

# Exercises

- 1) Write a Python program to find the maximum element in a matrix.
- 2) Write a Python program to find the minimum element in a matrix.
- 3) Write a Python program to find the trace of a matrix.
- 4) Write a Python program to find the determinant of a matrix.
- 5) Write a Python program to find the inverse of a matrix.
- 6) Write a Python program to find the rank of a matrix.
- 7) Write a Python program to find the eigenvalues and eigenvectors of a matrix.
- 8) Write a Python program to sort an array in ascending order.
- 9) Write a Python program to sort an array in descending order.
- 10) Write a Python program to sort a list of tuples based on the second element of each tuple.

# Exercises and solution

1) Find the maximum element in a matrix:

```
def find_max(matrix):  
    max_element = matrix[0][0]  
    for row in matrix:  
        for element in row:  
            if element > max_element:  
                max_element = element  
    return max_element
```

# Example usage:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print("Maximum element:", find_max(matrix)) # Output:  
Maximum element: 9
```

2) Find the minimum element in a matrix:

```
def find_min(matrix):  
    min_element = matrix[0][0]  
    for row in matrix:  
        for element in row:  
            if element < min_element:  
                min_element = element  
    return min_element
```

# Example usage:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print("Minimum element:", find_min(matrix)) # Output:  
Minimum element: 1
```

3) Find the trace of a matrix:

```
def find_trace(matrix):
    trace = 0
    for i in range(len(matrix)):
        trace += matrix[i][i]
    return trace

# Example usage:
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print("Trace:", find_trace(matrix)) # Output: Trace: 15
```

4) Find the determinant of a matrix:

```
def determinant(matrix):
    n = len(matrix)
    if n == 1:
        return matrix[0][0]
    elif n == 2:
        return matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]
    else:
        det = 0
        for j in range(n):
            minor = [[matrix[i][k] for k in range(n) if k != j] for i in
range(1, n)]
            det += matrix[0][j] * (-1) ** j * determinant(minor)
        return det

# Example usage:
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print("Determinant:", determinant(matrix)) # Output:
Determinant: 0
```

5) Find the inverse of a matrix:

```
import numpy as np

# creating a sample matrix
matrix = np.array([[1, 2], [3, 4]])

# finding the inverse of the matrix
inverse = np.linalg.inv(matrix)

print("Original Matrix:")
print(matrix)

print("Inverse of the Matrix:")
print(inverse)
```

6) Find the rank of a matrix:

```
import numpy as np

# creating a sample matrix
matrix = np.array([[1, 2], [3, 4]])

# finding the rank of the matrix
rank = np.linalg.matrix_rank(matrix)

print("Original Matrix:")
print(matrix)

print("Rank of the Matrix:")
print(rank)
```

7) Find the eigenvalues and eigenvectors of a matrix:

```
import numpy as np

# creating a sample matrix
```

```
matrix = np.array([[1, 2], [3, 4]])

# finding the eigenvalues and eigenvectors of the matrix
eigenvalues, eigenvectors = np.linalg.eig(matrix)

print("Original Matrix:")
print(matrix)

print("Eigenvalues:")
print(eigenvalues)

print("Eigenvectors:")
print(eigenvectors)
```

8) Sort an array in ascending order:

```
# creating a sample array
array = [3, 2, 1, 5, 4]

# sorting the array in ascending order
sorted_array = sorted(array)

print("Original Array:")
print(array)

print("Sorted Array in Ascending Order:")
print(sorted_array)
```

9) Sort an array in descending order:

```
# creating a sample array
array = [3, 2, 1, 5, 4]

# sorting the array in descending order
sorted_array = sorted(array, reverse=True)
```

```
print("Original Array:")  
print(array)
```

```
print("Sorted Array in Descending Order:")  
print(sorted_array)
```

- 10) Sort a list of tuples based on the second element of each tuple:

```
# create a list of tuples  
tuples_list = [("apple", 3), ("banana", 2), ("cherry", 4), ("date", 1)]  
  
# convert the list of tuples to an array  
arr = np.array(tuples_list)  
  
# sort the array based on the second column (index 1)  
arr_sorted = arr[arr[:, 1].argsort()]  
  
# convert the sorted array back to a list of tuples  
tuples_list_sorted = list(map(tuple, arr_sorted))  
  
print(tuples_list_sorted)
```