Create a null vector of size 10 but the fifth value which is 1.

In [3]:

```
import numpy as np

size = 10
position = int(input("Enter the position (0 to 9) for the value 1: "))

if 0 <= position < size:
    null_vector = np.zeros(size)
    null_vector[position] = 1
    print("Null vector:", null_vector)
else:
    print("Invalid position. Please enter a value between 0 and 9.")
```

```
Enter the position (0 to 9) for the value 1: 8
Null vector: [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
```

2 WAY

In [2]:

```
import numpy as np

# Get user input for the position of the non-zero element
position = int(input("Enter the position (0-9) of the non-zero element: "))

# Create a null vector of size 10
null_vector = np.zeros(10)

# Set the specified position to 1
null_vector[position] = 1

print("Resulting vector:", null_vector)
```

```
Enter the position (0-9) of the non-zero element: 5
Resulting vector: [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
```

2. Create a vector with values ranging from 10 to 49

In [5]:

```python
import numpy as np

# Create a vector with values ranging from 10 to 49
vector = np.arange(10, 50)

print("Vector:", vector)
```

```
Vector: [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49]
```

In [8]:

```python
import numpy as np

# Get user input for the range start and end
start = int(input("Enter the starting value: "))
end = int(input("Enter the ending value: "))

# Create a vector with values ranging from start to end (inclusive)
vector = np.arange(start, end + 1)

print("Resulting vector:", vector)
```

```
Enter the starting value: 10
Enter the ending value: 49
Resulting vector: [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 2
8 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49]
```

3. Create a 3x3 matrix with values ranging from 0 to 8

In [9]:

```python
import numpy as np

# Create a 3x3 matrix with values ranging from 0 to 8
matrix = np.arange(9).reshape(3, 3)

print("Resulting matrix:")
print(matrix)
```

```
Resulting matrix:
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

2 WAY

In [11]:

```
import numpy as np

# Get user input for the matrix values
values = [int(input(f"Enter value for row {i+1}, column {j+1}: ")) for i in range(3) for

# Reshape the values list into a 3x3 matrix
matrix = np.array(values).reshape(3, 3)

print("Resulting matrix:")
print(matrix)
```

```
Enter value for row 1, column 1: 0
Enter value for row 1, column 2: 1
Enter value for row 1, column 3: 2
Enter value for row 2, column 1: 3
Enter value for row 2, column 2: 4
Enter value for row 2, column 3: 5
Enter value for row 3, column 1: 6
Enter value for row 3, column 2: 7
Enter value for row 3, column 3: 8
Resulting matrix:
[[0 1 2]
 [3 4 5]
 [6 7 8]]
```

4. Find indices of non-zero elements from [1,2,0,0,4,0]

In [12]:

```
import numpy as np

arr = np.array([1, 2, 0, 0, 4, 0])
non_zero_indices = np.nonzero(arr)

print("Indices of non-zero elements:", non_zero_indices)
```

```
Indices of non-zero elements: (array([0, 1, 4], dtype=int64),)
```

2 WAY

In [13]:

```
import numpy as np

# Get user input for the list
input_list = [int(x) for x in input("Enter a list of numbers separated by spaces: ").spl

# Convert the list to a NumPy array
array = np.array(input_list)

# Find the indices of non-zero elements
non_zero_indices = np.nonzero(array)[0]

print("Indices of non-zero elements:", non_zero_indices)
```

```
Enter a list of numbers separated by spaces: 1 2 0 0 4 0
Indices of non-zero elements: [0 1 4]
```

5. Create a 10x10 array with random values and find the minimum and maximum values.

In [15]:

```
import numpy as np

# Get user input for the range of random values
min_value = int(input("Enter the minimum value: "))
max_value = int(input("Enter the maximum value: "))

# Create a 10x10 array with random values in the specified range
array = np.random.randint(min_value, max_value + 1, size=(10, 10))

# Find the minimum and maximum values in the array
min_val = np.min(array)
max_val = np.max(array)

print("Generated array:")
print(array)
print("Minimum value:", min_val)
print("Maximum value:", max_val)
```

```
Enter the minimum value: 1
Enter the maximum value: 100
Generated array:
[[  1  66  76  88  87  16  94  65  55  25]
 [ 55  63  84  63  66  46  47  90  65   2]
 [ 82  91  42  14  16  21   4  38  94  64]
 [ 81  37  95   3  57  69  98  42  37  66]
 [ 53  29  79  40  98  88  47  71  17  55]
 [ 26  66  81  75  74  16  44   1   3  89]
 [ 95  50   9  11   3  21  17  57  54  61]
 [ 25  88  22  64  17  51  34  66   4   7]
 [ 51  19  94  10  63  76  59  93  94  71]
 [100  11  34  97  64  68  68  47  79  24]]
Minimum value: 1
Maximum value: 100
```

2 way

In [14]:

```
import numpy as np

# Create a 10x10 array with random values between 0 and 1
array = np.random.rand(10, 10)

# Find the minimum and maximum values in the array
min_value = np.min(array)
max_value = np.max(array)

print("Array:")
print(array)
print("Minimum value:", min_value)
print("Maximum value:", max_value)
```

```
Array:
[[0.37311204 0.74601592 0.30532929 0.20294863 0.75974288 0.23142032
  0.23820393 0.40648697 0.26969152 0.38521401]
 [0.51288606 0.66089498 0.03733771 0.33839631 0.047805   0.55440514
  0.80023272 0.41776224 0.20017186 0.94778123]
 [0.07518326 0.63411411 0.99241826 0.67795836 0.9061897  0.43804765
  0.11538449 0.92221126 0.4068538  0.70868381]
 [0.19383651 0.44556379 0.28295997 0.96523548 0.117181   0.06366185
  0.20371744 0.72809334 0.78693996 0.25860525]
 [0.96229678 0.86360225 0.09329241 0.05874321 0.08817509 0.80650532
  0.36209751 0.77059182 0.42277887 0.93417511]
 [0.305923   0.94239528 0.97255677 0.15168851 0.26264888 0.72006024
  0.52279932 0.57342643 0.22415448 0.91929125]
 [0.01962505 0.19318263 0.88594185 0.54659062 0.04299913 0.58185933
  0.42461841 0.1175759  0.73404937 0.37630964]
 [0.02015043 0.06421294 0.0500493  0.46626509 0.79024006 0.02215276
  0.30077557 0.18174099 0.69900155 0.13715084]
 [0.93967898 0.80707431 0.50432886 0.62731209 0.03819608 0.64926832
  0.85703243 0.35135597 0.72525928 0.52543558]
 [0.08092935 0.72614852 0.70865343 0.02404323 0.99316871 0.7648281
  0.21815805 0.16279806 0.19235555 0.35234071]]
Minimum value: 0.019625054623561344
Maximum value: 0.9931687102006814
```

6. Create a random vector of size 30 and find the mean value.

In [16]:

```
import numpy as np

# Create a random vector of size 30
random_vector = np.random.random(30)

# Calculate the mean value of the random vector
mean_value = np.mean(random_vector)

print("Random vector:", random_vector)
print("Mean value:", mean_value)
```

```
Random vector: [0.72163704 0.45466521 0.72449463 0.49843971 0.20867727 0.7
7245042
 0.45949729 0.69474473 0.81134253 0.03910824 0.68096732 0.16172196
 0.41497481 0.01112145 0.25632871 0.55006636 0.40392978 0.9082125
 0.79957307 0.09782312 0.17618839 0.82281661 0.95148413 0.480609
 0.93851484 0.06894554 0.64110502 0.7548358  0.62352575 0.02464301]
Mean value: 0.5050814741774604
```

In [17]:

```
import numpy as np

# Get user input for the range of random values
min_value = int(input("Enter the minimum value: "))
max_value = int(input("Enter the maximum value: "))

# Create a random vector of size 30 with values in the specified range
vector = np.random.randint(min_value, max_value + 1, size=30)

# Calculate the mean value of the vector
mean_value = np.mean(vector)

print("Generated random vector:")
print(vector)
print("Mean value:", mean_value)
```

```
Enter the minimum value: 1
Enter the maximum value: 50
Generated random vector:
[44  9 38 37 37 35  7 21 42 38 31 35 38 47 32 30 26 17  8 44 29 12 45 45
 27 46 49 30 23  7]
Mean value: 30.966666666666665
```

In [ ]: