In [10]:
```
#Slicing of 1D array

import numpy as np
#creating single Diamentional array

num1=np.array([40,20,22,46,45,98,89,78,66,55,56])
print("Array is:",num1)

print("----------------------------------------")

#Slicing array from 2nd index to 9th index with gap of 2

sub_array=num1[2:8:2]
print("Slicing array:",sub_array)

#updating elements of subarray

sub_array[0]=123
sub_array[1]=345
sub_array[2]=567
print("Array after modification:",sub_array)
```

```
Array is: [40 20 22 46 45 98 89 78 66 55 56]
----------------------------------------
Slicing array: [22 45 89]
Array after modification: [123 345 567]
```

In [11]:
```
#Slicing of 2D array

import numpy as np
#creating two Diamentional array

num2=np.array([[10, 20, 30, 40, 50],
    [60, 70, 80, 90, 100],
    [110, 120, 130, 140, 150]])
print("2D Array is:",num2)

print("----------------------------------------")

# Slicing: Extracting rows from index 1 to 2 and columns from index 1 to 4 (step of

sub_array = num2[1:3, 1:5:2]
print("Sliced sub array:",sub_array)

#updating elements of subarray

sub_array[0,0]=123
sub_array[0,1]=345
sub_array[1,0]=567
sub_array[1,1] = 111
print("Sub Array after modification:",sub_array)

# Checking if the changes reflect in the original array (NumPy slicing creates a vi
```

```
print("Updated Original 2D Array:")
print(num2)
```

```
2D Array is: [[ 10  20  30  40  50]
 [ 60  70  80  90 100]
 [110 120 130 140 150]]
----------------------------------------
Sliced sub array: [[ 70  90]
 [120 140]]
Sub Array after modification: [[123 345]
 [567 111]]
Updated Original 2D Array:
[[ 10  20  30  40  50]
 [ 60 123  80 345 100]
 [110 567 130 111 150]]
```

In [12]:

```
import numpy as np

# Creating a one-dimensional NumPy array
number3 = np.array([40,23,45,67,12,31,54,65,70,91,57,61,89,65,92])
print("1D Array is:")
print(number3)

print("----------------------------------------")

# Creating a Boolean array with elements greater than 60 from original array
arra1 = number3 > 60
print("Boolean array having elements greater than 60 from original array:")
print(arra1)

print("----------------------------------------")

# Filtering the array based on condition
filtered_Array = number3[arra1]
print("Filtered array based on condition:")
print(filtered_Array)
```

```
1D Array is:
[40 23 45 67 12 31 54 65 70 91 57 61 89 65 92]
----------------------------------------
Boolean array having elements greater than 60 from original array:
[False False False  True False False False  True  True  True False  True
  True  True  True]
----------------------------------------
Filtered array based on condition:
[67 65 70 91 61 89 65 92]
```

In [1]:

```
import numpy as np

# Creating a 3D array (2 blocks × 3 rows × 4 columns)
num3D = np.array([
    [[10, 20, 30, 40],
     [50, 60, 70, 80],
     [90, 100, 110, 120]],

    [[130, 140, 150, 160],
     [170, 180, 190, 200],
```

```
     [210, 220, 230, 240]]
])

print("Original 3D Array:")
print(num3D)

print("---------------------------------------")

# Slicing: Extracting first block (depth index 0)
slice1 = num3D[0, :, :]  # Taking all rows and columns from the first block
print("Slicing first block:")
print(slice1)

print("---------------------------------------")

# Slicing: Extracting second column from all blocks
slice2 = num3D[:, :, 1]  # Taking all depth layers, all rows, but only column index
print("Slicing second column from all blocks:")
print(slice2)

print("---------------------------------------")

# Slicing: Extracting first two rows from both blocks
slice3 = num3D[:, 0:2, :]
print("Slicing first two rows from both blocks:")
print(slice3)

print("---------------------------------------")

# Slicing: Extracting elements from the last row of the last block
slice4 = num3D[1, -1, :]
print("Slicing last row of last block:")
print(slice4)
```

```
Original 3D Array:
[[[ 10  20  30  40]
  [ 50  60  70  80]
  [ 90 100 110 120]]

 [[130 140 150 160]
  [170 180 190 200]
  [210 220 230 240]]]
----------------------------------------
Slicing first block:
[[ 10  20  30  40]
 [ 50  60  70  80]
 [ 90 100 110 120]]
----------------------------------------
Slicing second column from all blocks:
[[ 20  60 100]
 [140 180 220]]
----------------------------------------
Slicing first two rows from both blocks:
[[[ 10  20  30  40]
  [ 50  60  70  80]]

 [[130 140 150 160]
  [170 180 190 200]]]
----------------------------------------
Slicing last row of last block:
[210 220 230 240]
```

In [3]:

```python
import numpy as np

# Creating a 2D array (4x4)
arr = np.array([
    [10, 20, 30, 40],
    [50, 60, 70, 80],
    [90, 100, 110, 120],
    [130, 140, 150, 160]
])

print("\nOriginal 2D Array:\n", arr)

#1. Get the Last Row (-1 Index)
last_row = arr[-1]
print("\nLast Row:", last_row)

#2. Get the Last Column (-1 Index)
last_column = arr[:, -1]
print("\nLast Column:", last_column)

#3. Get the Last Two Rows (-2:)
last_two_rows = arr[-2:]
print("\nLast Two Rows:\n", last_two_rows)

#4. Get the Last Two Columns (-2:)
last_two_columns = arr[:, -2:]
print("\nLast Two Columns:\n", last_two_columns)
```

```
#5. Get Bottom-Right 2x2 Submatrix
bottom_right = arr[-2:, -2:]
print("\nBottom Right 2x2 Submatrix:\n", bottom_right)
```

```
Original 2D Array:
 [[ 10  20  30  40]
 [ 50  60  70  80]
 [ 90 100 110 120]
 [130 140 150 160]]

Last Row: [130 140 150 160]

Last Column: [ 40  80 120 160]

Last Two Rows:
 [[ 90 100 110 120]
 [130 140 150 160]]

Last Two Columns:
 [[ 30  40]
 [ 70  80]
 [110 120]
 [150 160]]

Bottom Right 2x2 Submatrix:
 [[110 120]
 [150 160]]
```

In [ ]: