**1. #import numpy as np**

even\_array = np.arange(2, 101, 2)

reversed\_array = even\_array[::-1]

print(reversed\_array)

**OUTPUT**

[100 98 96 94 92 90 88 86 84 82 80 78 76 74 72 70 68 66

64 62 60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30

28 26 24 22 20 18 16 14 12 10 8 6 4 2]

**2. #import numpy as np**

row\_even = np.tile([1, 0], 4)

row\_odd = np.tile([0, 1], 4)

checkerboard = np.array([row\_even, row\_odd] \* 4)

print(checkerboard)

**OUTPUT**

[[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]

[0 1 0 1 0 1 0 1]]

**3. # Importing the NumPy library with an alias 'np'**

import numpy as np

# Creating a diagonal matrix with diagonal elements 1, 2, 3, 4, 5 using np.diag()

x = np. diag ([1, 2, 3, 4, 5])

print(x)

**OUTPUT**

[[1 0 0 0 0]

[0 2 0 0 0]

[0 0 3 0 0]

[0 0 0 4 0]

[0 0 0 0 5]]

**7.** A = np.array([

[9, 8, 7],

[6, 5, 4],

[3, 2, 1]])

a, b, c = A[0]

d, e, f = A[1]

g, h, i = A[2]

det = a\*(e\*i - f\*h) - b\*(d\*i - f\*g) + c\*(d\*h - e\*g)

print ("Determinant:", det)

**OUTPUT**

Determinant: 0

**5.** arr = np.array([

[1, 2],

[3, 4],

[1, 2]])

unique\_rows = np.unique(arr, axis=0)

print ("Unique rows:")

**OUTPUT**

Unique rows:

[[1 2]

[3 4]]