Que1.Imagine you have a matrix represented as a numpy array, and your task is to find the sum of its diagonal elements using a loop.

Ans. Here's an example code snippet that calculates the sum of diagonal elements of a numpy array using a loop:

import numpy as np

# Define a square matrix

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

# Get the size of the matrix

size = matrix.shape[0]

# Initialize sum variable

diagonal\_sum = 0

# Loop through diagonal elements

for i in range(size):

diagonal\_sum += matrix[i, i]

print("Sum of diagonal elements:", diagonal\_sum)

``

In this code:

1. We define a square matrix (3x3) using numpy.

2. We get the size of the matrix using `matrix.shape[0]`.

3. We initialize a variable `diagonal\_sum` to store the sum.

4. We loop through the diagonal elements using `range(size)`.

5. Inside the loop, we add each diagonal element `matrix[i, i]` to `diagonal\_sum`.

6. Finally, we print the sum.

For the given matrix:

[[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

The diagonal elements are 1, 5, and 9. The sum is 1 + 5 + 9 = 15.

Alternatively, you can use numpy's `trace` function to calculate the sum of diagonal elements in a single line:

python

diagonal\_sum = np.trace(matrix)

```