1.Write a numpy program to test equal, not equal , greater equal, greater and less test of all the elements of two given arrays.

```
!pip install numpy
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.22.4)
```

```
import numpy as np

x1=np.array([1,2,3,4,5,6])
x2=np.array([7,8,9,10,11,12])

print("Array 1 :" ,x1)
print("Array 2 :" ,x2)

#equal
print("equal case")
r=np.equal(x1,x2)
print(r)

#not_equal
print("not_equal case")
r1=np.not_equal(x1,x2)
print(r1)

#less_equal
print("less_equal case")
r2=np.less_equal(x1,x2)
print(r2)

#greater_equal
print("greater_equal case")
r3=np.greater_equal(x1,x2)
print(r3)

#less
print("less case")
r4=np.less(x1,x2)
print(r4)
```

```
Array 1 : [1 2 3 4 5 6]
Array 2 : [ 7  8  9 10 11 12]
equal case
[False False False False False False]
not_equal case
[ True  True  True  True  True  True]
less_equal case
[ True  True  True  True  True  True]
greater_equal case
[False False False False False False]
less case
[ True  True  True  True  True  True]
```

2.Write a Numpy program to create an array with the values 10,20,30,40 and determine the size of the memory occupied by the array

```
import numpy as np

arr=np.array([10,20,30,40])
print(arr)
print("Size of the memory occupied by the array:")
print("%d bytes" % (arr.size * arr.itemsize))
```

```
[10 20 30 40]
Size of the memory occupied by the array:
32 bytes
```

3.Write a Numpy program to create :

a. An array of the integers from 1 to 100.

```
import numpy as np
arr=np.arange(1,101)
```

```
print(arr)
```

```
[  1   2   3   4   5   6   7   8   9  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  50  51  52  53  54
  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
  91  92  93  94  95  96  97  98  99 100]
```

b.An array of all the even integers from 1 to 100.

```
import numpy as np
arr=np.arange(2,101,2)
print(arr)
```

```
[  2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36
  38  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72
  74  76  78  80  82  84  86  88  90  92  94  96  98 100]
```

c.An array of all the odd integers from 1 to 100.

```
import numpy as np
arr=np.arange(1,101,2)
print(arr)
```

```
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95
 97 99]
```

4.Write a Numpy program to compute sum of all elements, sum of each column and sum of each row of a given array

```
import numpy as np
arr=np.array([[1,2],[3,4]])

print("Original array")
print(arr)

print("Sum of all elements")
print(np.sum(arr))

print("Sum of rows")
print(np.sum(arr, axis=1))

print("Sum of columns")
print(np.sum(arr, axis=0))
```

```
Original array
[[1 2]
 [3 4]]
Sum of all elements
10
Sum of rows
[3 7]
Sum of columns
[4 6]
```

5.Write a numpy program to get the floor, ceiling and truncated values of the elements of a numpy array.

```
import numpy as np
arr=np.array([-2.0, -1.6, -1.0, 0, 1.0, 2.6, 3.1])
print(arr)
print("floor values :: ")
print(np.floor(arr))
print("ceil values :: ")
print(np.ceil(arr))
print("trunc values :: ")
print(np.trunc(arr))
```

```
[-2.  -1.6 -1.   0.   1.   2.6  3.1]
floor values ::
[-2. -2. -1.  0.  1.  2.  3.]
ceil values ::
[-2. -1. -1.  0.  1.  3.  4.]
```

```
trunc values ::
[-2. -1. -1.  0.  1.  2.  3.]
```

✓ 0s completed at 4:40 PM