In [25]:

```
# Write a NumPy program to create a structured array from given student name, height, age and their data
import numpy as np
data_type = [('name', 'S15'), ('age', int), ('height', float)]
students_details = [('Manash', 16, 5.7), ('Jha', 19, 6.2),('Piyush', 19, 6.1), ('Mehta', 15, 5.9)]
students = np.array(students_details, dtype=data_type)
print("Original array:")
print(students)
print("Sort by height")
print(np.sort(students, order=['age','height']))
```

```
Original array:
[(b'Manash', 16, 5.7) (b'Jha', 19, 6.2) (b'Piyush', 19, 6.1)
 (b'Mehta', 15, 5.9)]
Sort by height
[(b'Mehta', 15, 5.9) (b'Manash', 16, 5.7) (b'Piyush', 19, 6.1)
 (b'Jha', 19, 6.2)]
```

In [23]:

```
# Write a NumPy program to get the indices of the sorted elements of a given array.
import numpy as np
array = np.array([10, 52, 62, 16, 16, 54, 453])
print(array)
indices = np.argsort(array)
print(indices)
```

```
[ 10  52  62  16  16  54 453]
[0 3 4 1 5 2 6]
```

In [27]:

```
# Write a NumPy program to test whether none of the elements of a given array is zero.
import numpy as np
arr = np.array([12,4,2,0,7,9,10,8])
if(np.all(arr)):
    print("No zero")
else:
    print("Zero")
```

```
It has zero
```

In [45]:

```
# Write a NumPy program to swap rows and columns of a given array in reverse order.
import numpy as np
nums = np.array([[12,4,7],
               [9,2,5],
               [13,9,2],
               [5, 0, 3]])
print("Original array:")
print(nums)
print("\nSwap rows and columns of the said array in reverse order:")
print(nums[::-1, ::-1])
```

```
Original array:
[[12  4  7]
 [ 9  2  5]
 [13  9  2]
 [ 5  0  3]]

Swap rows and columns of the said array in reverse order:
[ 7  4 12]
```

In [43]:

```
# Write a NumPy program to multiply two given arrays of same size element- by-element.
import numpy as np
n1 =  np.array([[12,4,7,3],
               [9,2,5,7],
               [13,9,2,8],
               [5, 0, 3,0]])
n2 =  np.array([[1,4,9,1],
               [8,5,7,3],
               [1,6,0,8],
               [1,8,8,9]])
n = np.multiply(n1, n1)
print(n)
```

```
[[144  16  49   9]
 [ 81   4  25  49]
 [169  81   4  64]
 [ 25   0   9   0]]
```

In [46]:

```
# Write a NumPy program to convert a given list into an array, then again convert it into a list. Check i
import numpy as np
a = [[7,8,9], [4,7,9]]
x = np.array(a)
a2 = x.tolist()
print(a == a2)
```

```
True
```

In [47]:

```
# Write a NumPy program to convert a list of numeric value into a one- dimensional NumPy array.
import numpy as np
l = [12.23, 13.32, 100, 36.32]
print("Original List:",l)
a = np.array(l)
print("One-dimensional NumPy array: ",a)
```

```
Original List: [12.23, 13.32, 100, 36.32]
One-dimensional NumPy array:  [ 12.23  13.32 100.    36.32]
```

In [ ]:

```
# Write a NumPy program to convert a list and tuple into arrays.
import numpy as np
my_list = [1, 2, 3, 4, 5, 6, 7, 8]
print("List to array: ")
print(np.asarray(my_list))
my_tuple = ([8, 4, 6], [1, 2, 3])
print("Tuple to array: ")
print(np.asarray(my_tuple))
```

In [48]:

```
# Write a NumPy program to get the unique elements of an array.
import numpy as np
arr = np.array([3, 3, 4, 5, 6, 5, 6, 4])
rslt = np.unique(arr)
print(rslt)
```

```
[3 4 5 6]
```

In [50]:

```
# Write a NumPy program to construct an array by repeating.
import numpy as np
a = [1, 2, 3, 4, 6]
print("Original array")
print(a)
print("Repeating 2 times")
x = np.tile(a, 2)
print(x)
```

```
Original array
[1, 2, 3, 4, 6]
Repeating 2 times
[1 2 3 4 6 1 2 3 4 6]
```