

```
In [27]: import matplotlib.image as bnm  
modi1 = bnm.imread("modi1.jpg.jpeg")  
modi1
```

```
Out[27]: array([[[210, 196, 231],  
                 [211, 197, 232],  
                 [211, 197, 233],  
                 ...,  
                 [223, 210, 236],  
                 [223, 210, 236],  
                 [223, 210, 236]],  
  
                [[211, 197, 232],  
                 [211, 197, 232],  
                 [211, 197, 233],  
                 ...,  
                 [223, 210, 236],  
                 [223, 210, 236],  
                 [223, 210, 236]],  
  
                [[212, 198, 233],  
                 [212, 198, 233],  
                 [212, 198, 234],  
                 ...,  
                 [223, 210, 236],  
                 [223, 210, 236],  
                 [223, 210, 236]],  
  
                ...,  
  
                [[ 83, 106,  80],  
                 [ 83, 106,  80],  
                 [ 83, 106,  80],  
                 ...,  
                 [224, 211, 229],  
                 [224, 211, 228],  
                 [223, 210, 227]],  
  
                [[ 83, 106,  80],  
                 [ 83, 106,  80],  
                 [ 83, 106,  80],  
                 ...,  
                 [224, 211, 229],  
                 [224, 211, 228],  
                 [223, 210, 227]],
```

```
[[ 83, 106, 80],  
 [ 83, 106, 80],  
 [ 83, 106, 80],  
 ...,  
 [224, 211, 229],  
 [224, 211, 228],  
 [223, 210, 227]]], dtype=uint8)
```

these are the values of the pixels

pixel range 0-255

```
In [34]: import matplotlib.pyplot as plt  
plt.imshow(modi1)
```

```
Out[34]: <matplotlib.image.AxesImage at 0x1f1569ca740>
```



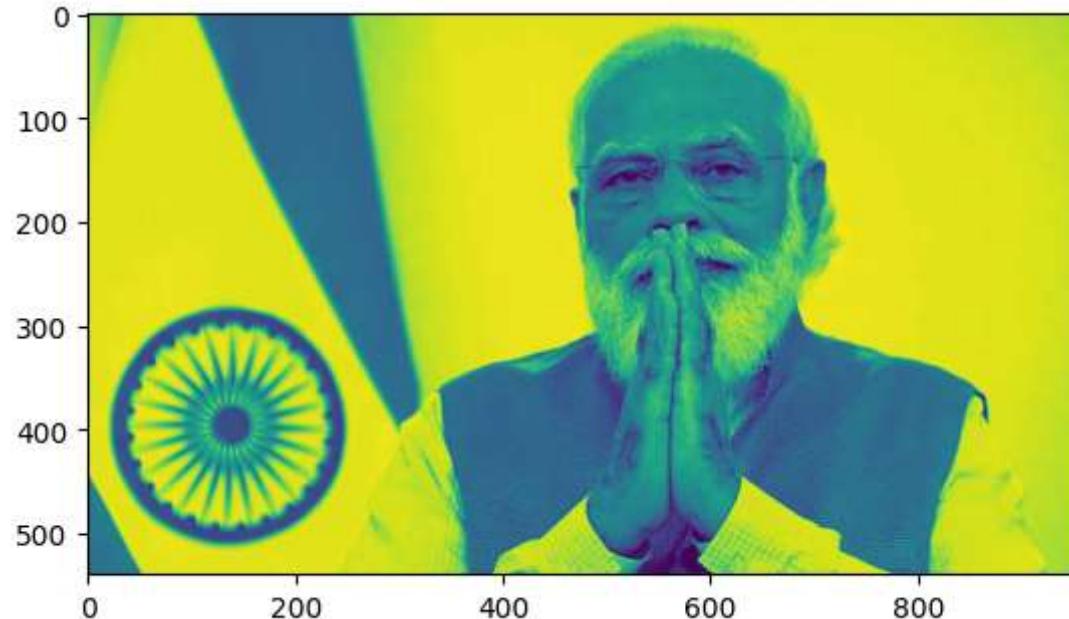
```
In [35]: modi1.shape #3 indicates the color rbg #540*960 IMAGE SIZE
```

```
Out[35]: (540, 960, 3)
```

```
In [32]: import cv2  
modified = cv2.cvtColor(modi1, cv2.COLOR_BGR2GRAY)
```

```
In [33]: plt.imshow(modified)
```

```
Out[33]: <matplotlib.image.AxesImage at 0x1f155de4b80>
```



```
In [ ]: #ENCRYPTION OF IMAGE
```

```
In [39]: prime =[]  
for num in range(2,255): #for 2 to 255-1  
    if all(num%i!=0 for i in range (2,num)):  
        prime.append(num)  
prime
```

```
Out[39]: [2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,
```

```
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251]
```

```
In [40]: len(modi1) #total number pf rows
```

```
Out[40]: 540
```

```
In [42]: prime =prime*len(modi1)  
prime #increases the occurence of prime values by 540 times
```

```
Out[42]: [2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,
```

181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,

113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,

61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,

13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,

223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,

151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,

89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,

37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,

251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,

181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,

113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,

61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,

13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,

223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,

151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,

89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,

37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,

251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,

181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,

113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,

61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,  
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,

13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
109,  
113,  
127,  
131,  
137,  
139,  
149,  
151,  
157,  
163,  
167,  
173,  
179,  
181,  
191,  
193,  
197,  
199,  
211,

```
223,  
227,  
229,  
233,  
239,  
241,  
251,  
2,  
3,  
5,  
7,  
11,  
13,  
17,  
19,  
23,  
29,  
31,  
37,  
41,  
43,  
47,  
53,  
59,  
61,  
67,  
71,  
73,  
79,  
83,  
89,  
97,  
101,  
103,  
107,  
...]
```

```
In [47]: #for shuffling prime values  
import numpy as np  
import random  
np.random.shuffle(prime)  
prime
```



```
Out[47]: [83,
 151,
 139,
 47,
 23,
 43,
 2,
 239,
 43,
 181,
 67,
 97,
 199,
 167,
 131,
 109,
 37,
 199,
 13,
 163,
 241,
 19,
 191,
 89,
 37,
 251,
 2,
 13,
 2,
 199,
 193,
 71,
 19,
 29,
 157,
 53,
 163,
 59,
 47,
 17,
 229,
```

197,  
149,  
241,  
239,  
127,  
139,  
173,  
73,  
3,  
149,  
251,  
103,  
107,  
197,  
241,  
53,  
163,  
53,  
89,  
3,  
127,  
11,  
179,  
97,  
97,  
151,  
151,  
61,  
23,  
193,  
89,  
239,  
61,  
137,  
227,  
19,  
223,  
31,  
71,  
97,  
191,  
101,

113,  
173,  
239,  
211,  
157,  
233,  
149,  
241,  
17,  
163,  
157,  
79,  
179,  
229,  
191,  
41,  
31,  
191,  
17,  
43,  
239,  
2,  
73,  
167,  
167,  
97,  
229,  
139,  
23,  
31,  
241,  
113,  
227,  
211,  
41,  
97,  
223,  
149,  
181,  
181,  
223,  
163,

61,  
113,  
13,  
227,  
53,  
241,  
157,  
67,  
47,  
11,  
233,  
79,  
251,  
2,  
181,  
223,  
241,  
167,  
47,  
7,  
89,  
53,  
131,  
137,  
167,  
227,  
43,  
79,  
31,  
173,  
227,  
7,  
7,  
211,  
41,  
103,  
137,  
103,  
89,  
251,  
137,  
157,

191,  
179,  
241,  
37,  
151,  
211,  
139,  
223,  
157,  
37,  
113,  
179,  
67,  
13,  
89,  
211,  
79,  
251,  
11,  
5,  
17,  
109,  
53,  
167,  
41,  
5,  
107,  
3,  
199,  
19,  
7,  
233,  
47,  
197,  
199,  
241,  
149,  
71,  
181,  
73,  
181,  
223,

103,  
61,  
199,  
29,  
89,  
233,  
71,  
97,  
5,  
7,  
127,  
163,  
13,  
13,  
7,  
23,  
61,  
5,  
2,  
23,  
151,  
17,  
191,  
239,  
151,  
181,  
43,  
181,  
193,  
47,  
61,  
43,  
2,  
211,  
67,  
37,  
53,  
251,  
149,  
107,  
2,  
109,

5,  
97,  
107,  
127,  
131,  
157,  
97,  
229,  
109,  
53,  
227,  
71,  
67,  
29,  
197,  
229,  
113,  
73,  
3,  
241,  
239,  
41,  
167,  
73,  
227,  
7,  
83,  
173,  
137,  
233,  
2,  
173,  
151,  
43,  
19,  
11,  
53,  
71,  
83,  
233,  
137,  
31,

127,  
3,  
239,  
131,  
53,  
3,  
73,  
17,  
17,  
13,  
73,  
199,  
109,  
181,  
137,  
173,  
191,  
37,  
113,  
13,  
197,  
157,  
139,  
151,  
103,  
167,  
181,  
149,  
41,  
229,  
41,  
193,  
239,  
7,  
13,  
151,  
83,  
43,  
151,  
79,  
151,  
83,

179,  
59,  
211,  
59,  
17,  
223,  
67,  
151,  
103,  
47,  
113,  
157,  
137,  
83,  
149,  
79,  
37,  
97,  
239,  
199,  
5,  
101,  
17,  
31,  
239,  
31,  
73,  
113,  
113,  
79,  
113,  
29,  
83,  
101,  
163,  
193,  
179,  
223,  
191,  
191,  
223,  
67,

193,  
139,  
127,  
7,  
79,  
43,  
239,  
83,  
197,  
71,  
79,  
251,  
7,  
193,  
73,  
23,  
179,  
83,  
73,  
89,  
31,  
179,  
233,  
101,  
127,  
107,  
191,  
241,  
109,  
89,  
37,  
157,  
139,  
139,  
113,  
3,  
193,  
23,  
73,  
199,  
71,  
19,

43,  
67,  
163,  
229,  
61,  
2,  
11,  
107,  
149,  
107,  
137,  
197,  
11,  
181,  
19,  
167,  
173,  
67,  
223,  
137,  
241,  
181,  
37,  
211,  
157,  
181,  
223,  
59,  
67,  
127,  
59,  
59,  
5,  
197,  
29,  
89,  
151,  
43,  
3,  
173,  
103,  
61,

167,  
199,  
109,  
101,  
29,  
97,  
101,  
167,  
137,  
229,  
229,  
157,  
43,  
23,  
43,  
149,  
19,  
157,  
47,  
101,  
109,  
197,  
47,  
7,  
131,  
97,  
229,  
67,  
193,  
239,  
89,  
211,  
193,  
139,  
7,  
131,  
97,  
13,  
83,  
223,  
23,  
181,

181,  
19,  
149,  
13,  
227,  
61,  
19,  
251,  
2,  
227,  
179,  
151,  
101,  
103,  
227,  
73,  
193,  
167,  
151,  
61,  
233,  
229,  
71,  
139,  
3,  
43,  
37,  
149,  
199,  
61,  
37,  
163,  
241,  
179,  
137,  
67,  
173,  
73,  
37,  
31,  
139,  
41,

113,  
241,  
151,  
97,  
31,  
89,  
71,  
13,  
109,  
227,  
197,  
139,  
73,  
233,  
23,  
41,  
197,  
239,  
109,  
73,  
43,  
11,  
107,  
2,  
79,  
3,  
127,  
193,  
101,  
7,  
59,  
5,  
43,  
71,  
13,  
41,  
229,  
23,  
211,  
3,  
181,  
113,

127,  
67,  
223,  
191,  
181,  
109,  
181,  
193,  
163,  
47,  
251,  
227,  
29,  
211,  
113,  
103,  
59,  
41,  
193,  
211,  
197,  
229,  
43,  
149,  
71,  
71,  
5,  
73,  
163,  
89,  
61,  
13,  
53,  
41,  
241,  
113,  
83,  
197,  
89,  
67,  
227,  
83,

89,  
67,  
5,  
97,  
251,  
79,  
181,  
223,  
131,  
3,  
223,  
29,  
127,  
2,  
71,  
223,  
17,  
7,  
239,  
17,  
11,  
61,  
251,  
179,  
113,  
67,  
89,  
167,  
139,  
101,  
79,  
41,  
167,  
109,  
23,  
173,  
29,  
191,  
79,  
107,  
127,  
107,

41,  
17,  
199,  
211,  
151,  
97,  
193,  
19,  
59,  
59,  
31,  
11,  
179,  
199,  
89,  
107,  
131,  
61,  
157,  
11,  
7,  
139,  
71,  
67,  
193,  
17,  
191,  
239,  
97,  
37,  
157,  
107,  
101,  
23,  
103,  
29,  
199,  
101,  
163,  
251,  
61,  
163,

29,  
151,  
31,  
199,  
251,  
79,  
47,  
7,  
79,  
5,  
97,  
137,  
191,  
167,  
211,  
197,  
151,  
5,  
53,  
19,  
2,  
89,  
113,  
167,  
157,  
53,  
151,  
11,  
241,  
3,  
59,  
59,  
197,  
37,  
83,  
157,  
211,  
13,  
199,  
233,  
223,  
109,

101,  
191,  
37,  
193,  
11,  
2,  
109,  
29,  
113,  
11,  
131,  
151,  
173,  
151,  
199,  
101,  
193,  
61,  
2,  
179,  
97,  
31,  
89,  
197,  
227,  
3,  
233,  
7,  
73,  
29,  
113,  
29,  
41,  
193,  
97,  
5,  
47,  
199,  
5,  
157,  
11,  
79,

191,  
173,  
5,  
229,  
163,  
197,  
163,  
73,  
151,  
239,  
73,  
163,  
149,  
2,  
251,  
233,  
79,  
199,  
241,  
73,  
137,  
163,  
227,  
139,  
211,  
167,  
167,  
233,  
241,  
137,  
107,  
17,  
113,  
17,  
157,  
251,  
41,  
83,  
149,  
137,  
157,  
223,

71,  
227,  
107,  
229,  
101,  
47,  
179,  
53,  
19,  
193,  
193,  
109,  
5,  
229,  
67,  
239,  
251,  
67,  
229,  
73,  
23,  
3,  
149,  
101,  
137,  
233,  
137,  
179,  
251,  
7,  
113,  
211,  
43,  
3,  
19,  
251,  
233,  
173,  
47,  
19,  
239,  
103,

71,  
67,  
241,  
139,  
107,  
251,  
3,  
131,  
37,  
103,  
193,  
211,  
109,  
17,  
41,  
61,  
83,  
211,  
19,  
59,  
241,  
79,  
251,  
47,  
53,  
211,  
173,  
227,  
251,  
109,  
59,  
59,  
149,  
37,  
113,  
229,  
89,  
103,  
5,  
157,  
137,  
11,

89,  
137,  
227,  
89,  
53,  
107,  
7,  
241,  
227,  
97,  
199,  
2,  
131,  
7,  
5,  
173,  
109,  
5,  
167,  
239,  
47,  
59,  
11,  
149,  
101,  
67,  
3,  
83,  
223,  
167,  
101,  
149,  
71,  
3,  
83,  
101,  
43,  
83,  
47,  
107,  
59,  
11,

```
47,  
61,  
83,  
53,  
193,  
179,  
101,  
103,  
179,  
113,  
199,  
197,  
113,  
229,  
3,  
223,  
37,  
29,  
163,  
251,  
101,  
2,  
211,  
211,  
31,  
7,  
139,  
163,  
241,  
103,  
137,  
223,  
41,  
37,  
127,  
...]
```

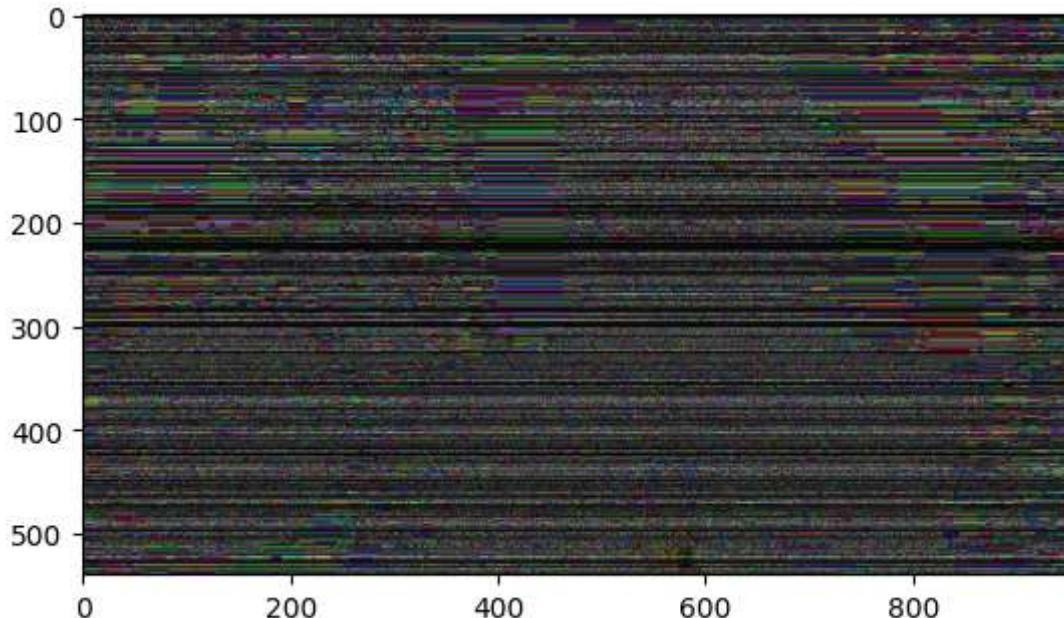
```
In [49]: ra =np.random.randint(0,255,size=len(modi1)) #generating an array with the values 0-255 with 540 rows  
ra
```

```
Out[49]: array([153, 217, 202, 165, 208, 91, 61, 219, 42, 96, 12, 246, 61,
   104, 18, 241, 56, 88, 147, 11, 234, 29, 38, 225, 38, 81,
   172, 90, 97, 233, 253, 231, 42, 13, 138, 243, 66, 178, 52,
   106, 17, 169, 1, 217, 130, 138, 130, 86, 247, 220, 218, 170,
   118, 9, 176, 188, 139, 39, 181, 76, 160, 15, 221, 248, 241,
   148, 206, 246, 135, 110, 145, 14, 97, 57, 42, 41, 204, 67,
   71, 30, 81, 40, 68, 185, 74, 42, 212, 111, 24, 149, 65,
   143, 40, 88, 207, 246, 212, 186, 241, 88, 22, 158, 81, 49,
   190, 249, 37, 246, 170, 1, 147, 226, 68, 4, 113, 183, 121,
   120, 174, 168, 3, 77, 52, 49, 5, 167, 63, 73, 29, 70,
   192, 189, 10, 88, 165, 198, 254, 43, 91, 14, 204, 171, 160,
   232, 108, 64, 28, 21, 167, 30, 89, 93, 213, 92, 105, 185,
   48, 100, 205, 217, 121, 77, 180, 132, 79, 10, 113, 123, 103,
   54, 40, 112, 157, 215, 149, 212, 23, 72, 81, 98, 209, 108,
   162, 31, 208, 120, 237, 235, 197, 85, 233, 14, 191, 97, 26,
   102, 14, 73, 107, 91, 197, 48, 141, 208, 63, 67, 90, 229,
   141, 35, 151, 65, 177, 67, 7, 36, 34, 239, 226, 141, 18,
   174, 37, 228, 63, 81, 29, 156, 247, 79, 46, 39, 3, 214,
   158, 46, 189, 92, 233, 108, 158, 1, 128, 24, 163, 44, 14,
   167, 118, 252, 83, 98, 209, 191, 93, 60, 249, 239, 142, 254,
   234, 184, 137, 63, 107, 152, 207, 60, 247, 88, 15, 97, 232,
   122, 23, 148, 82, 81, 2, 86, 74, 89, 126, 40, 161, 125,
   239, 5, 159, 158, 190, 69, 49, 238, 62, 69, 120, 206, 81,
   168, 178, 92, 121, 142, 142, 93, 179, 168, 116, 242, 20, 73,
   152, 215, 1, 151, 160, 51, 12, 179, 62, 122, 89, 22, 185,
   10, 231, 172, 234, 20, 219, 120, 187, 155, 153, 188, 154, 84,
   76, 228, 229, 111, 243, 17, 215, 174, 31, 4, 244, 151, 247,
   110, 221, 138, 193, 138, 167, 147, 28, 197, 13, 154, 37, 7,
   244, 191, 211, 176, 251, 46, 69, 152, 122, 101, 121, 180, 88,
   149, 129, 44, 210, 77, 13, 125, 229, 103, 249, 83, 92, 200,
   221, 140, 115, 82, 2, 109, 190, 153, 94, 46, 53, 5, 155,
   99, 236, 158, 4, 25, 33, 126, 193, 142, 162, 141, 7, 131,
   122, 59, 178, 174, 242, 145, 189, 64, 42, 5, 47, 61, 231,
   1, 140, 205, 35, 58, 97, 1, 84, 214, 207, 1, 32, 99,
   121, 224, 119, 68, 243, 29, 218, 249, 30, 246, 184, 208, 121,
   197, 161, 102, 201, 12, 113, 195, 79, 236, 218, 49, 68, 1,
   112, 5, 25, 30, 98, 116, 130, 104, 157, 234, 88, 35, 205,
   236, 234, 160, 208, 69, 14, 175, 15, 144, 44, 20, 168, 252,
   22, 12, 153, 176, 81, 30, 220, 147, 213, 201, 228, 172, 30,
   25, 178, 210, 171, 141, 145, 6, 174, 252, 234, 194, 142, 3,
```

```
102, 25, 219, 112, 127, 75, 236, 174, 245, 248, 215, 7, 193,
225, 12, 76, 12, 219, 115, 128])
```

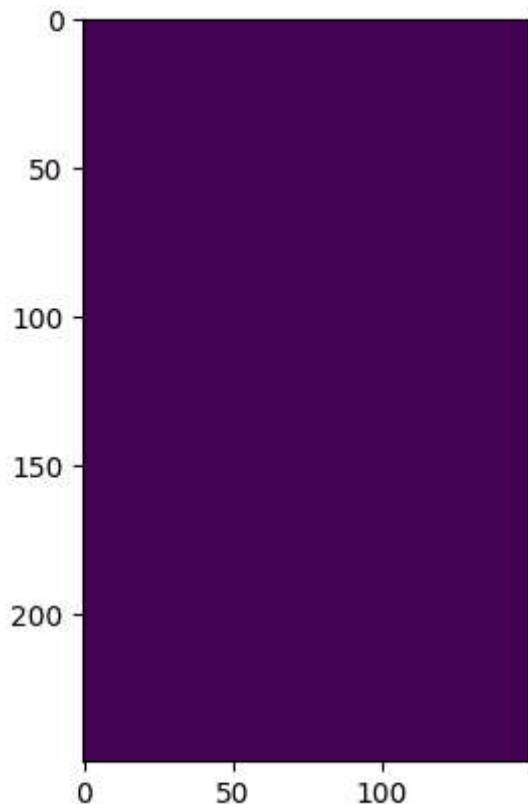
```
In [50]: encrypt =np.array([x*(x^y)%z for x, y, z in zip(modi1,ra,prime)])
plt.imshow(encrypt) #for displaying image
#modi1 consists of pixels
```

```
Out[50]: <matplotlib.image.AxesImage at 0x1f157182e00>
```



```
In [136... #creating own images
#creating θ matix of size 250,150
imag1 =np.zeros((250,150))
plt.imshow(imag1)
```

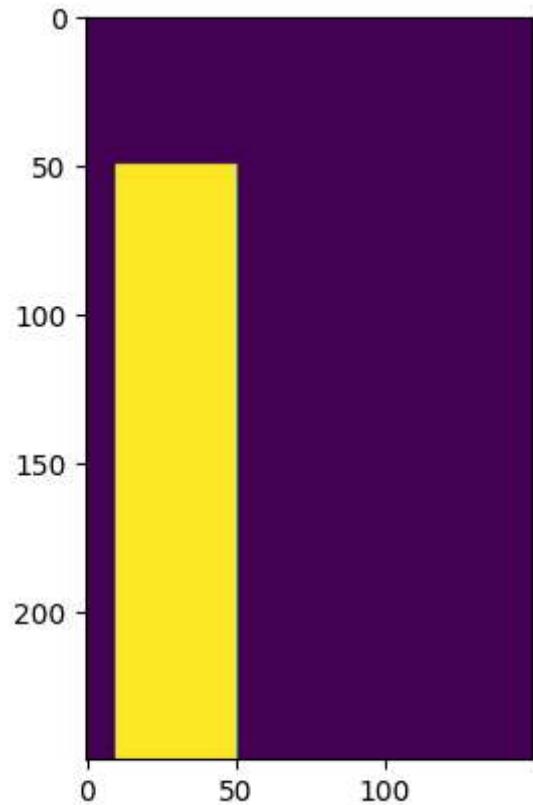
```
Out[136]: <matplotlib.image.AxesImage at 0x1f167b3ba00>
```



```
In [137]: rectangle =cv2.rectangle(imag1,(50,50),(10,1000),254,-1)
#top left starting point -->1st parenthesis
#bottom right ending point
#-1 is thickness of rectangle
#255 color
```

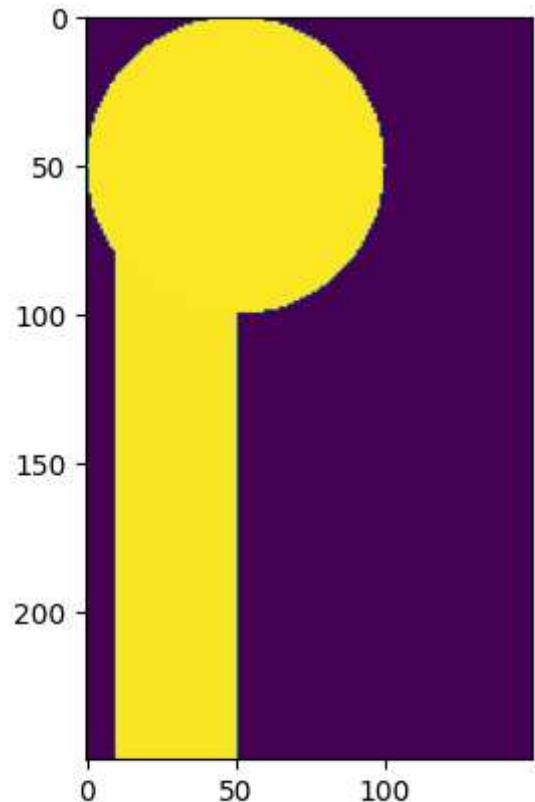
```
In [138]: plt.imshow(rectangle)
```

```
Out[138]: <matplotlib.image.AxesImage at 0x1f167baa080>
```



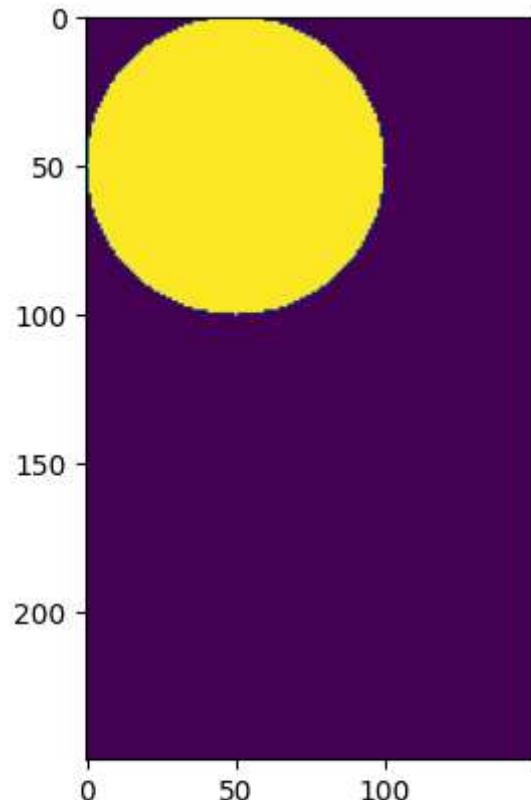
```
In [140]: circle = cv2.circle(imag1,(50,50),50,255,-1)
#2nd value is radius that is 50.
plt.imshow(circle)
```

Out[140]: <matplotlib.image.AxesImage at 0x1f167a277f0>



```
In [141]: img2=np.zeros((250,150))
circle = cv2.circle(img2,(50,50),50,255,-1)
plt.imshow(circle)
```

Out[141]: <matplotlib.image.AxesImage at 0x1f167a99ae0>



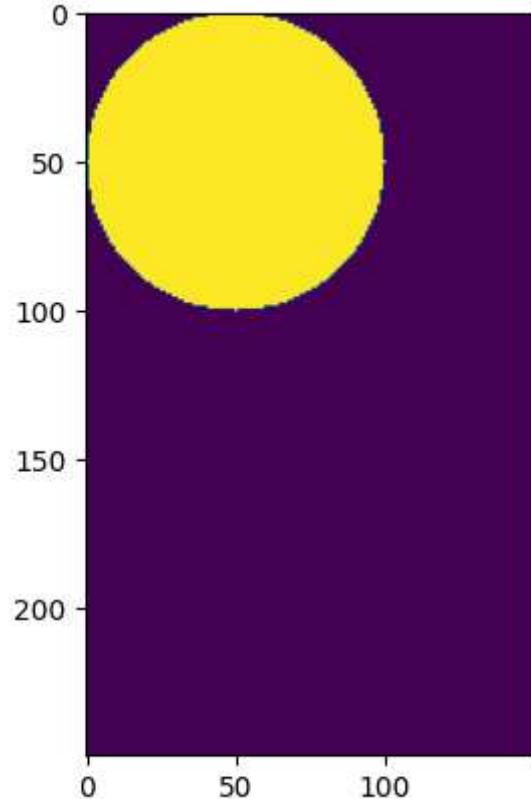
In [ ]:

```
In [142]: bitand = cv2.bitwise_and(rectangle,circle) #opn performed between each pixel of rectangle and circle  
#it will give common pixel values  
bitand
```

```
Out[142]: array([[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 [143]: plt.imshow(bitand)
```

```
Out[143]: <matplotlib.image.AxesImage at 0x1f167d98280>
```

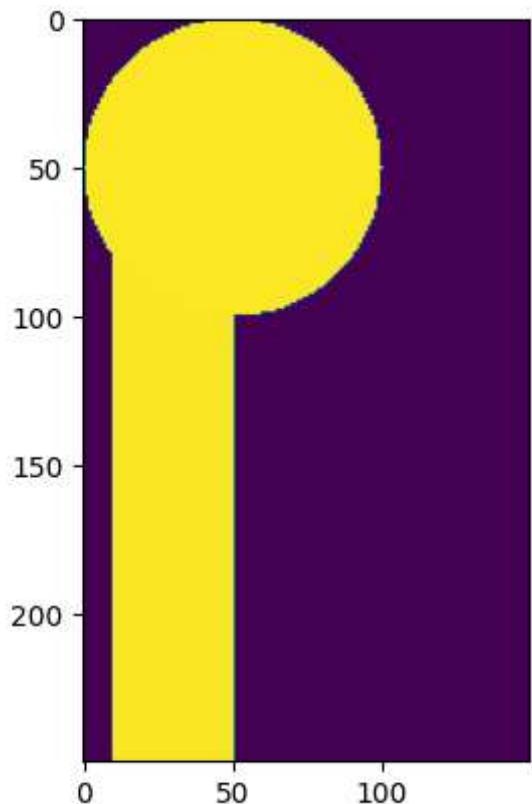


```
In [144]: bitor=cv2.bitwise_or(rectangle,circle)
bitor
```

```
Out[144]: array([[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.]])
```

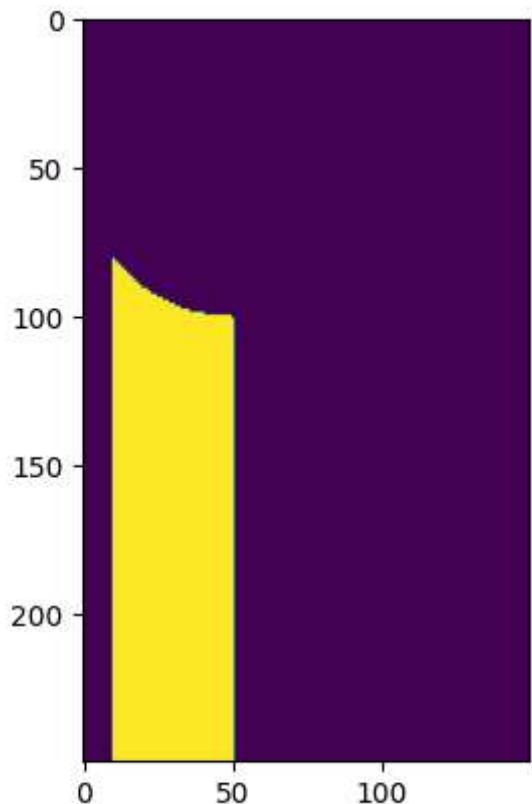
```
In [145]: plt.imshow(bitor) #merges 2 images
```

```
Out[145]: <matplotlib.image.AxesImage at 0x1f167de2830>
```



```
In [146]: bitxor=cv2.bitwise_xor(rectangle,circle)
plt.imshow(bitxor)
```

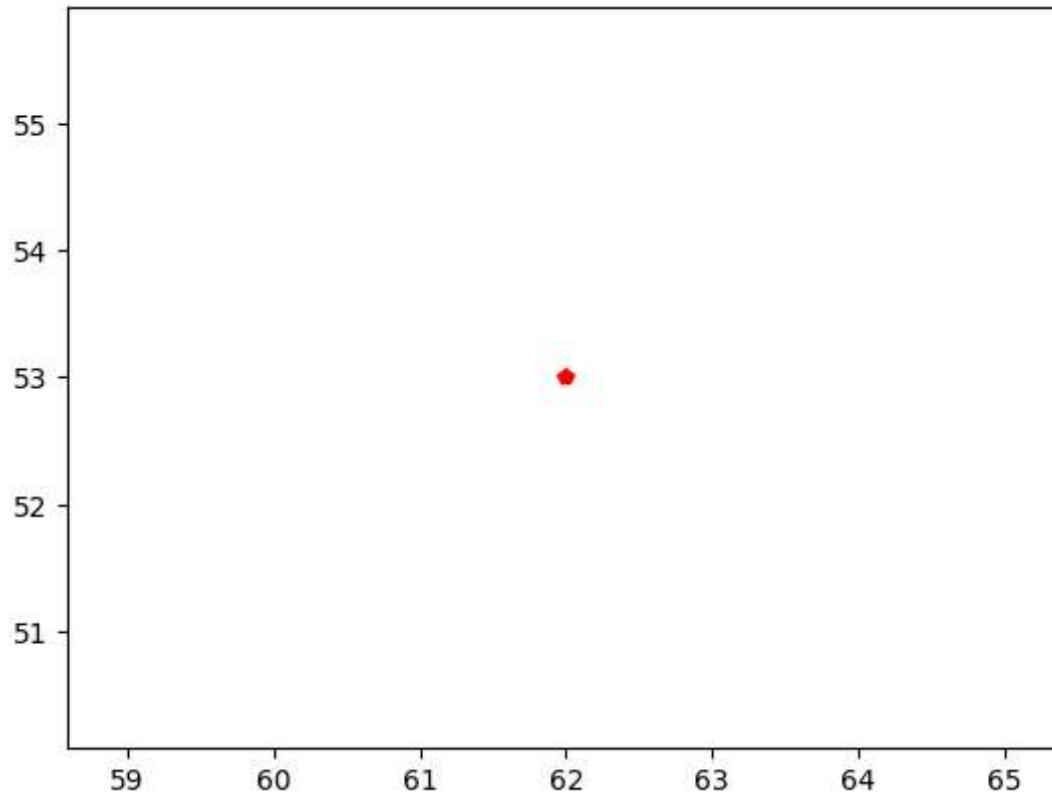
Out[146]: <matplotlib.image.AxesImage at 0x1f167c11600>



Scalar: quantity described by numerical value

```
In [150]: #string is a categorical value
x=62
y=53
z='aiml'
plt.plot(x,y,'rp') #o is a point in the graph #r is a color
```

```
Out[150]: []
```



```
In [151]: a1=x+y  
a1
```

Out[151]: 115

```
In [152]: c1=x*y  
c1
```

Out[152]: 3286

```
In [153]: d1=x^y  
d1
```

Out[153]: 11

vector: ordered list of scalar values

```
In [158... v1=np.array([1,2,3])
v2=np.array([3,5,-7])
```

```
In [159... v1.shape #tells no. of elements present in v1
```

```
Out[159]: (3,)
```

```
In [160... v3= v1+v2
v3
```

```
Out[160]: array([ 4,  7, -4])
```

```
In [161... v3 =v1-v2
v3
```

```
Out[161]: array([-2, -3, 10])
```

```
In [163... v3=v1%v2
v3
```

```
Out[163]: array([ 1,  2, -4])
```

```
In [164... v3 = v1*v2
v3
```

```
Out[164]: array([ 3, 10, -21])
```

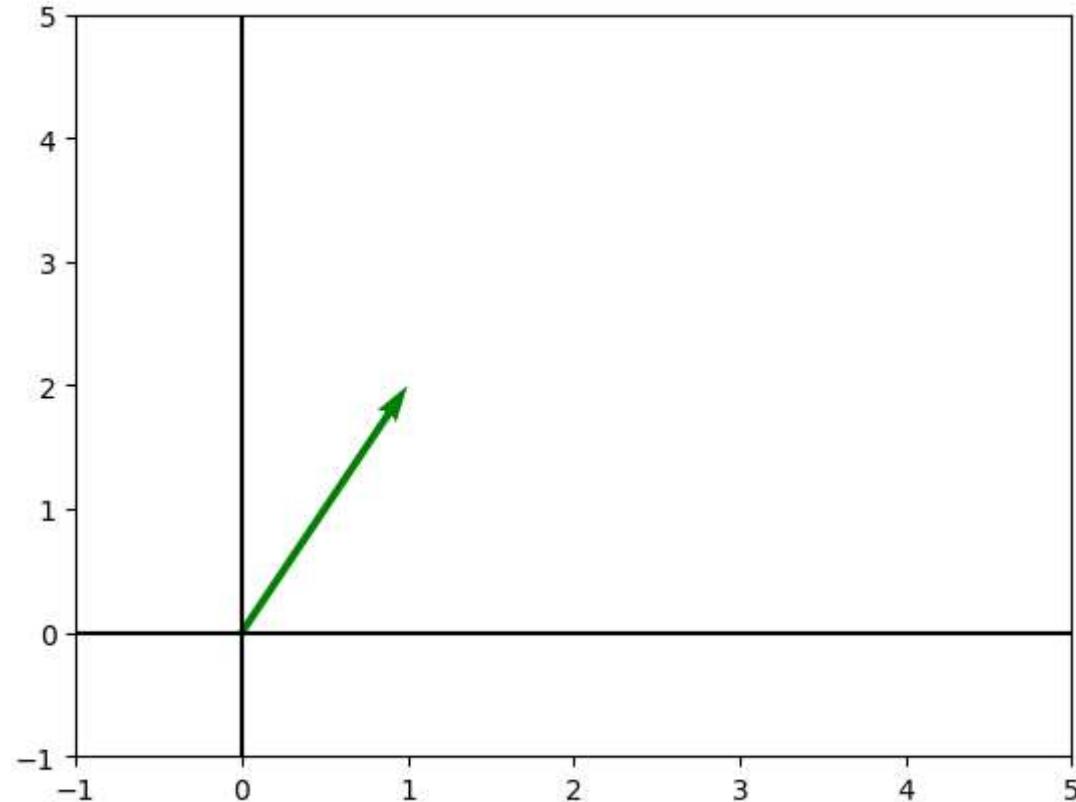
```
In [165... v3[0]
```

```
Out[165]: 3
```

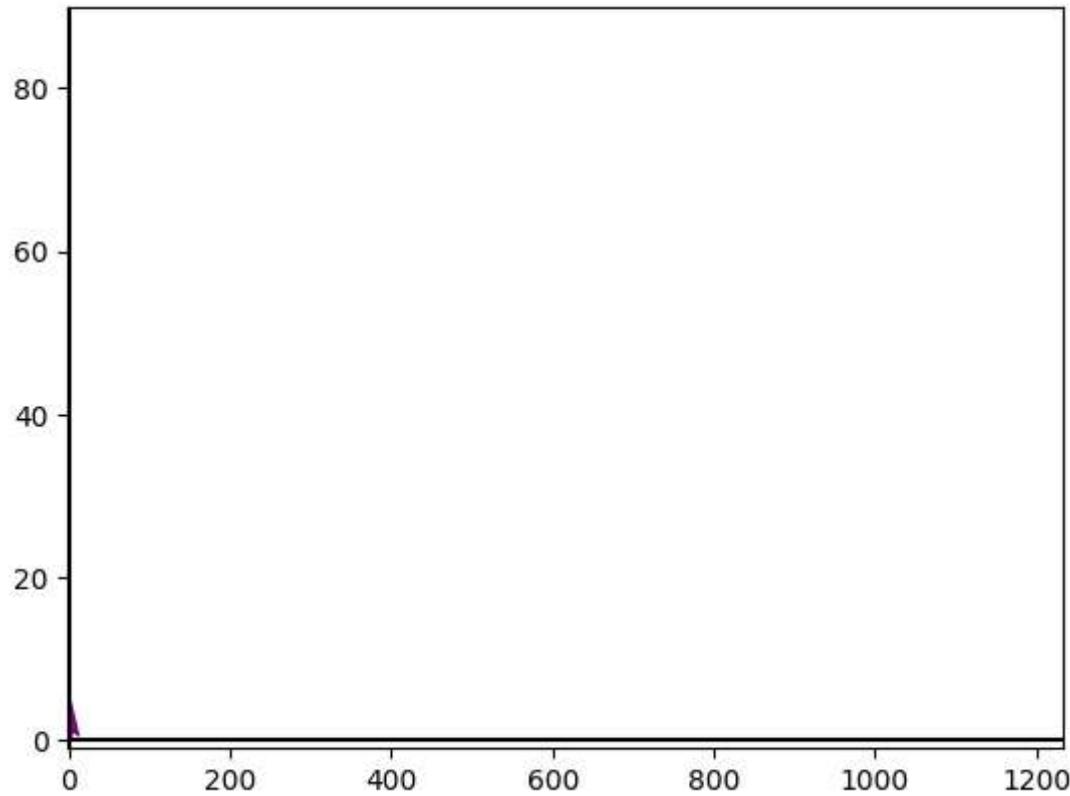
```
In [199... def plot_vectors(vectors, colors):
plt.axvline(x=0,color='black')
plt.axhline(y=0,color='black')#plotting vectors
#v=[1,2] x=[0,0,1,2] #0 is starting of x then starting of y and ends at 1 and 2
for i in range(len(vectors)):#any no. of vectors can be taken
    x = np.concatenate([[0,0],vectors[i]])
```

```
plt.quiver([x[0]],[x[1]],[x[2]],[x[3]],
angles='xy',scale_units='xy',scale=1, color=colors[i])
```

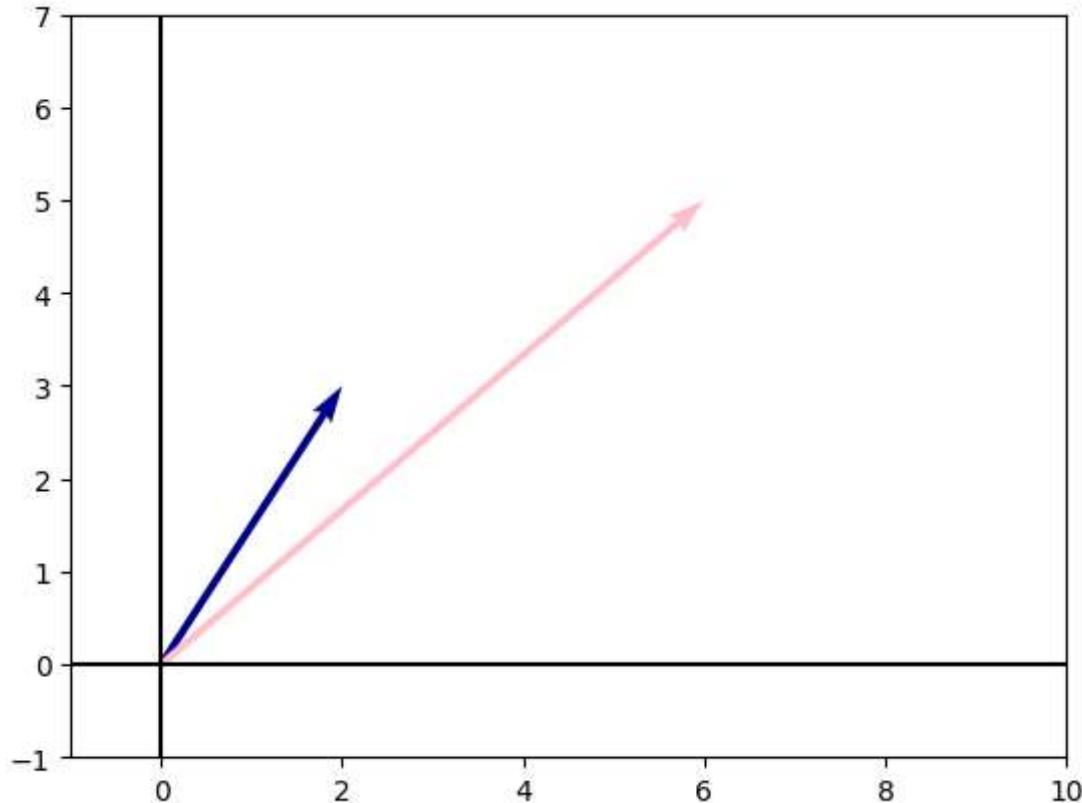
```
In [200... v=[1,2]
plot_vectors([v],['green'])
plt.xlim(-1,5)
_=plt.ylim(-1,5)
```



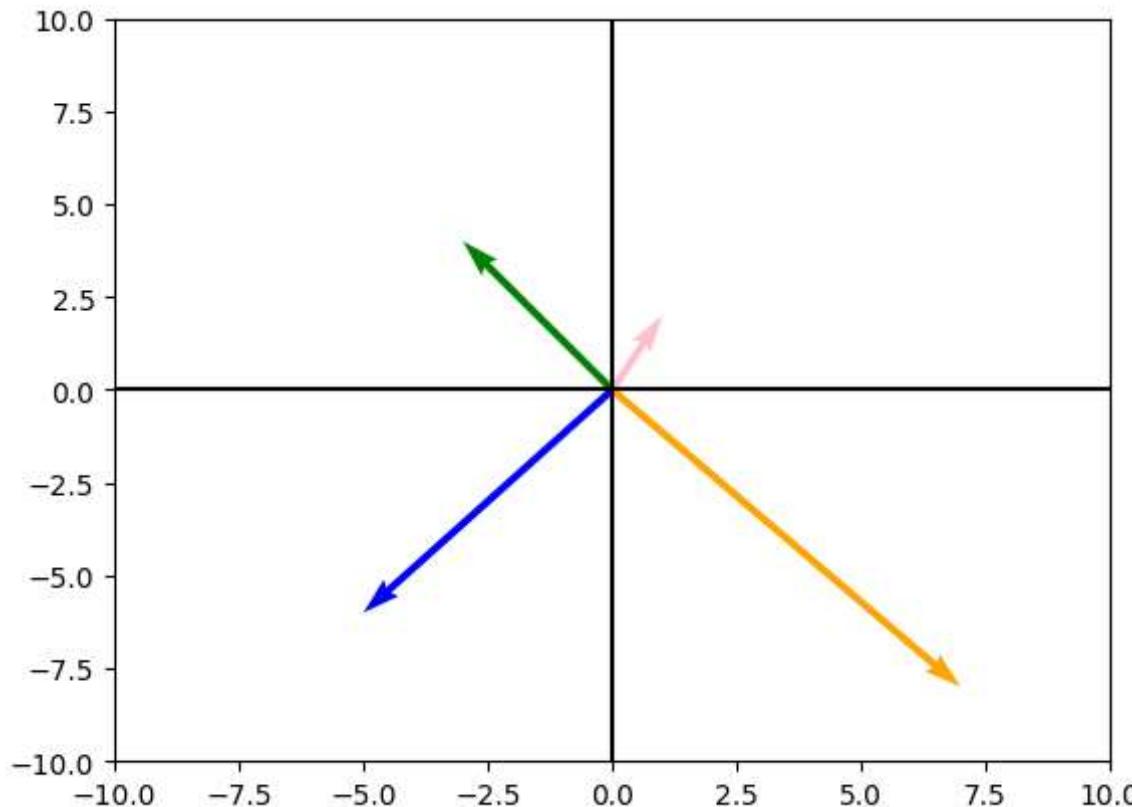
```
In [201... v=[3,5]
plot_vectors([v],['purple'])
plt.xlim(-1,1234)
_=plt.ylim(-1,90)
```



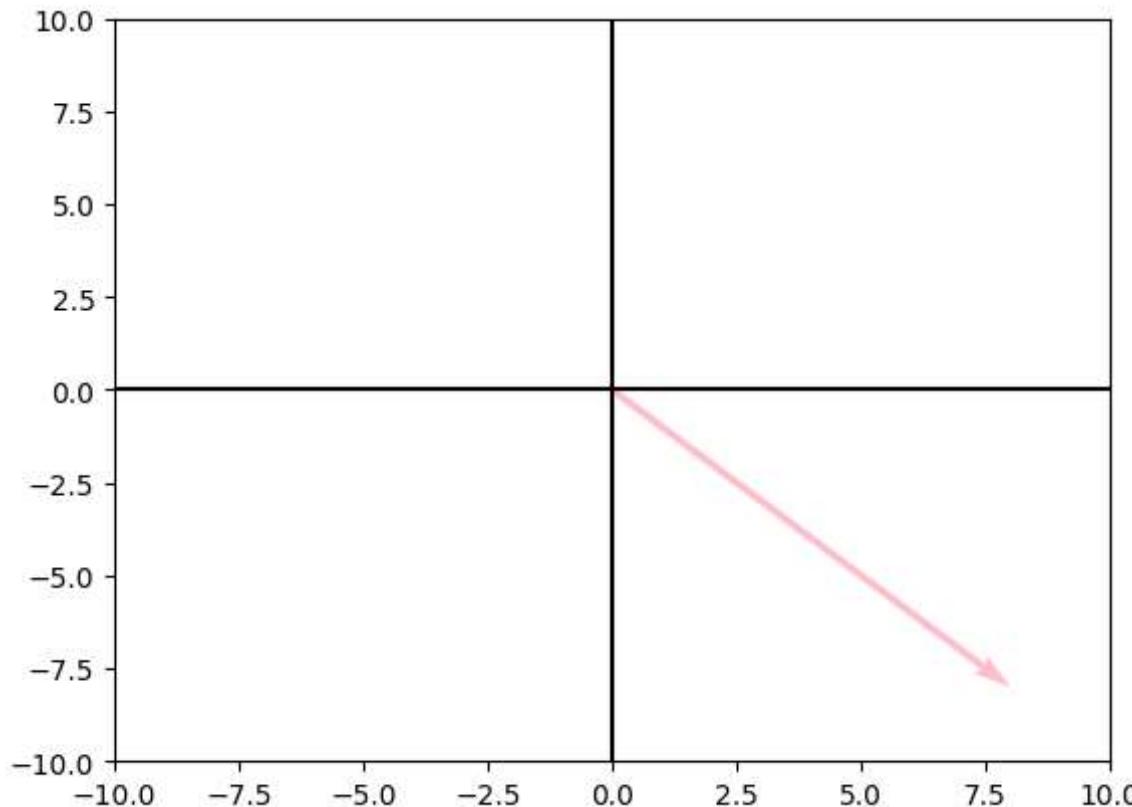
```
In [202]: v1=[2,3]
v2=[6,5]
plot_vectors([v1,v2],['darkblue','pink'])
plt.xlim(-1,10)
_=plt.ylim(-1,7)
```



```
In [203]: v1=[1,2]
v2=[-3,4]
v3=[-5,-6]
v4=[7,-8]
plot_vectors([v1,v2,v3,v4],['pink','green','blue','orange'])
plt.xlim(-10,10)
_=plt.ylim(-10,10)
```



```
In [207]: v1=np.array([1,2])
v2=np.array([-3,4])
v3=np.array([-5,-6])
v4=np.array([15,-8])
v5=v1+v2+v3+v4
plot_vectors([v5],['pink'])
plt.xlim(-10,10)
_=plt.ylim(-10,10)
```



```
In [208]: #mag of vector  
np.linalg.norm(v1)
```

Out[208]: 2.23606797749979

```
In [214]: np.dot(v1,v2)#inner product of vectors
```

Out[214]: 5

```
In [236]: angle = (np.dot(v1,v2))/((np.linalg.norm(v1))*np.linalg.norm(v2))  
np.arccos(angle) #finding angle btw vectors
```

Out[236]: 1.1071487177940904

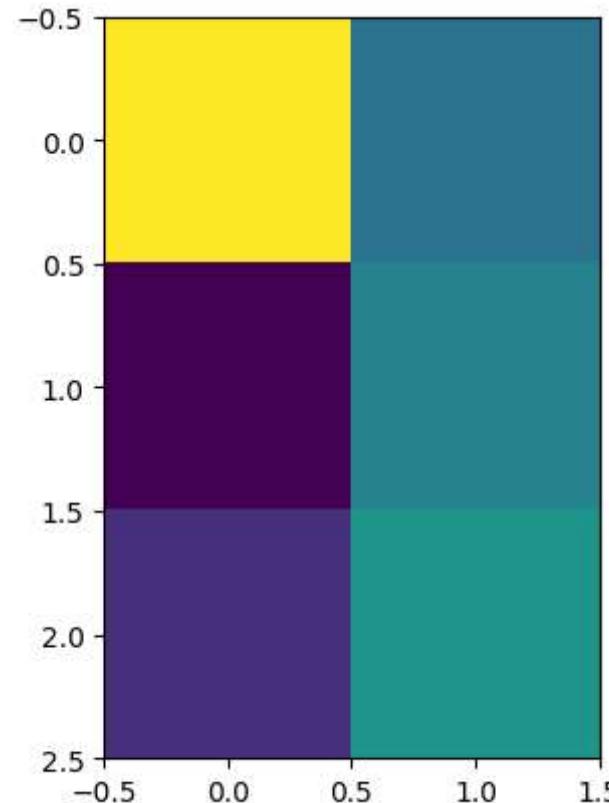
```
In [259]: #ordered List of vectors-->matrices
```

```
m1=np.array([[20,2],[-9,4],[-5,6]]) #[1,2] is 1st row  
m1
```

```
Out[259]: array([[20,  2],  
                  [-9,  4],  
                  [-5,  6]])
```

```
In [260... plt.imshow(m1) #each box corresponds to rows and coloumns colors=valuesb
```

```
Out[260]: <matplotlib.image.AxesImage at 0x1f167f22680>
```



```
In [261... m1.shape
```

```
Out[261]: (3, 2)
```

```
In [263... m2=np.array([[900,6],[-60,9],[-1,3]])
```

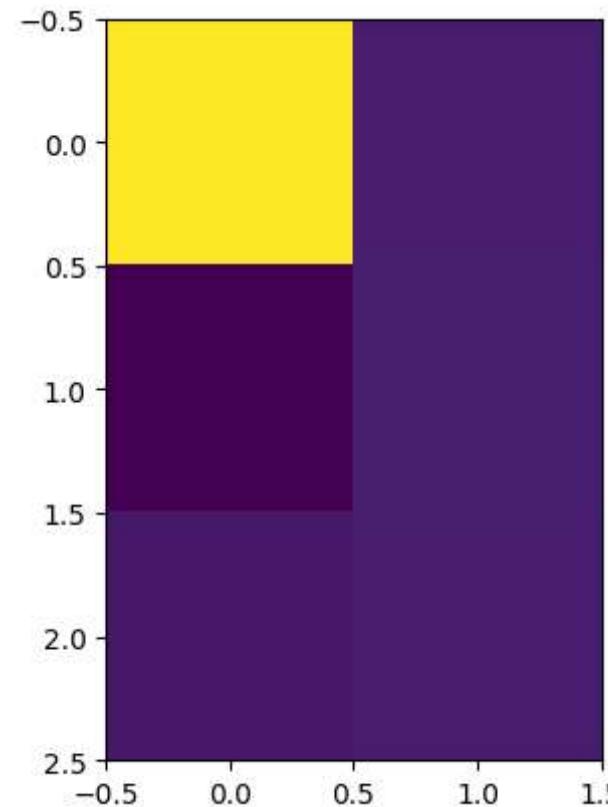
```
In [264]: m3=m1+m2
```

```
m3
```

```
Out[264]: array([[920,    8],  
                  [-69,   13],  
                  [- 6,    9]])
```

```
In [265]: plt.imshow(m3)
```

```
Out[265]: <matplotlib.image.AxesImage at 0x1f16955bee0>
```



```
In [267]: m4=m1*m2
```

```
m4
```

```
Out[267]: array([[18000,    12],  
                  [   540,    36],  
                  [     5,    18]])
```

```
In [270... m5=np.array([[1,2],[3,4]])  
m6=np.array([[1,1],[2,2]])
```

```
In [273... m5
```

```
Out[273]: array([[1,  2],  
                  [3,  4]])
```

```
In [274... m6
```

```
Out[274]: array([[1,  1],  
                  [2,  2]])
```

```
In [275... np.dot(m5,m6)
```

```
Out[275]: array([[ 5,  5],  
                  [11, 11]])
```