Que-1 Write a NumPy program to convert a list of numeric values into a one-dimensional NumPy array.

In [2]:
```
import numpy as np
```

In [3]:
```
x=[32,3.2,1.5,14,18]
```

In [4]:
```
print(x)
```

```
[32, 3.2, 1.5, 14, 18]
```

In [5]:
```
y=np.array(x)
```

In [6]:
```
print('one dimentional numpy array:',y)
```

```
one dimentional numpy array: [32.   3.2  1.5 14.  18. ]
```

Que-2 Write a NumPy program to create a 4x4 matrix with values ranging from 7 to 15.

In [7]:
```
import numpy as np
```

In [19]:
```
a=np.arange(7,15).reshape((4, 4))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In [19], line 1
----> 1 a=np.arange(7,15).reshape((4, 4))

ValueError: cannot reshape array of size 8 into shape (4,4)
```

In [20]:
```
print(a)
```

```
[ 7  8  9 10 11 12 13 14]
```

Que-3 Write Write a NumPy program to create a null vector of size 10 and update the sixth value to 11.a NumPy program to create a null vector of size 10 and update the sixth value to 11.

In [21]:
```
import numpy as np
```

In [22]:
```
n=np.zeros(10)
```

In [23]:
```
print(n)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

In [ ]:
```

```

In [37]:
```
n[6]=11
```

In [27]:
```
print(n)
```

```
[ 0.  0.  0.  0.  0.  0. 11.  0.  0.  0.]
```

Que-4 Write a NumPy program to reverse an array (first element becomes last).

In [28]:
```
import numpy as np
```

In [29]:
```
z=np.arange(6,18)
```

In [30]:
```
print(z)
```

```
[ 6  7  8  9 10 11 12 13 14 15 16 17]
```

In [32]:
```
z=z[::-1]
```

In [33]:
```
print(z)
```

```
[17 16 15 14 13 12 11 10  9  8  7  6]
```

Que-5 Write a NumPy program to create a 2d array with 1 on the border and 0 inside.

In [34]:
```
import numpy as np
```

In [35]:
```
k=np.ones((7,7))
```

In [36]:
```
print(k)
```

```
[[1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1.]]
```

In [38]:
```
k[1:-1,1:-1] = 0
```

In [39]:
```
print(k)
```

```
[[1. 1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1. 1. 1.]]
```

Que-6 Write a NumPy program to create a 8x8 matrix and fill it with a checkerboard pattern.

In [40]:
```
import numpy as np
```

In [41]:
```
l=np.ones((4,4))
```

In [42]:
```
print(l)
```

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

In [43]:
```
l = np.zeros((8,8),dtype=int)
```

In [44]:
```
print(l)
```

```
[[0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
```

In [45]:
```
l[1::2,::2] = 1
```

In [46]:
```
l
```

Out[46]:
```
array([[0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0],
       [1, 0, 1, 0, 1, 0, 1, 0]])
```

In [47]:
```
l[::2,1::2] = 1
```

In [48]:
```
print(l)
```

```
[[0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]]
```

Que-7 Write a NumPy program to append values to the end of an array.

In [49]:
```
import numpy as np
```

In [ ]:
```
program to convert the values of Centigrade degrees into Fahrenheit degrees and v
```

In [50]:
```
import numpy as np
```

Que-9 Write a NumPy program to find the number of elements of an array, length of one array element in bytes and total bytes consumed by the elements.

In [51]:
```
import numpy as np
```

In [52]:
```
p = np.array([4,4,6], dtype=np.float64)
```

In [53]:
```
print(p)
```

```
[4. 4. 6.]
```

In [54]:
```
p.size
```

Out[54]: 3

In [56]:
```
p.itemsize
```

Out[56]: 8

In [57]:
```
p.nbytes
```

Out[57]: 24

que-10 Write a NumPy program to test whether each element of a 1-D array is also present in a second array.

In [58]:
```
import numpy as np
```

In [59]:
```
a1 = np.array([12, 15, 20, 30, 50])
```

In [60]:
```
a2=[0,30]
```

In [61]:
```
print(np.in1d(a1, a2))
```

```
[False False False  True False]
```

```
Que-11 Write a NumPy program to create an array of ones and an array of zeros.
```

In [62]:
```
import numpy as np
```

In [63]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```