- Write a Python Program to Add Two Matrices?

```python
In [1]:   1  def add_matrices(matrix1, matrix2):
          2      # Check if the matrices have the same dimensions
          3      if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
          4          print("Matrices must have the same dimensions for addition.")
          5          return None
          6
          7      result = []
          8      for i in range(len(matrix1)):
          9          row = []
         10          for j in range(len(matrix1[0])):
         11              row.append(matrix1[i][j] + matrix2[i][j])
         12          result.append(row)
         13
         14      return result
         15
         16  # Example matrices
         17  matrix_a = [
         18      [1, 2, 3],
         19      [4, 5, 6],
         20      [7, 8, 9]
         21  ]
         22
         23  matrix_b = [
         24      [9, 8, 7],
         25      [6, 5, 4],
         26      [3, 2, 1]
         27  ]
         28
         29  # Adding matrices
         30  result_matrix = add_matrices(matrix_a, matrix_b)
         31
         32  if result_matrix:
         33      print("Matrix A:")
         34      for row in matrix_a:
         35          print(row)
         36
         37      print("\nMatrix B:")
         38      for row in matrix_b:
         39          print(row)
         40
         41      print("\nSum of Matrix A and Matrix B:")
         42      for row in result_matrix:
         43          print(row)
         44
```

```
Matrix A:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Matrix B:
[9, 8, 7]
[6, 5, 4]
[3, 2, 1]

Sum of Matrix A and Matrix B:
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
```

- Write a Python Program to Multiply Two Matrices?

```
In [2]:  1  def multiply_matrices(matrix1, matrix2):
         2      # Check if matrices can be multiplied
         3      if len(matrix1[0]) != len(matrix2):
         4          print("Matrices cannot be multiplied due to incompatible dimensions.")
         5          return None
         6
         7      result = []
         8      for i in range(len(matrix1)):
         9          row = []
        10          for j in range(len(matrix2[0])):
        11              val = 0
        12              for k in range(len(matrix2)):
        13                  val += matrix1[i][k] * matrix2[k][j]
        14              row.append(val)
        15          result.append(row)
        16
        17      return result
        18
        19  # Example matrices
        20  matrix_a = [
        21      [1, 2, 3],
        22      [4, 5, 6],
        23      [7, 8, 9]
        24  ]
        25
        26  matrix_b = [
        27      [9, 8, 7],
        28      [6, 5, 4],
        29      [3, 2, 1]
        30  ]
        31
        32  # Multiplying matrices
        33  result_matrix = multiply_matrices(matrix_a, matrix_b)
        34
        35  if result_matrix:
        36      print("Matrix A:")
        37      for row in matrix_a:
        38          print(row)
        39
        40      print("\nMatrix B:")
        41      for row in matrix_b:
        42          print(row)
        43
        44      print("\nResult of Matrix A multiplied by Matrix B:")
        45      for row in result_matrix:
        46          print(row)
        47
```

```
Matrix A:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Matrix B:
[9, 8, 7]
[6, 5, 4]
[3, 2, 1]

Result of Matrix A multiplied by Matrix B:
[30, 24, 18]
[84, 69, 54]
[138, 114, 90]
```

- Write a Python Program to Transpose a Matrix?

```python
def transpose_matrix(matrix):
    # Calculate dimensions of the matrix
    rows = len(matrix)
    cols = len(matrix[0])

    # Create a new matrix to store the transpose
    transpose = []
    for j in range(cols):
        new_row = []
        for i in range(rows):
            new_row.append(matrix[i][j])
        transpose.append(new_row)

    return transpose

# Example matrix
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

# Transposing the matrix
transposed_matrix = transpose_matrix(matrix)

print("Original Matrix:")
for row in matrix:
    print(row)

print("\nTransposed Matrix:")
for row in transposed_matrix:
    print(row)
```

```
Original Matrix:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

Transposed Matrix:
[1, 4, 7]
[2, 5, 8]
[3, 6, 9]
```

- Write a Python Program to Sort Words in Alphabetic Order?

```python
In [4]:  1  def sort_words_alphabetically(sentence):
         2      # Split the sentence into words
         3      words = sentence.split()
         4
         5      # Sort the words in alphabetical order
         6      words.sort()
         7
         8      return words
         9
        10  # Example sentence
        11  input_sentence = "This is an example sentence to sort words alphabetically"
        12
        13  # Sorting words in the sentence
        14  sorted_words = sort_words_alphabetically(input_sentence)
        15
        16  print("Original Sentence:")
        17  print(input_sentence)
        18
        19  print("\nWords Sorted in Alphabetical Order:")
        20  print(sorted_words)
        21
```

```
Original Sentence:
This is an example sentence to sort words alphabetically

Words Sorted in Alphabetical Order:
['This', 'alphabetically', 'an', 'example', 'is', 'sentence', 'sort', 'to', 'words']
```

- Write a Python Program to Remove Punctuation From a String?

```python
In [5]:  1  import string
         2
         3  def remove_punctuation(input_string):
         4      # Define punctuation characters
         5      punctuation = string.punctuation
         6
         7      # Remove punctuation from the input string
         8      no_punctuation = input_string.translate(str.maketrans('', '', punctuation))
         9
        10      return no_punctuation
        11
        12  # Example string with punctuation
        13  input_string = "Hello! How are you? I'm doing fine, thank you!"
        14
        15  # Removing punctuation from the string
        16  result_string = remove_punctuation(input_string)
        17
        18  print("Original String:")
        19  print(input_string)
        20
        21  print("\nString with Punctuation Removed:")
        22  print(result_string)
        23
```

```
Original String:
Hello! How are you? I'm doing fine, thank you!

String with Punctuation Removed:
Hello How are you Im doing fine thank you
```

In [ ]: 1