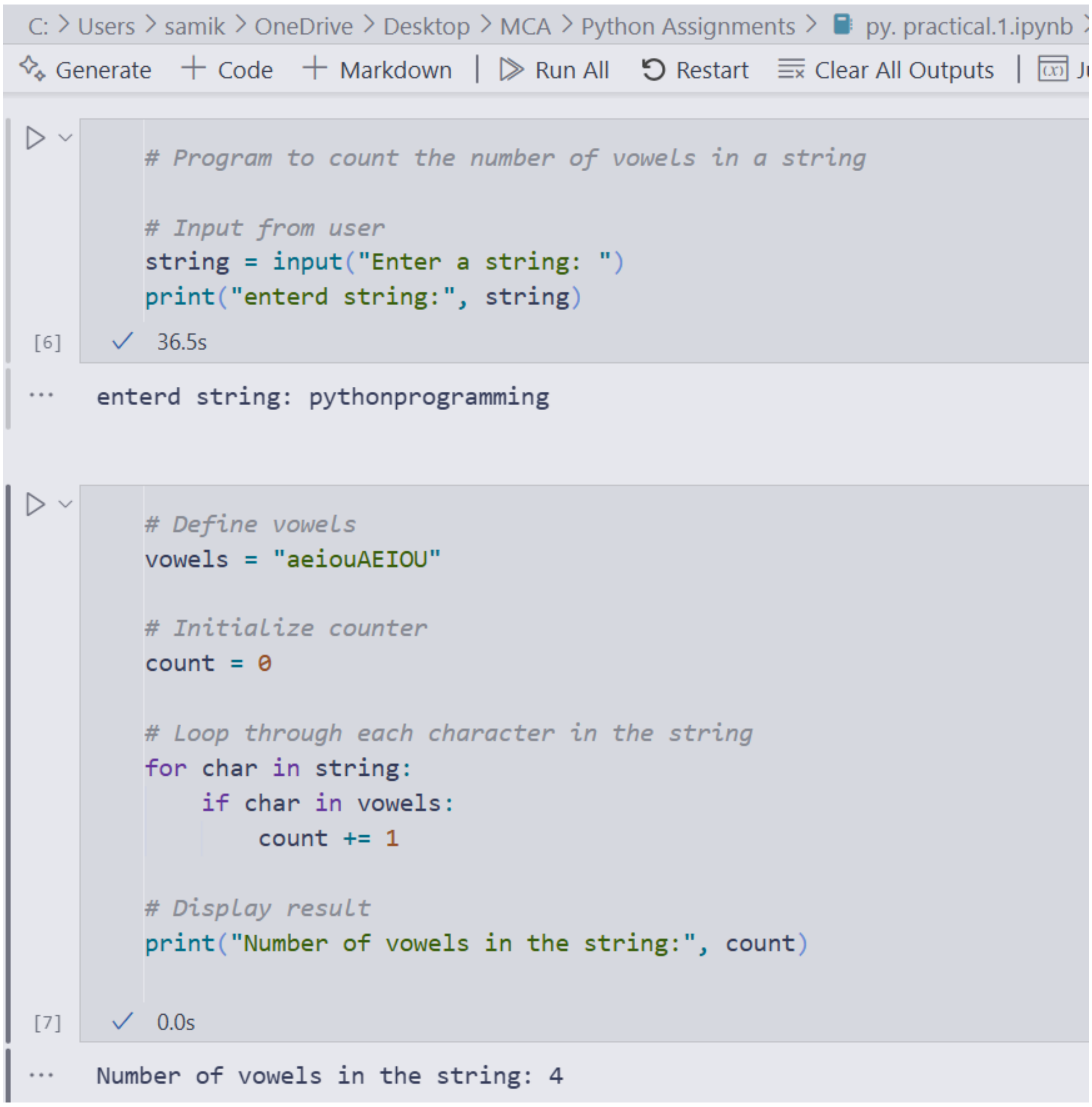
```python
if (year % 400 == 0) or (year % 100 != 0 and year % 4 == 0):
    print(year, "is a leap year.")
else:
    print(year, "is not a leap year.")
```

[5] ✓ 0.0s

```
2013 is not a leap year.
```

Practical-2

Aim: Write a program to count no. of vowels in a string.


C: > Users > samik > OneDrive > Desktop > MCA > Python Assignments > py. practical.1.ipynb
Generate
Code
Markdown
Run All
Restart
Clear All Outputs

```python
# Program to count the number of vowels in a string

# Input from user
string = input("Enter a string: ")
print("enterd string:", string)
```

[6] ✓ 36.5s

```
enterd string: pythonprogramming
```

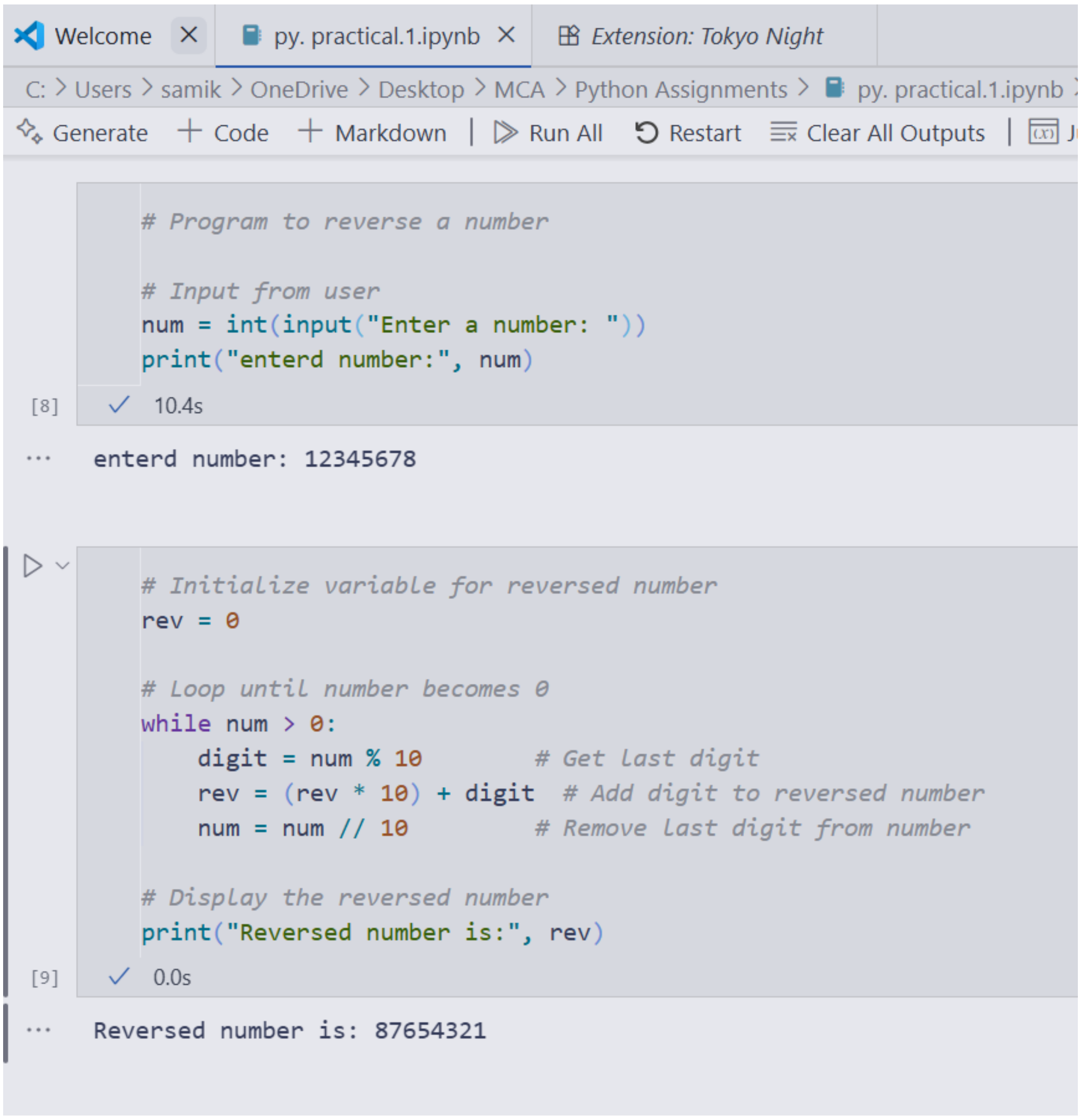
```python
# Define vowels
vowels = "aeiouAEIOU"

# Initialize counter
count = 0

# Loop through each character in the string
for char in string:
    if char in vowels:
        count += 1

# Display result
print("Number of vowels in the string:", count)
```

[7] ✓ 0.0s

```
Number of vowels in the string: 4
```

Practical-3

Aim: Write a program to reverse a number.



```python
# Program to reverse a number

# Input from user
num = int(input("Enter a number: "))
print("enterd number:", num)
```

[8] ✓ 10.4s

```
enterd number: 12345678
```

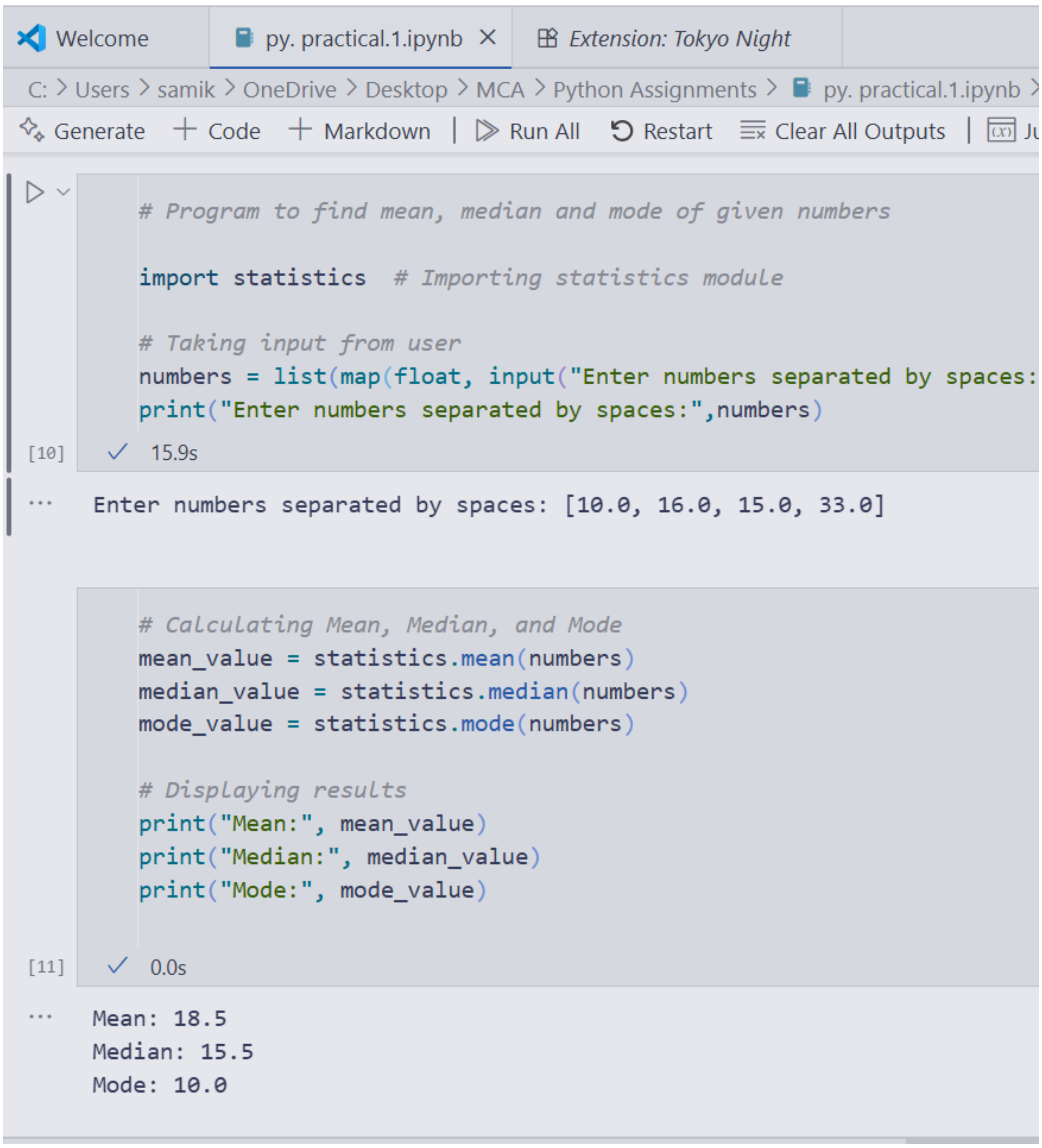
```python
# Initialize variable for reversed number
rev = 0

# Loop until number becomes 0
while num > 0:
    digit = num % 10          # Get last digit
    rev = (rev * 10) + digit  # Add digit to reversed number
    num = num // 10           # Remove last digit from number

# Display the reversed number
print("Reversed number is:", rev)
```

[9] ✓ 0.0s

```
Reversed number is: 87654321
```

Practical-4

Aim: Write a program to find mean, median and mode of given number.



```
# Program to find mean, median and mode of given numbers

import statistics  # Importing statistics module

# Taking input from user
numbers = list(map(float, input("Enter numbers separated by spaces:
print("Enter numbers separated by spaces:",numbers)
```

[10] ✓ 15.9s

```
Enter numbers separated by spaces: [10.0, 16.0, 15.0, 33.0]
```

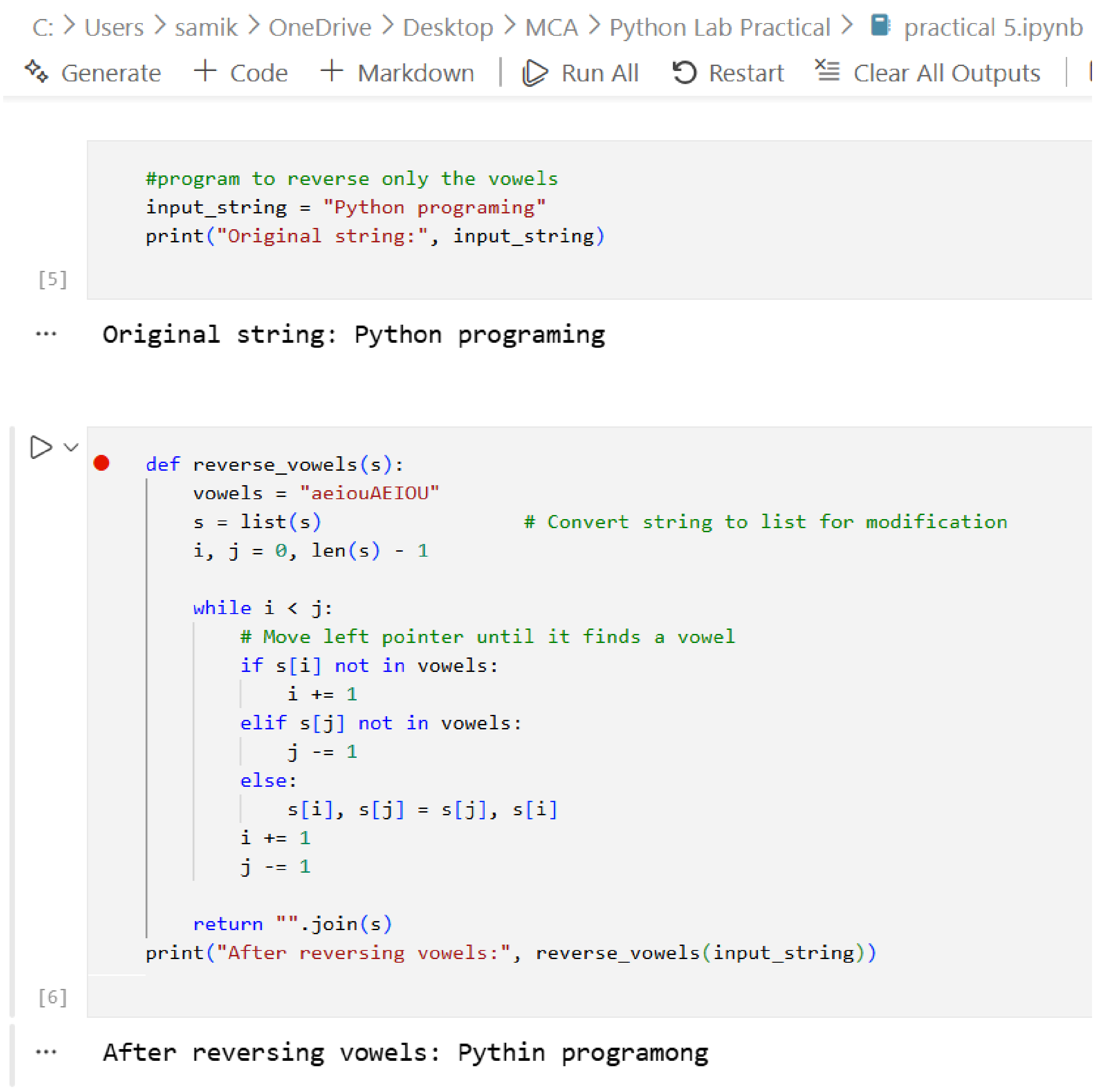
```
# Calculating Mean, Median, and Mode
mean_value = statistics.mean(numbers)
median_value = statistics.median(numbers)
mode_value = statistics.mode(numbers)

# Displaying results
print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
```

[11] ✓ 0.0s

```
Mean: 18.5
Median: 15.5
Mode: 10.0
```

Practical – 5

Aim: Write a Python program to reverse only the vowels in a given string, keeping other characters in their original positions.


C: > Users > samik > OneDrive > Desktop > MCA > Python Lab Practical > practical 5.ipynb
Generate + Code + Markdown | Run All Restart Clear All Outputs

```
#program to reverse only the vowels
input_string = "Python programing"
print("Original string:", input_string)
```

[5]

```
Original string: Python programing
```

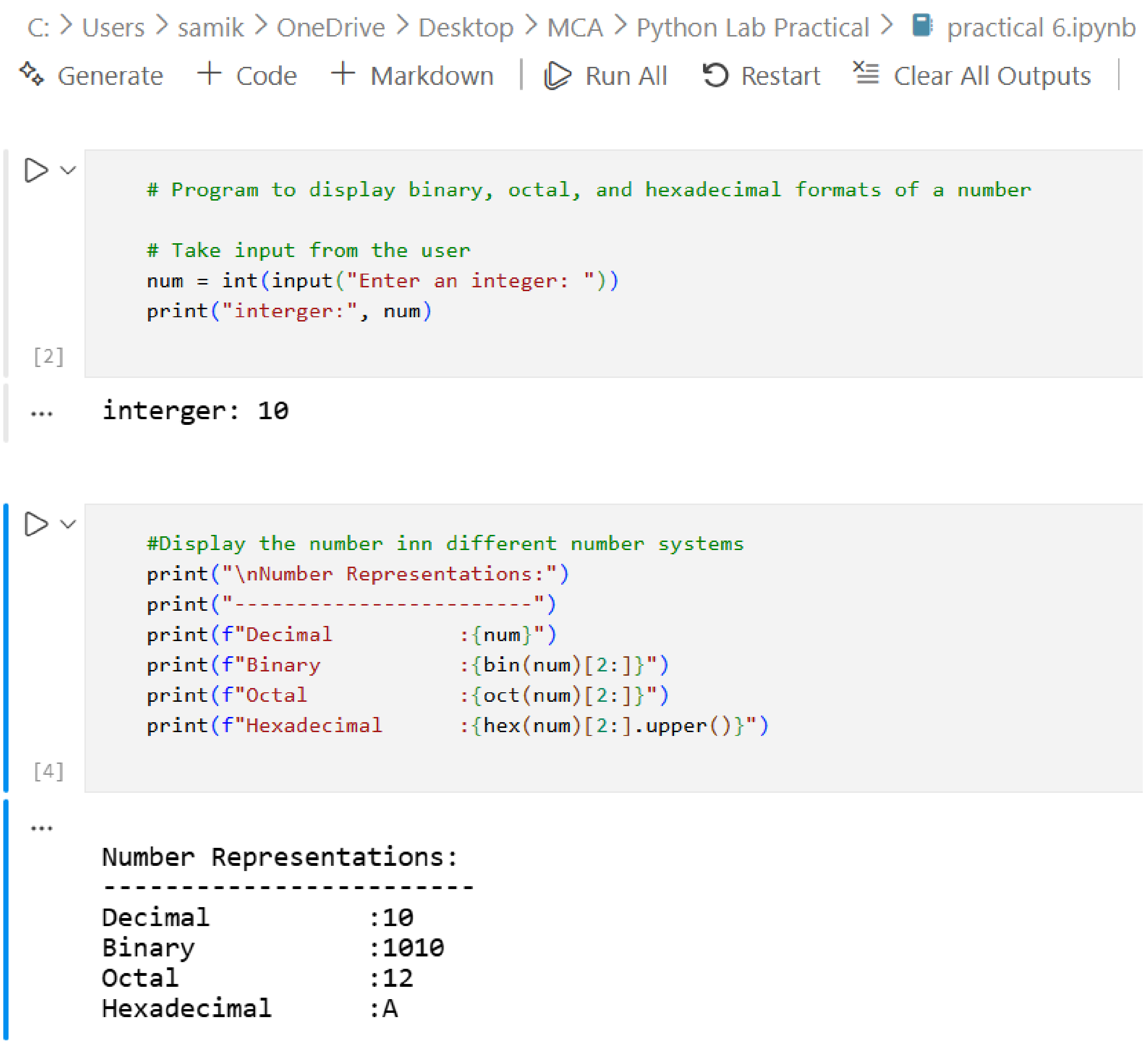
```
def reverse_vowels(s):
    vowels = "aeiouAEIOU"
    s = list(s)             # Convert string to list for modification
    i, j = 0, len(s) - 1

    while i < j:
        # Move left pointer until it finds a vowel
        if s[i] not in vowels:
            i += 1
        elif s[j] not in vowels:
            j -= 1
        else:
            s[i], s[j] = s[j], s[i]
            i += 1
            j -= 1

    return "".join(s)
print("After reversing vowels:", reverse_vowels(input_string))
```

[6]

```
After reversing vowels: Pythin programong
```

Practical – 6

Aim: Create a script that takes an integer and displays its binary, octal, and hexadecimal representations neatly formatted.


C: > Users > samik > OneDrive > Desktop > MCA > Python Lab Practical > practical 6.ipynb
Generate + Code + Markdown | Run All Restart Clear All Outputs

```
# Program to display binary, octal, and hexadecimal formats of a number

# Take input from the user
num = int(input("Enter an integer: "))
print("interger:", num)
```

[2]

```
interger: 10
```

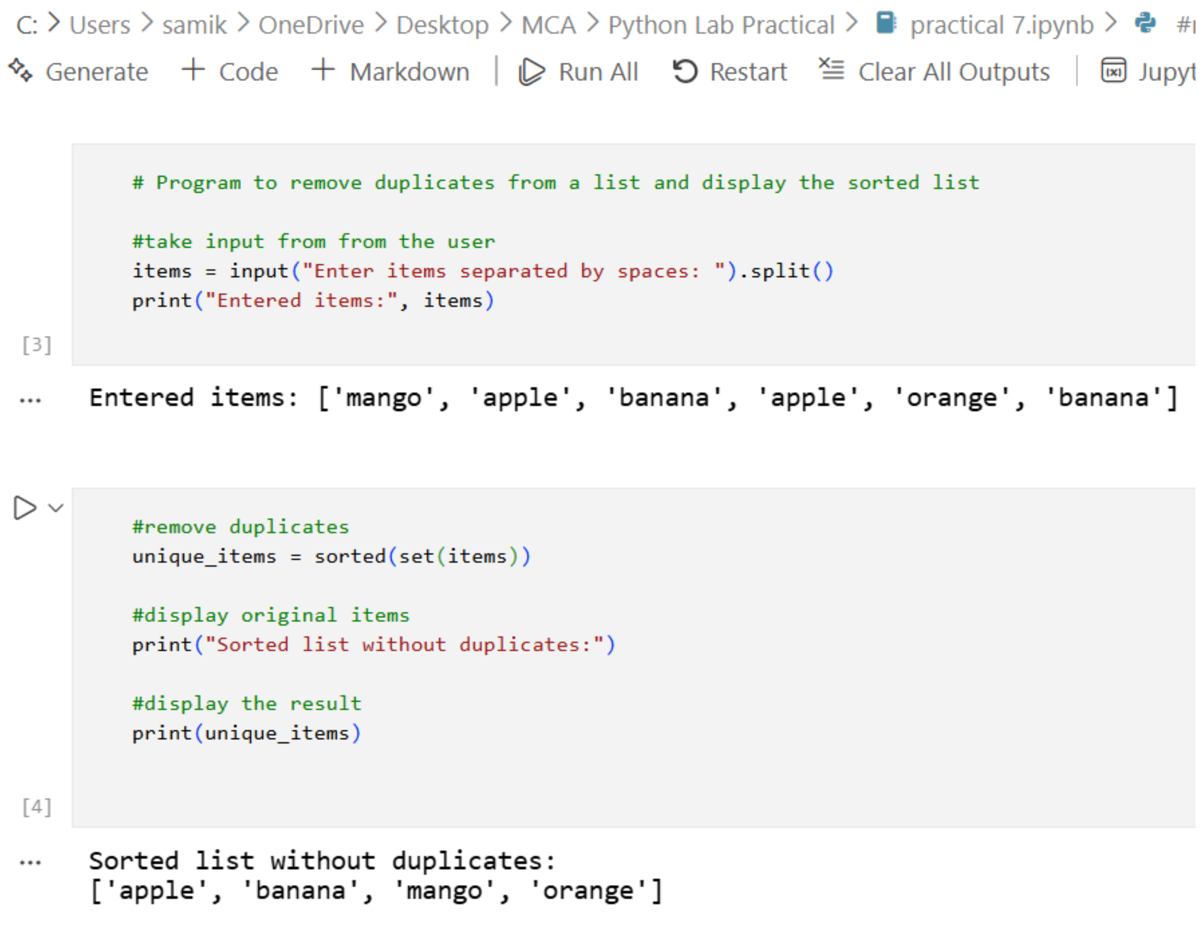
```
#Display the number inn different number systems
print("\nNumber Representations:")
print("-----------------------")
print(f"Decimal          :{num}")
print(f"Binary           :{bin(num)[2:]}")
print(f"Octal            :{oct(num)[2:]}")
print(f"Hexadecimal      :{hex(num)[2:].upper()}")
```

[4]

```

Number Representations:
------------------------
Decimal          :10
Binary           :1010
Octal            :12
Hexadecimal      :A
```

Practical – 7

Aim: Given a list of items (possibly with duplicates), write a program that removes duplicates and displays the sorted list.


C: > Users > samik > OneDrive > Desktop > MCA > Python Lab Practical > practical 7.ipynb > #
Generate
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Jupyt

```
# Program to remove duplicates from a list and display the sorted list

#take input from from the user
items = input("Enter items separated by spaces: ").split()
print("Entered items:", items)
```

[3]

```
Entered items: ['mango', 'apple', 'banana', 'apple', 'orange', 'banana']
```

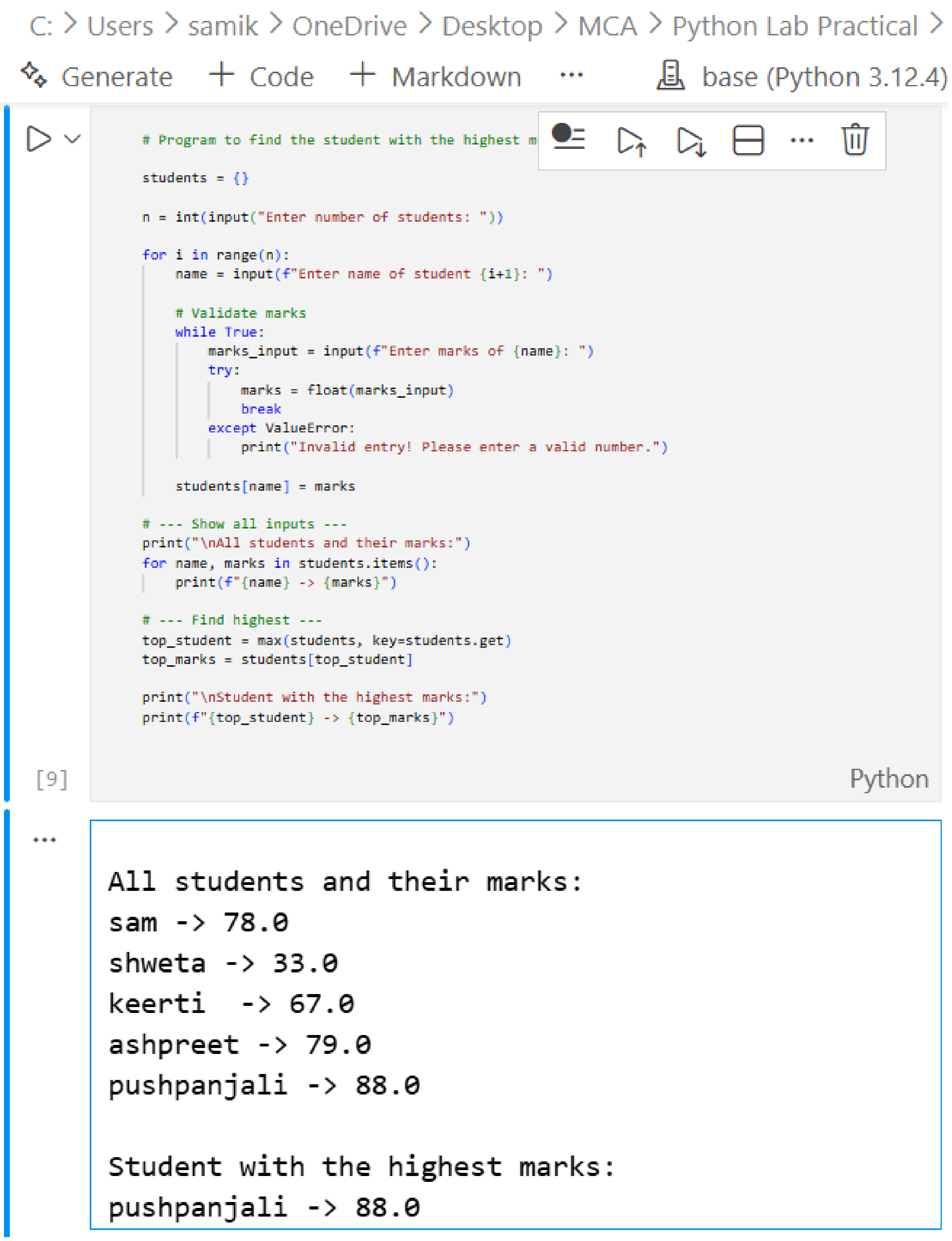
```
#remove duplicates
unique_items = sorted(set(items))

#display original items
print("Sorted list without duplicates:")

#display the result
print(unique_items)
```

[4]

```
Sorted list without duplicates:
['apple', 'banana', 'mango', 'orange']
```

Practical – 8

Aim: Accept a list of students and their marks as tuples. Display the name of the student with the highest marks.


C: > Users > samik > OneDrive > Desktop > MCA > Python Lab Practical >
Generate
+ Code
+ Markdown
base (Python 3.12.4)
[9]
Python

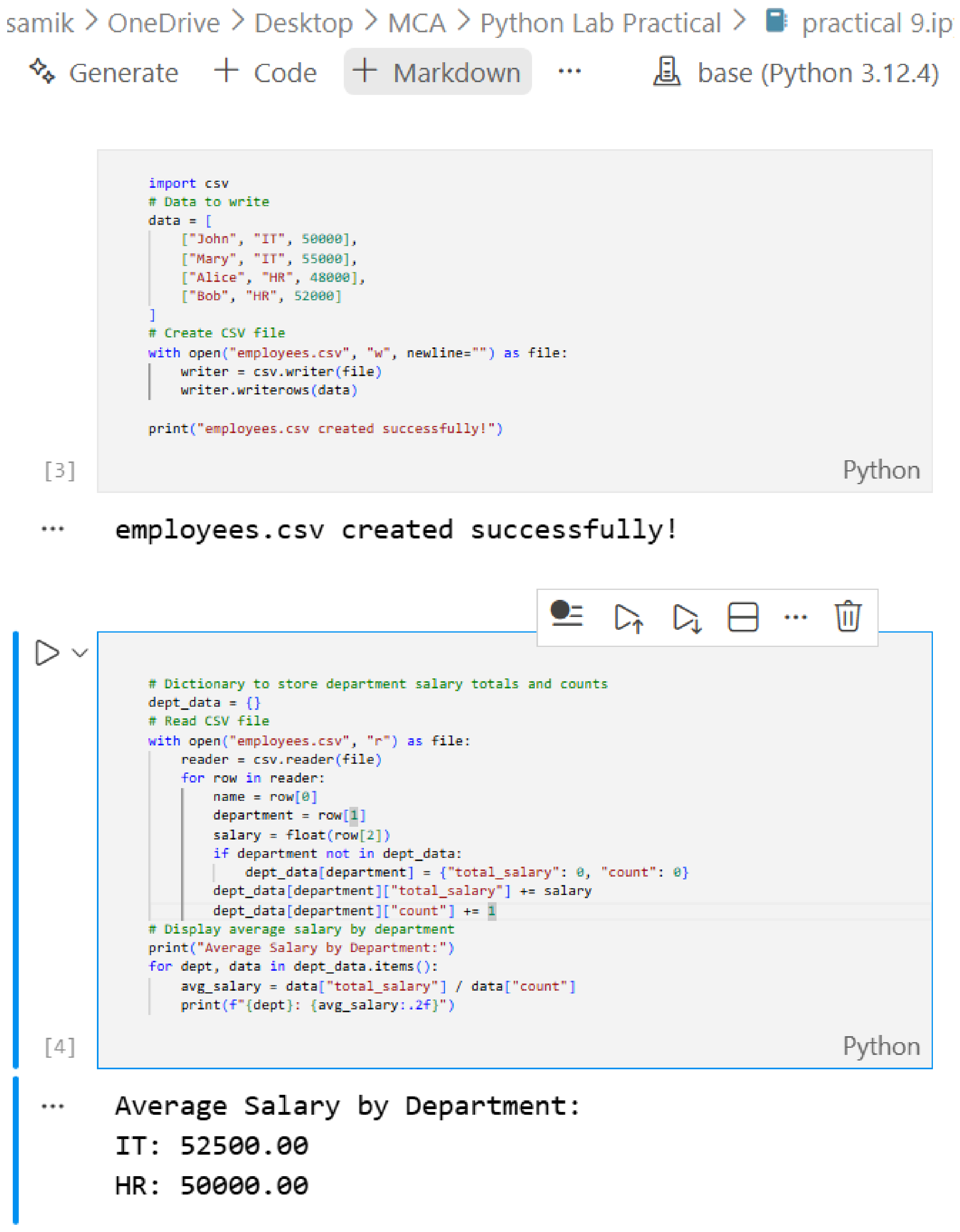
```python
# Program to find the student with the highest m

students = {}

n = int(input("Enter number of students: "))

for i in range(n):
    name = input(f"Enter name of student {i+1}: ")

    # Validate marks
    while True:
        marks_input = input(f"Enter marks of {name}: ")
        try:
            marks = float(marks_input)
            break
        except ValueError:
            print("Invalid entry! Please enter a valid number.")

    students[name] = marks

# --- Show all inputs ---
print("\nAll students and their marks:")
for name, marks in students.items():
    print(f"{name} -> {marks}")

# --- Find highest ---
top_student = max(students, key=students.get)
top_marks = students[top_student]

print("\nStudent with the highest marks:")
print(f"{top_student} -> {top_marks}")
```

```

All students and their marks:
sam -> 78.0
shweta -> 33.0
keerti  -> 67.0
ashpreet -> 79.0
pushpanjali -> 88.0

Student with the highest marks:
pushpanjali -> 88.0
```

Practical – 9

Aim: Read data from a CSV file containing employee details (name, department, salary) and display the average salary by department.

Input (CSV):{(John,IT,50000),(Mary,IT,55000),(Alice,HR,48000),(Bob,HR,52000)}



samik > OneDrive > Desktop > MCA > Python Lab Practical > practical 9.ip

Generate + Code + Markdown ··· base (Python 3.12.4)

```
import csv
# Data to write
data = [
    ["John", "IT", 50000],
    ["Mary", "IT", 55000],
    ["Alice", "HR", 48000],
    ["Bob", "HR", 52000]
]
# Create CSV file
with open("employees.csv", "w", newline="") as file:
    writer = csv.writer(file)
    writer.writerows(data)

print("employees.csv created successfully!")
```

[3] Python

```
employees.csv created successfully!
```

```
# Dictionary to store department salary totals and counts
dept_data = {}
# Read CSV file
with open("employees.csv", "r") as file:
    reader = csv.reader(file)
    for row in reader:
        name = row[0]
        department = row[1]
        salary = float(row[2])
        if department not in dept_data:
            dept_data[department] = {"total_salary": 0, "count": 0}
        dept_data[department]["total_salary"] += salary
        dept_data[department]["count"] += 1
# Display average salary by department
print("Average Salary by Department:")
for dept, data in dept_data.items():
    avg_salary = data["total_salary"] / data["count"]
    print(f"{dept}: {avg_salary:.2f}")
```

[4] Python

```
Average Salary by Department:
IT: 52500.00
HR: 50000.00
```