Alliance School of Advanced Computing

Department of Computer Science and Engineering

Class Assignment-1

Course Code: 5CS1025
Course Title: Artificial Intelligence
Semester: 04
Class : AIML

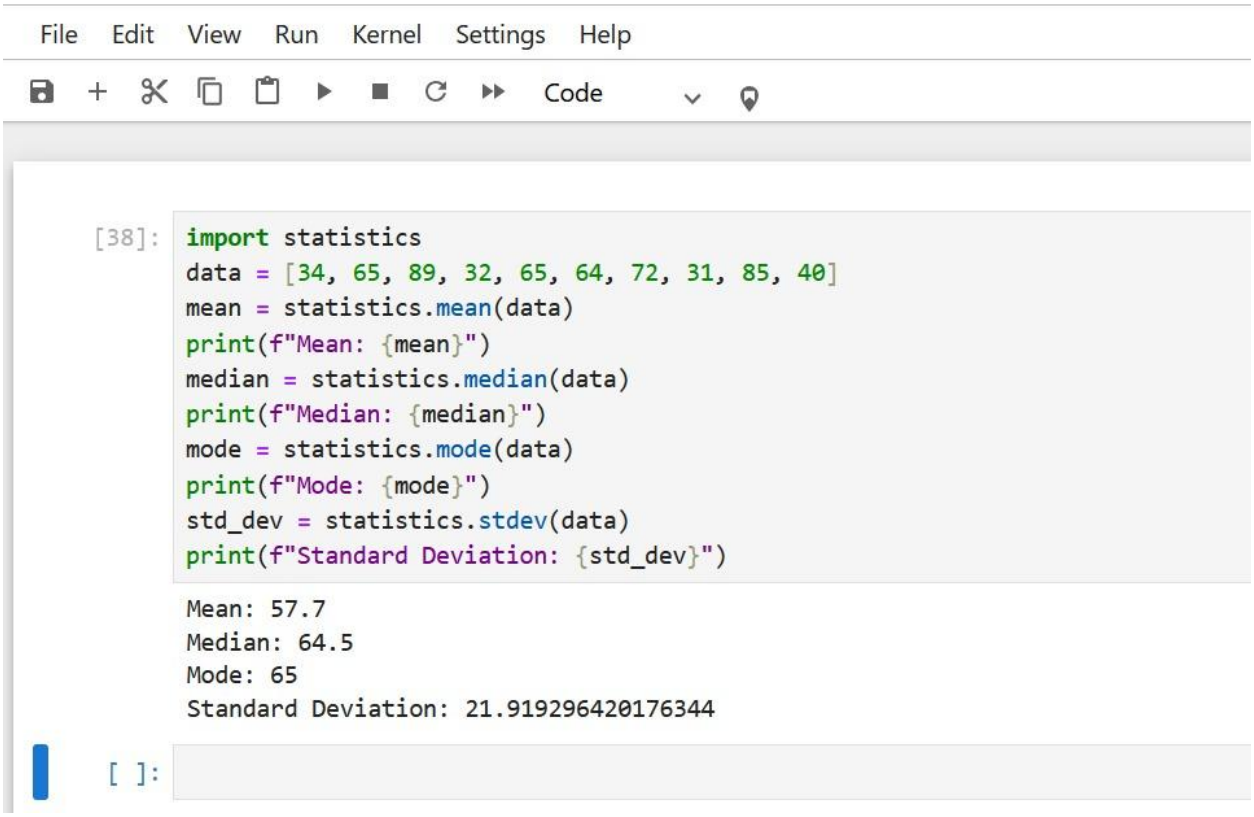
2024-25

Name : KAMMARA VAMSHI VARDHAN
Reg.NO :2023BCSE07AED241

1. Imagine you are tasked with designing a humanoid robot to assist in a home or office environment. The robot must be capable of interacting with people by **talking** and **listening**, **walking** to different locations, **seeing** and recognizing objects, and **learning** from its surroundings to adapt its behavior. What technologies, tools, and frameworks would you need to build such a robot? Give as flow chart

Frameworks	Tools	Technologies
Speech	Talking & Listening	ASR, TTS Dialog flow
Vision	Seeing & Recognizing Objects	Cameras depth sensors AI-powered object detection.
Mobility	Walking & Navigation	Motors, actuators, IMU sensors, SLAM-based navigation.
AI & Learning	Adapting to Environment	Machine learning Tensorflow Pytorch
Control System	Integration & Safety	Middleware like ROS

2. Calculate and interpret mean, median, mode, variance and standard deviation for a given dataset. Data =[15,21,29,21,15,24,32,21,15,30



3. You are analyzing a dataset that captures the daily performance and activity of a humanoid robot in a simulated environment. The dataset link robot_dataset(robot_dataset)_1.csv includes the following attributes

Interaction_Count: Number of conversations the robot had daily.
Steps_Walked: Total steps taken each day.
Objects_Recognized: Number of objects successfully identified by the robot.
Learning_Sessions: Number of learning tasks completed.
Energy_Consumption (kWh): Daily energy usage of robots.

Perform Basic Statistical Operations:

1) What is the **average (mean)** number of conversations the robot has daily?

2) Find the **total steps walked** by the robot over a given period.
3) Determine the **maximum and minimum energy consumption** in the dataset.
4) Calculate the **correlation** between the number of steps walked and energy consumption.
5) Analyze the **distribution** of objects recognized daily (e.g., histogram or box plot).
6) What is the **variance** in the number of learning sessions completed?

```
import pandas as pd
a=pd.read_csv("robot_dataset.csv")
a["Interaction_Count"].mean()
```

```
5.51
```

```
import pandas as pd
a=pd.read_csv("robot_dataset.csv")
a["Steps_Walked"].sum()
```

```
14379
```

```
import pandas as pd
a=pd.read_csv("robot_dataset.csv")
a["Energy_Consumption (kWh)"].max()
```

```
3.0
```

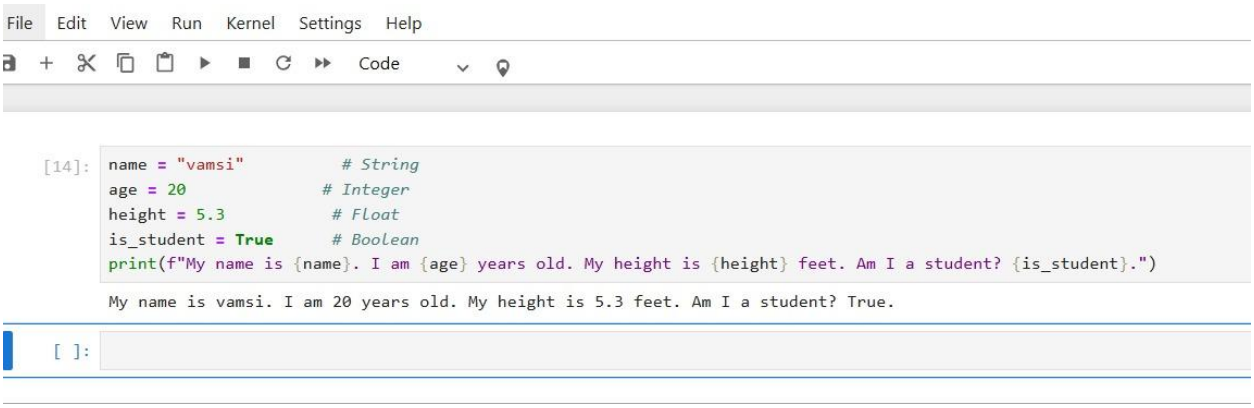
```
import pandas as pd
a=pd.read_csv("robot_dataset.csv")
a["Energy_Consumption (kWh)"].min()
```

```
1.0
```

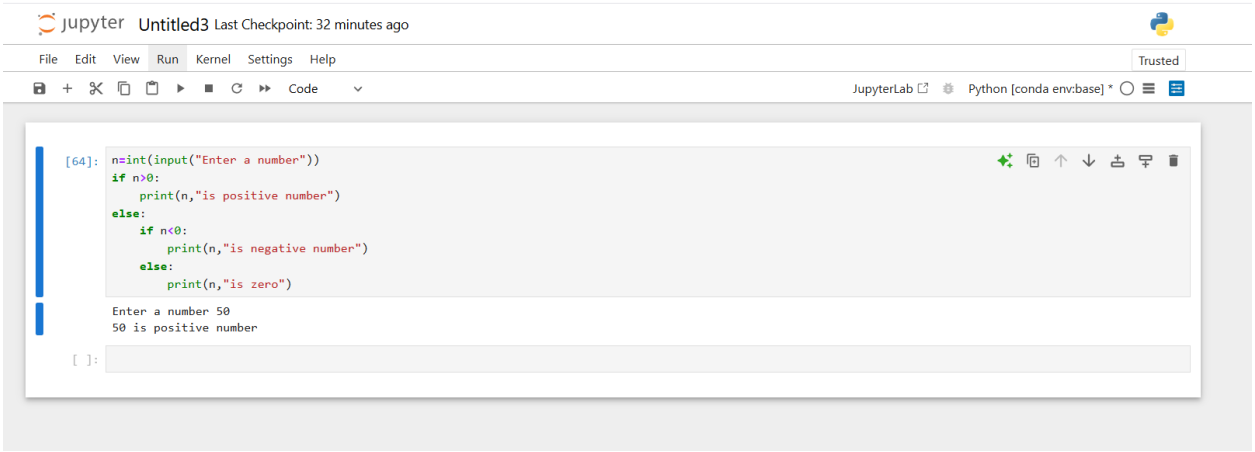
```
import pandas as pd
a=pd.read_csv("robot_dataset.csv")
a["Learning_Sessions"].var()
```

```
391.9422845691385
```

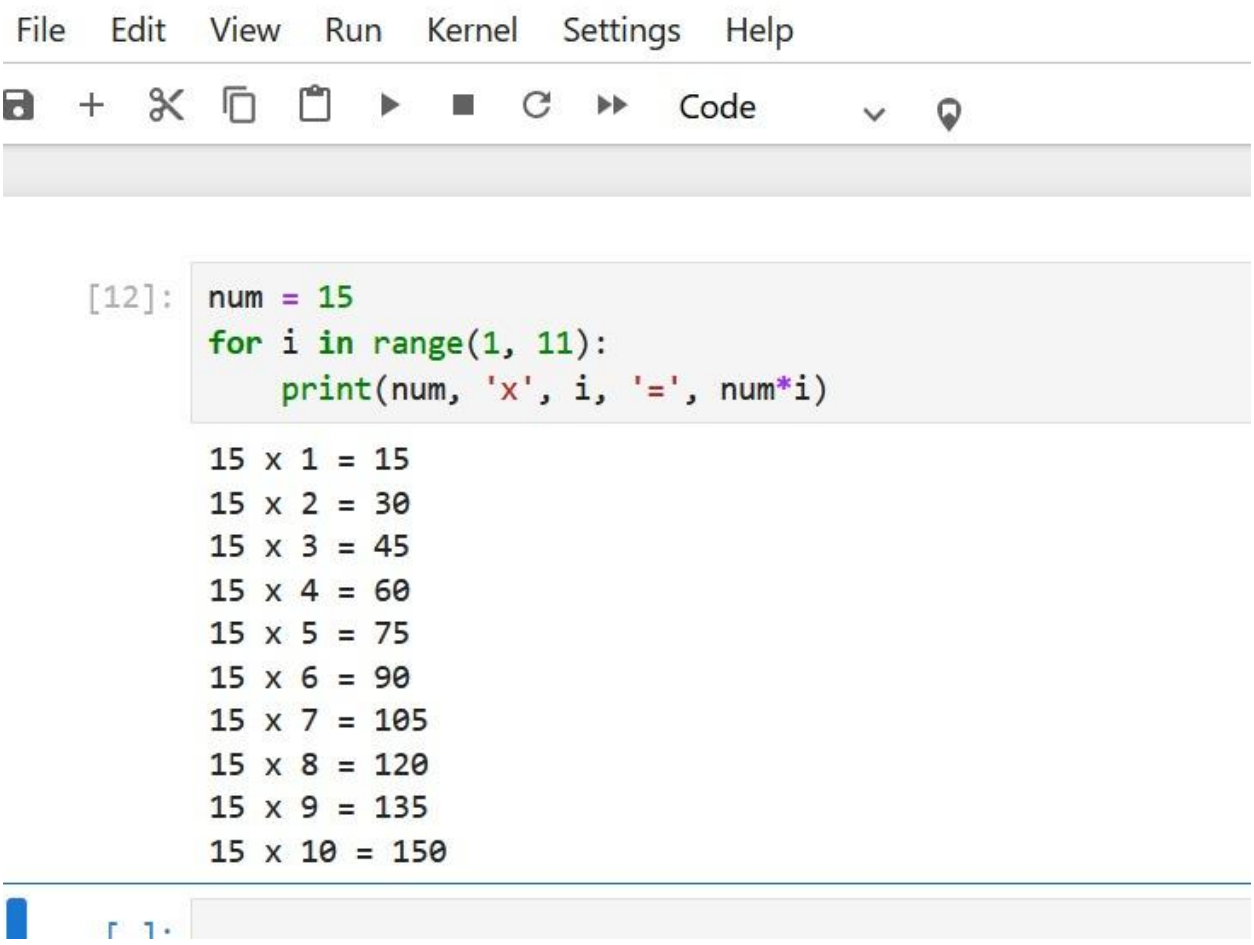
4. Write a Python program that declares variables of different data types (e.g., string, integer, float, and boolean). Output the variables in a sentence format using print() and f-strings.



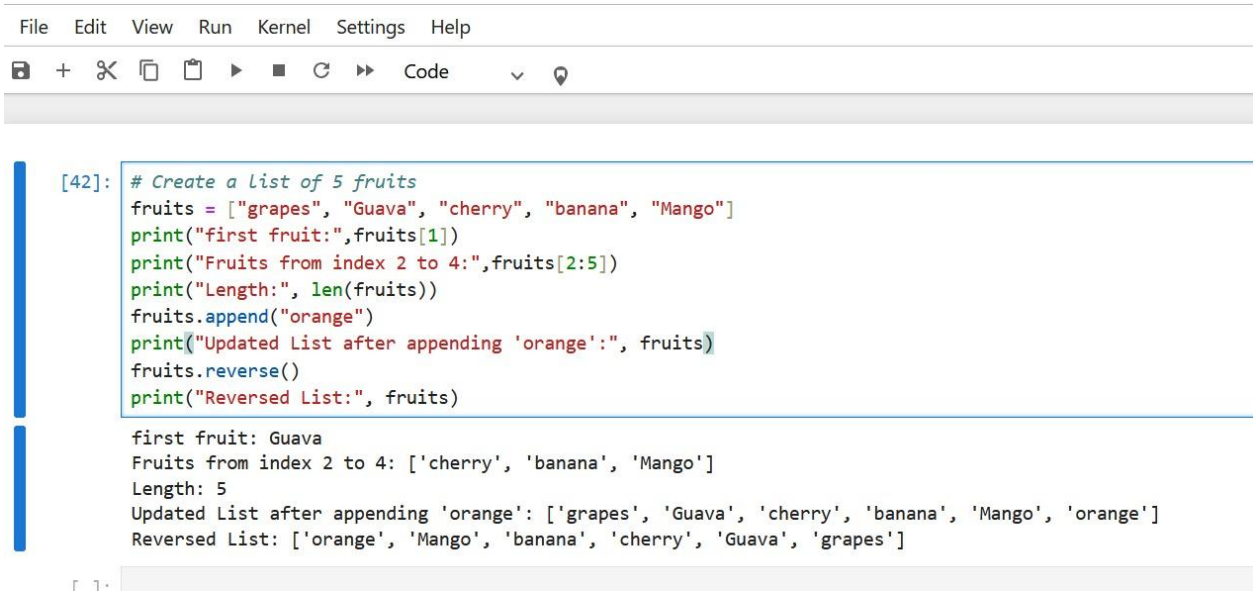
5. Write a Python program that takes an integer input and checks whether the number is positive, negative, or zero using conditional statements (if-else).



6. Write a Python program that takes a number as input and prints the multiplication table for that number (from 1 to 10).



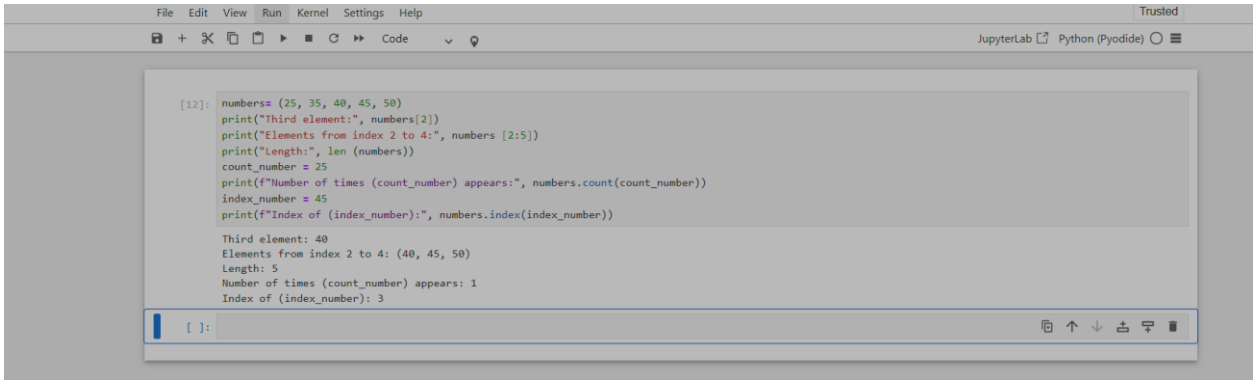
7. Create a Python list that contains the names of 5 different fruits. Perform the given operations on the list.



```
[42]: # Create a list of 5 fruits
fruits = ["grapes", "Guava", "cherry", "banana", "Mango"]
print("first fruit:",fruits[1])
print("Fruits from index 2 to 4:",fruits[2:5])
print("Length:", len(fruits))
fruits.append("orange")
print("Updated List after appending 'orange':", fruits)
fruits.reverse()
print("Reversed List:", fruits)
```

```
first fruit: Guava
Fruits from index 2 to 4: ['cherry', 'banana', 'Mango']
Length: 5
Updated List after appending 'orange': ['grapes', 'Guava', 'cherry', 'banana', 'Mango', 'orange']
Reversed List: ['orange', 'Mango', 'banana', 'cherry', 'Guava', 'grapes']
```

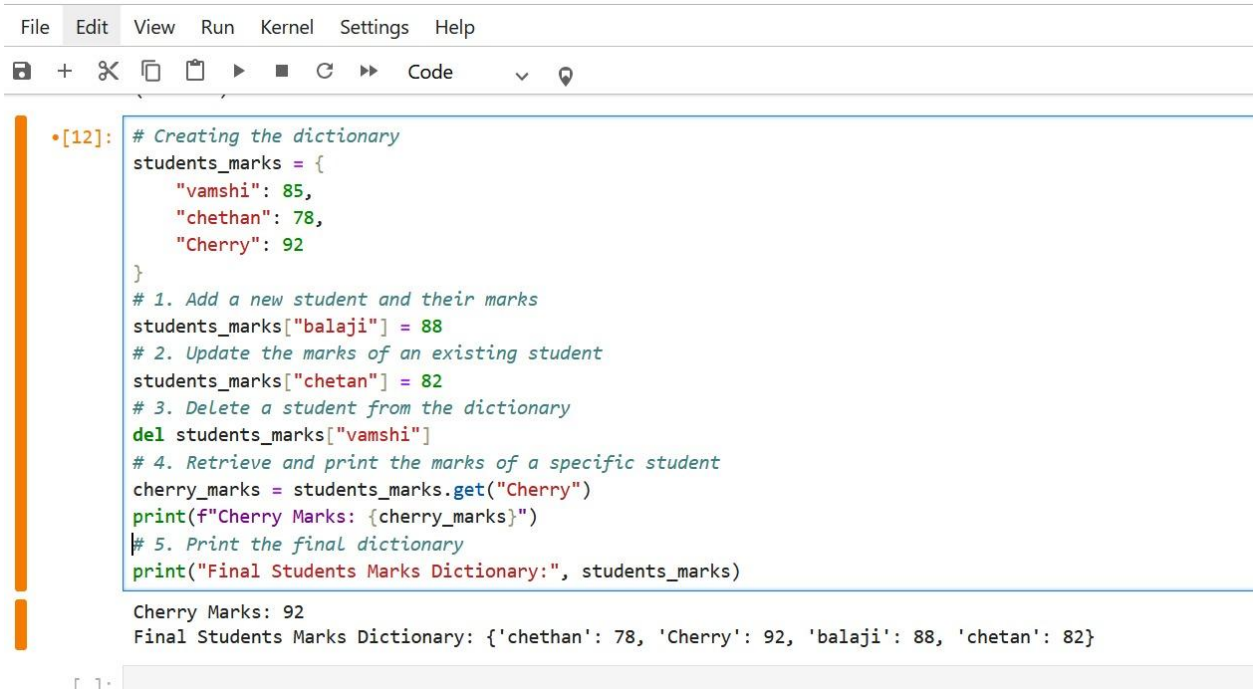
8. Write a Python program that creates a tuple containing 5 numbers. Perform the given operations on the tuple.



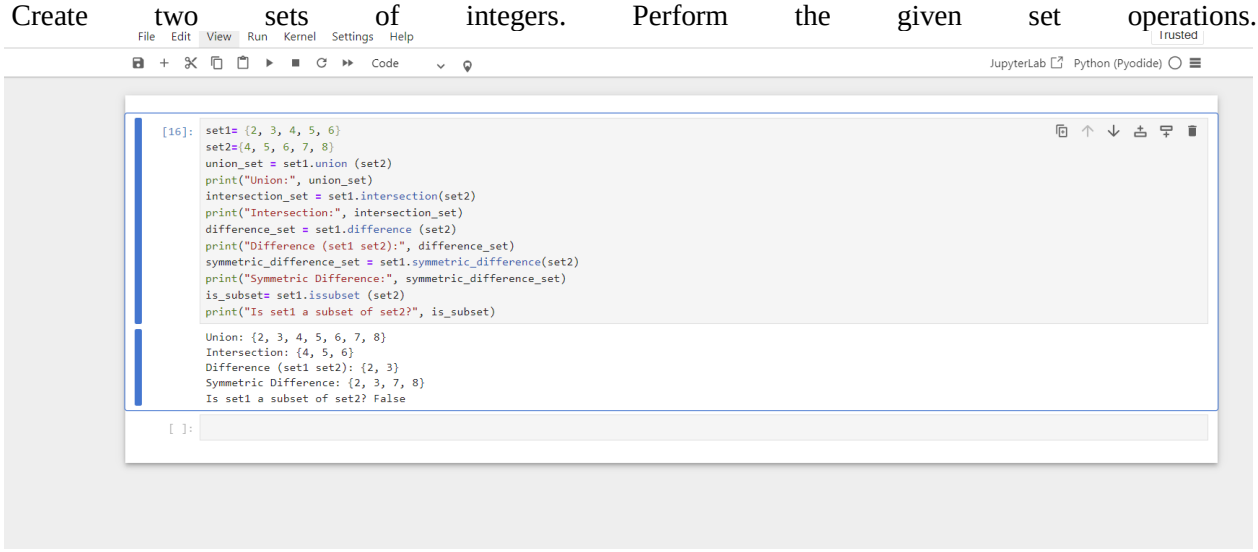
```
[12]: numbers= (25, 35, 40, 45, 50)
print("Third element:", numbers[2])
print("Elements from index 2 to 4:", numbers [2:5])
print("Length:", len (numbers))
count_number = 25
print(f"Number of times (count_number) appears:", numbers.count(count_number))
index_number = 45
print(f"Index of (index_number):", numbers.index(index_number))
```

```
Third element: 40
Elements from index 2 to 4: (40, 45, 50)
Length: 5
Number of times (count_number) appears: 1
Index of (index_number): 3
```

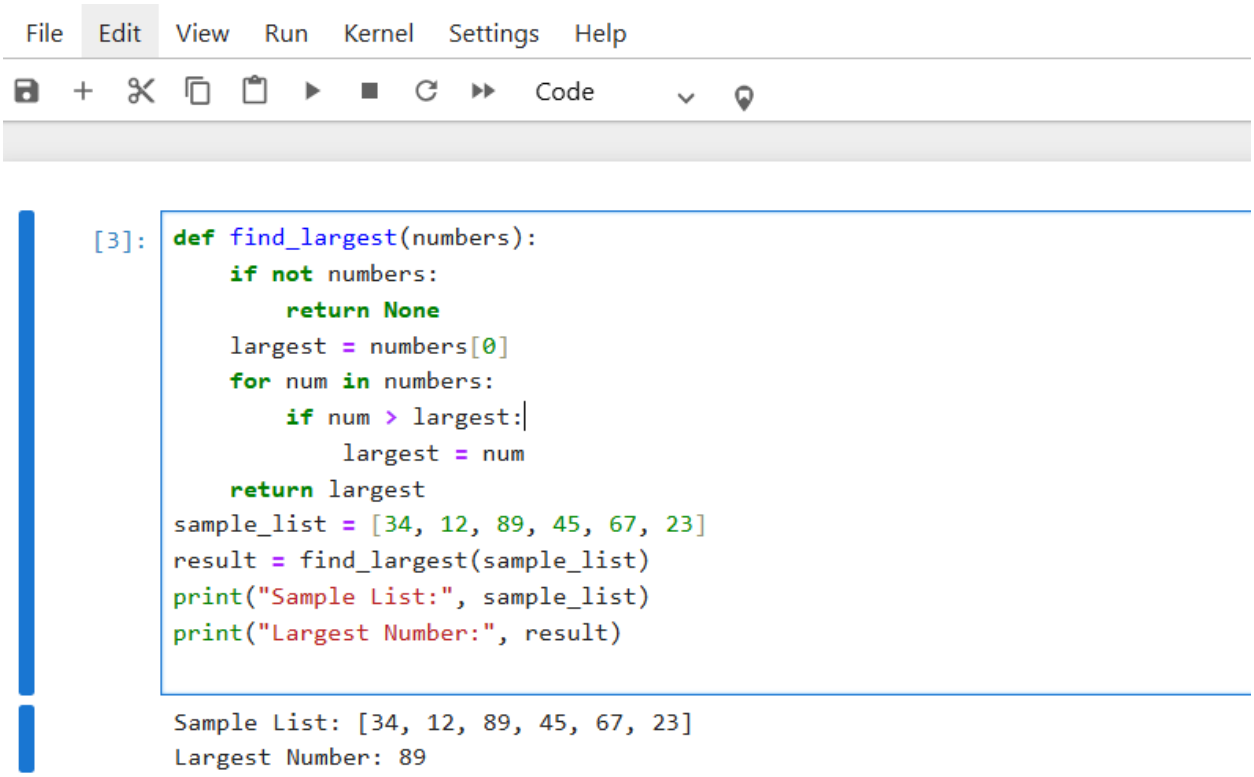
9. Create a dictionary that stores the names of 3 students as keys and their marks in mathematics as values. Performthegivenoperations.



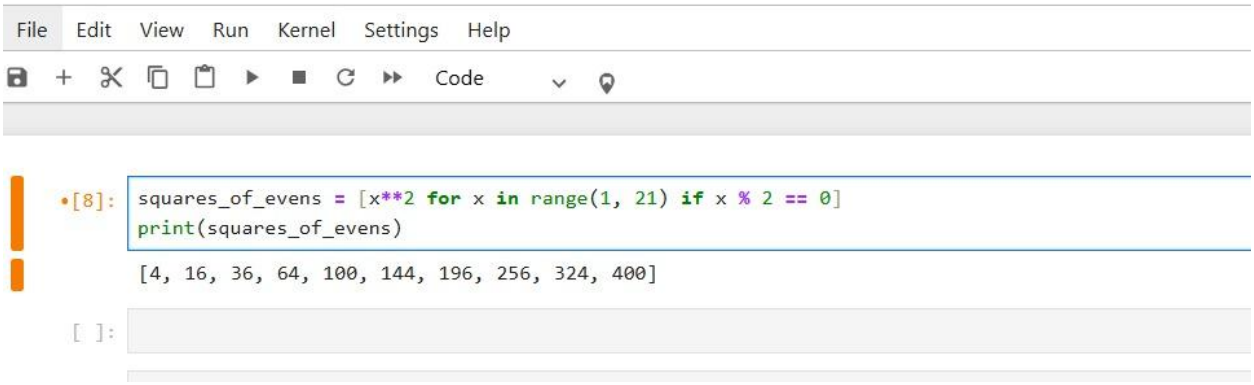
10. Create two sets of integers. Perform the given set operations.



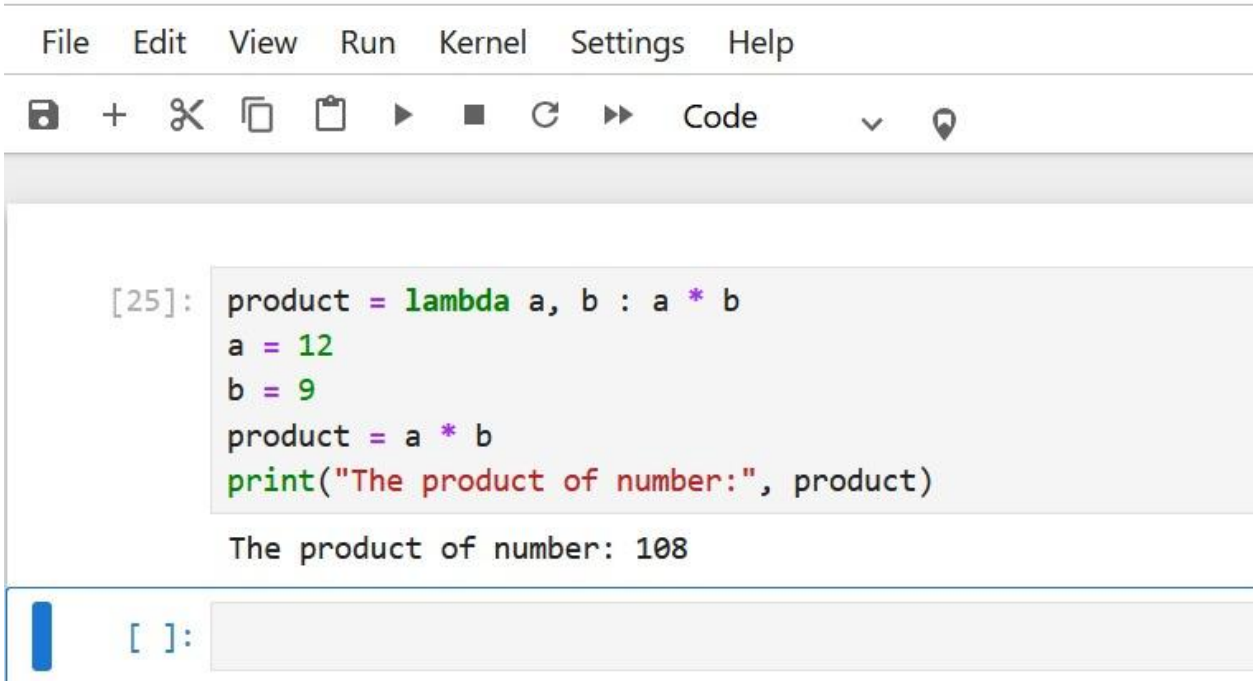
11. Write a Python function called find_largest() that takes a list of numbers as input and returns the largest number from the list. Test the function with a sample list.



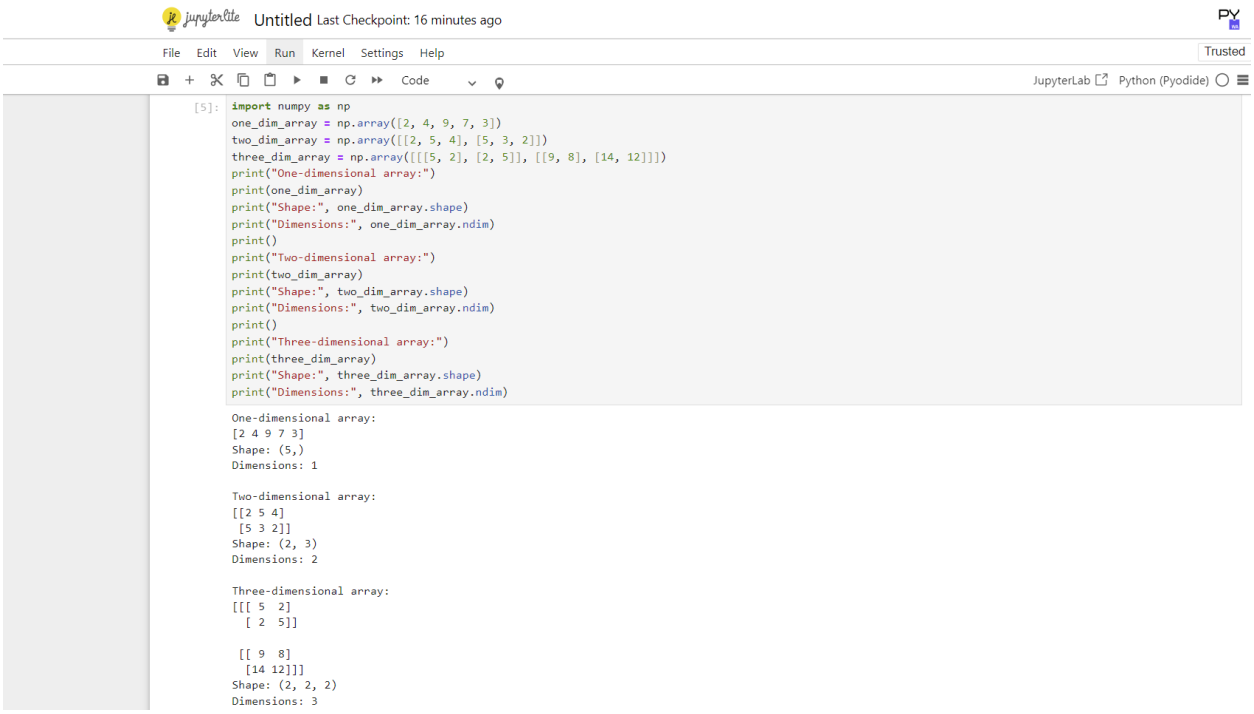
12. Use list comprehension to create a list of squares of all even numbers between 1 and 20.



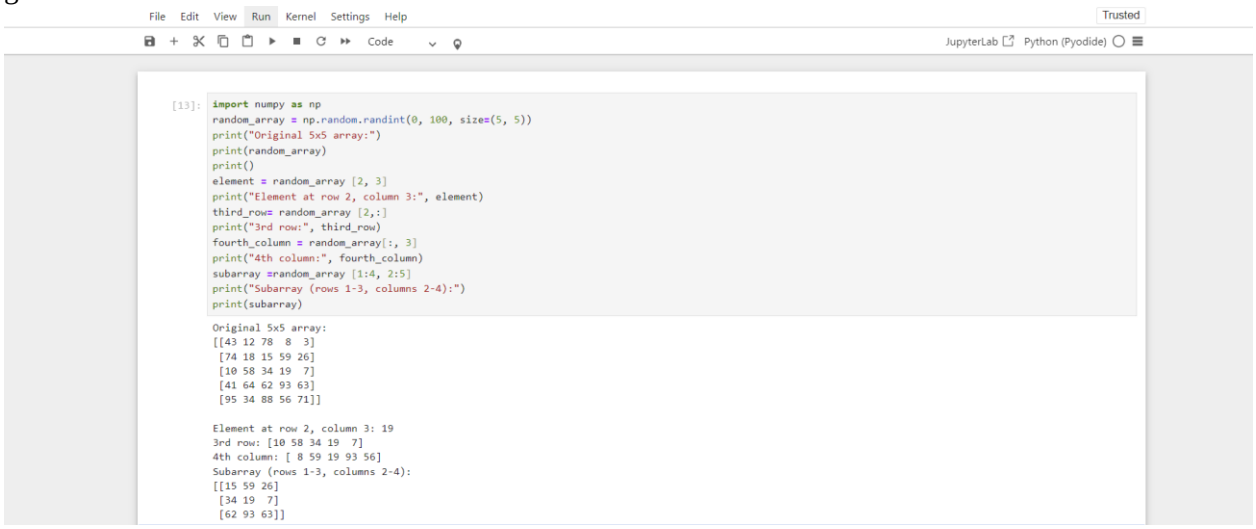
13. Write a Python script that uses a lambda function to calculate the product of two numbers provided by the user.



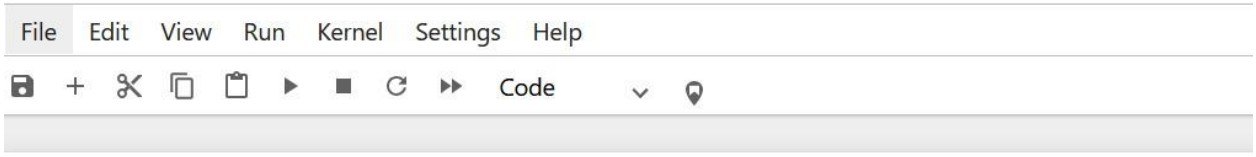
14. Write a Python program to create a one-dimensional, two-dimensional, and three-dimensional NumPy array. Print the shape and dimensions of each array.



15. Write a Python program to create a 5x5 NumPy array of random integers and Perform array indexing as given.

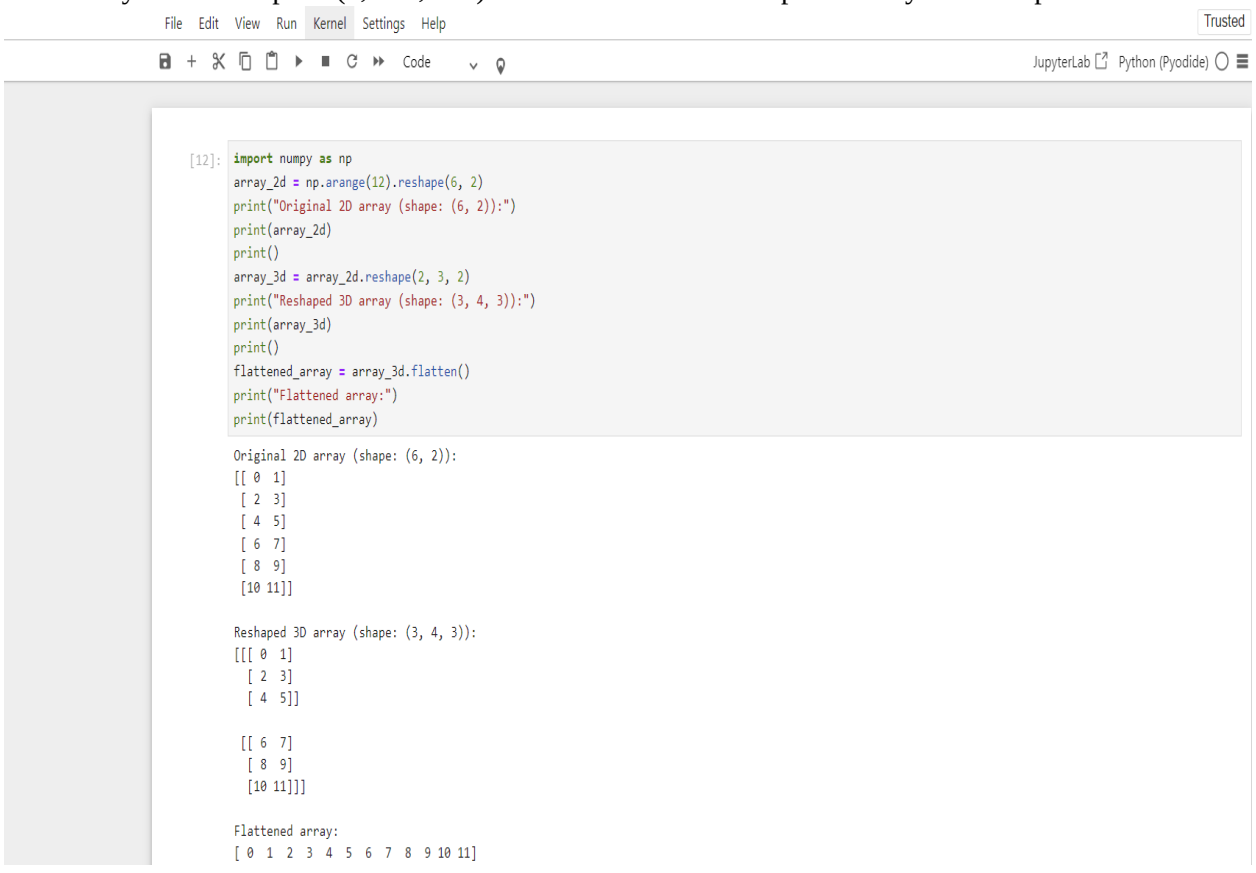


16. create a NumPy array of shape (4, 4) containing numbers from 1 to 16. Use slicing to extract for the given conditions




```
[5]: import numpy as np  # Importing NumPy
arr = np.arange(1, 17).reshape(4, 4)
print("Original 4x4 Array:\n", arr)
first_two_rows = arr[:2, :]
print("\nFirst two rows:\n", first_two_rows)
last_two_columns = arr[:, -2:]
print("\nLast two columns:\n", last_two_columns)
center_submatrix = arr[1:3, 1:3]
print("\n2×2 Center Sub-Matrix:\n", center_submatrix)
selected_rows = arr[[1, 3], :]
```

17. Write a Python program that creates a 2D array of shape (6, 2) using np.arange() and then reshapes it into a 3D array of shape (2, 3, 2). Flatten the reshaped array and print the result.




```
[12]: import numpy as np
array_2d = np.arange(12).reshape(6, 2)
print("Original 2D array (shape: (6, 2)):")
print(array_2d)
print()
array_3d = array_2d.reshape(2, 3, 2)
print("Reshaped 3D array (shape: (3, 4, 3)):")
print(array_3d)
print()
flattened_array = array_3d.flatten()
print("Flattened array:")
print(flattened_array)
```

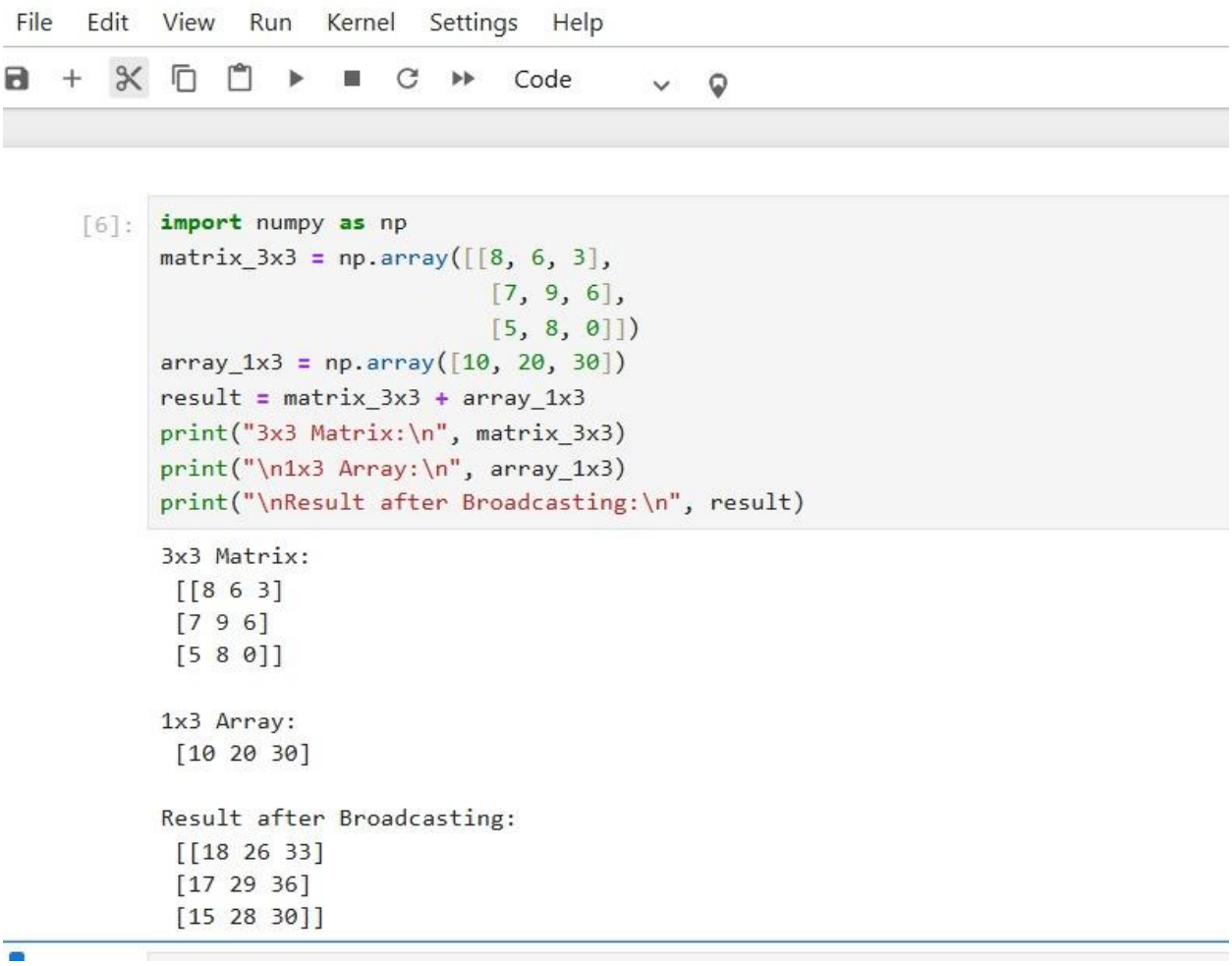
```
Original 2D array (shape: (6, 2)):
[[ 0  1]
 [ 2  3]
 [ 4  5]
 [ 6  7]
 [ 8  9]
 [10 11]]

Reshaped 3D array (shape: (3, 4, 3)):
[[[ 0  1]
  [ 2  3]
  [ 4  5]]

 [[ 6  7]
  [ 8  9]
  [10 11]]]

Flattened array:
[ 0  1  2  3  4  5  6  7  8  9 10 11]
```

18. Write a Python program to demonstrate broadcasting. Create an array of shape (3, 3) and add a one-dimensional array of shape (1, 3) to it using broadcasting.



19. Create two NumPy arrays of the same shape, A and B. Perform the following arithmetic operations:

    Element-wise addition.

    Element-wise subtraction.

    Element-wise multiplication.

    Element-wise division.

File Edit View Run Kernel Settings Help

Code

[30]:
```
import numpy as np
A = np.array ([ [3, 4, 5] ,[6, 7, 8] ])
B= np.array([[40, 50, 60], [70, 80, 90]])
print("Array A:")
print(A)
print()
print("Array B:")
print(B)
print()
addition = A + B
print("Element-wise addition (A + B): ")
print(addition)
print()
subtraction = A - B
print("Element-wise subtraction (A - B): ")
print(subtraction)
print()
multiplication = A  * B
print("Element-wise multiplication (A * B) :")
print(multiplication)
print()
division = A / B
print("Element-wise division (A / B): ")
print(division)
```

20. Create a Pandas DataFrame with the given Name and marks of 3 courses:

Add a new column named 'Total' that represents the sum of all the courses. Add 'Grade' based on the values of the 'Total'. Print the updated DataFrame with the new 'Total' and 'Grade' column.

File Edit View Run Kernel Settings Help

Code

[3]:
```
import pandas as pd # Importing Pandas
data = {
    "Name": ["vamshi", "chetan", "Cherry"],
    "Math": [85, 78, 92],
    "Science":[88, 74, 90],
    "English":[82, 80, 85]
}
df = pd.DataFrame(data)
df["Total"] = df["Math"] + df ["Science"] + df ["English"]
def assign_grade(total):
    if total >= 250:
        return "A"
    elif total >= 220:
        return "B"
    elif total >= 180:
        return "C"
    else:
        return "D"
df ["Grade"] = df["Total"].apply(assign_grade)
print(df)
```

```
     Name  Math  Science  English  Total Grade
0  vamshi    85       88       82    255     A
1  chetan    78       74       80    232     B
2  Cherry    92       90       85    267     A
```