Name: Asad Haroon

Reg#: SP17-BCS-012

Section C

Note: I am Using Scipy v1.1.0. In case of any errors you can install scipy 1.1.0 by the following command: pip install scipy==1.1.0

In [1]:

```
import scipy
scipy.__version__
```

Out[1]:

```
'1.1.0'
```

To download images Click on this url: https://drive.google.com/drive/folders/1pcaTwofZGfoCxZ3Hv2X6vW6xf_1i88eb?usp=sharing (https://drive.google.com/drive/folders/1pcaTwofZGfoCxZ3Hv2X6vW6xf_1i88eb?usp=sharing)

Import Libraries

In [2]:

```
from skimage import data
from scipy.misc import imread, imresize
import numpy as np
from scipy import ndimage
import matplotlib.pyplot as plt
```

Original Image

In [3]:

```
original_image =imread('camera.png',True, 'L')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: DeprecationWarning: `imread` is deprecate
d!
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
  """Entry point for launching an IPython kernel.
```

In [4]:

```
processed_image=original_image.copy()
processed_image=processed_image.astype(np.int32)
```

Helper Function

In [5]:

```
def returnNumber(x,b,m):
    a=(bin(x).replace("0b",''))
    a=[int(i) for i in a]
    count=8-len(a)
    for i in range(0,count):
        a.insert(i,0)
    a=m*a[b]
    return a
```

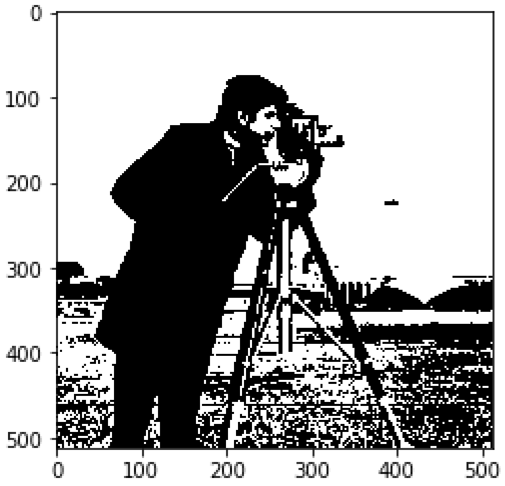
Eight Bit Plane Slicing

In [6]:

```
#At 7bit plane
eight_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
eight_bit_slice=f(eight_bit_slice,0,128)
```

In [7]:
```
plt.imshow(eight_bit_slice,plt.cm.gray)
```

Out[7]:
```
<matplotlib.image.AxesImage at 0x19a26588948>
```

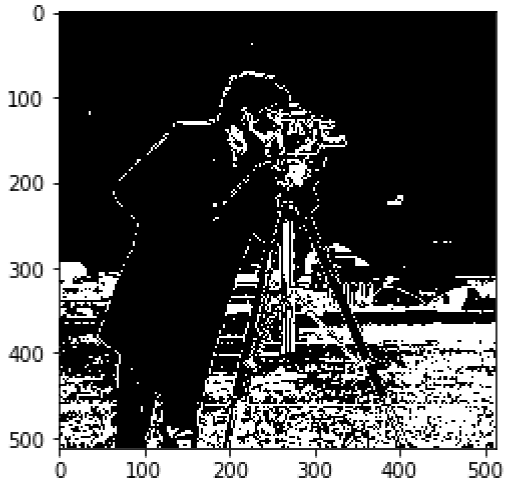


Seven Bit Plane Slicing

In [8]:
```
#At 6bit plane
seven_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
seven_bit_slice=f(seven_bit_slice,1,64)
```

In [9]:
```
plt.imshow(seven_bit_slice,plt.cm.gray)
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x19a3074d0c8>
```



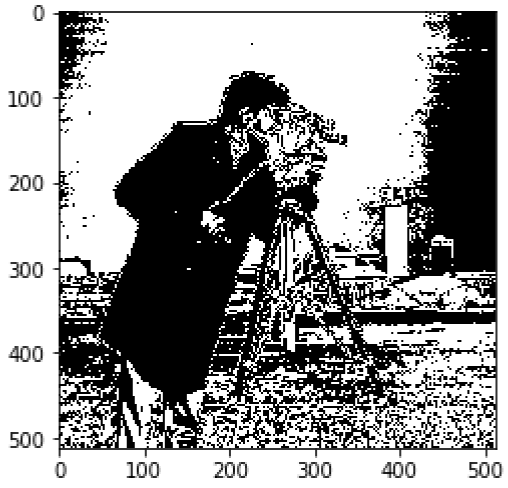
Six Bit Plane Slicing

In [10]:
```
#At 5bit plane
six_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
six_bit_slice=f(six_bit_slice,2,32)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(six_bit_slice),np.amax(six_bit_sl
```

```
Min intensity level is 0 and Max intensity level is 32
```

In [11]:
```
plt.imshow(six_bit_slice,plt.cm.gray)
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x19a308c6cc8>
```



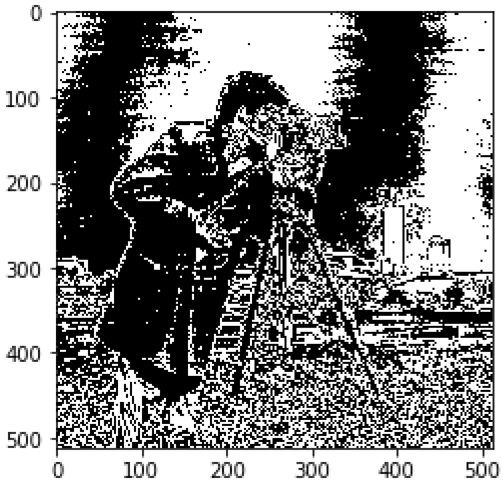
Five Bit Plane Slicing

In [12]:
```
#At 4bit plane
five_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
five_bit_slice=f(five_bit_slice,3,16)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(five_bit_slice),np.amax(five_bit_
```

```
Min intensity level is 0 and Max intensity level is 16
```

In [13]:
```
plt.imshow(five_bit_slice,plt.cm.gray)
```

Out[13]:
```
<matplotlib.image.AxesImage at 0x19a30a3bf48>
```



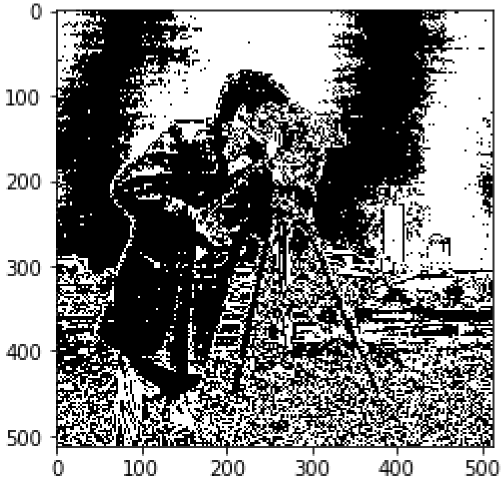
Four Bit Plane Slicing

In [14]:
```
#At 3bit plane
four_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
four_bit_slice=f(four_bit_slice,4,8)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(four_bit_slice),np.amax(four_bit_
```

```
Min intensity level is 0 and Max intensity level is 8
```

In [15]:
```
plt.imshow(five_bit_slice,plt.cm.gray)
```

Out[15]:
```
<matplotlib.image.AxesImage at 0x19a30baff48>
```



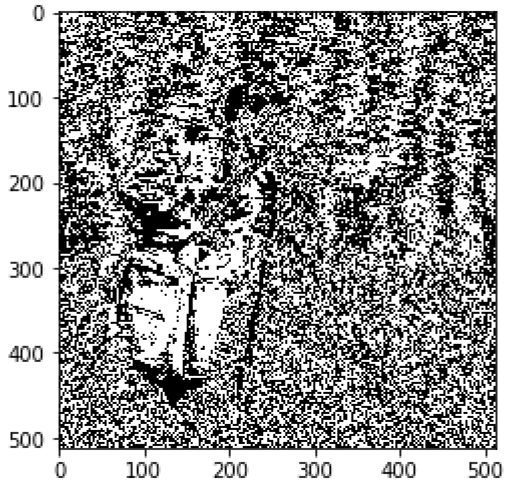
Three Bit Plane Slicing

In [16]:
```
#At 2 bit plane
three_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
three_bit_slice=f(three_bit_slice,5,4)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(three_bit_slice),np.amax(three_bi
```

```
Min intensity level is 0 and Max intensity level is 4
```

In [17]:
```
plt.imshow(three_bit_slice,plt.cm.gray)
```

Out[17]:
```
<matplotlib.image.AxesImage at 0x19a30c18988>
```



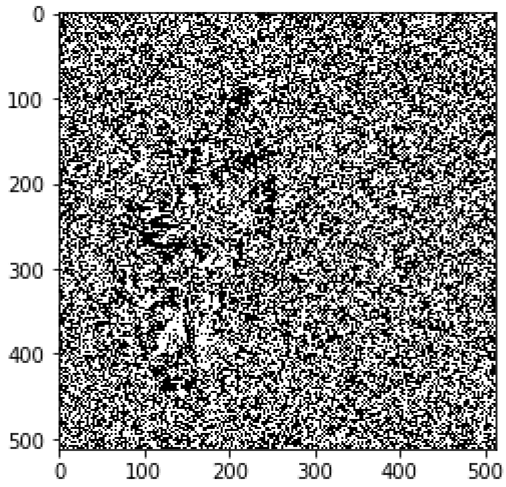
Two Bit Plane Slicing

In [18]:
```
# At 1th bit plane
two_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
two_bit_slice=f(two_bit_slice,6,2)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(two_bit_slice),np.amax(two_bit_sl
```

```
Min intensity level is 0 and Max intensity level is 2
```

In [19]:
```
plt.imshow(two_bit_slice,plt.cm.gray)
```

Out[19]:
```
<matplotlib.image.AxesImage at 0x19a30e9dac8>
```



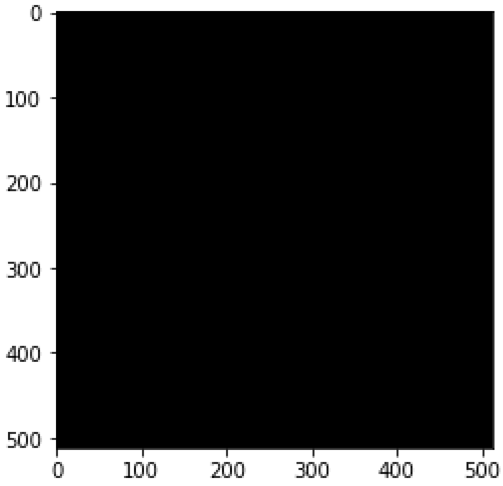
One Bit (i.e, at 0 bit) Plane Slicing

In [20]:
```
one_bit_slice=processed_image.copy()
f=np.vectorize(returnNumber)
one_bit_slice=f(one_bit_slice,7,0)
print("Min intensity level is {} and Max intensity level is {}".format(np.amin(one_bit_slice),np.amax(one_bit_sl
```

```
Min intensity level is 0 and Max intensity level is 0
```

```
In [21]: plt.imshow(one_bit_slice,plt.cm.gray)

Out[21]: <matplotlib.image.AxesImage at 0x19a30f044c8>
```



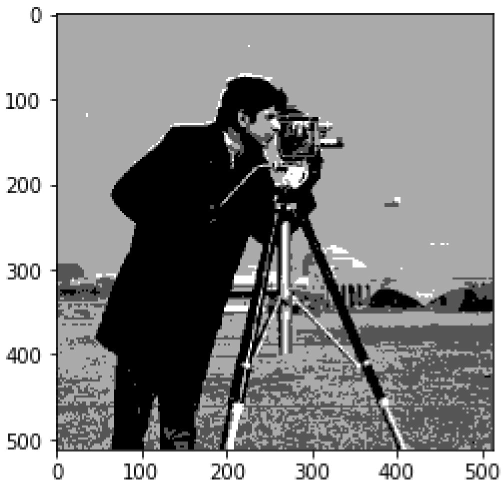
Concat of Bit planes

In [22]:

```
#8+7 bit
eight_seven_bit=eight_bit_slice+seven_bit_slice
plt.imshow(eight_seven_bit,plt.cm.gray)
```

Out[22]:

```
<matplotlib.image.AxesImage at 0x19a30f67588>
```

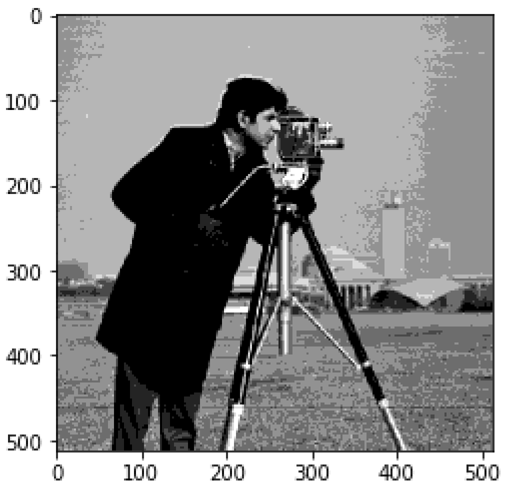


In [23]:

```
#8+7+6 bit
eight_seven_six_bit=eight_bit_slice+seven_bit_slice+six_bit_slice
plt.imshow(eight_seven_six_bit,plt.cm.gray)
```

Out[23]:

```
<matplotlib.image.AxesImage at 0x19a311ee2c8>
```

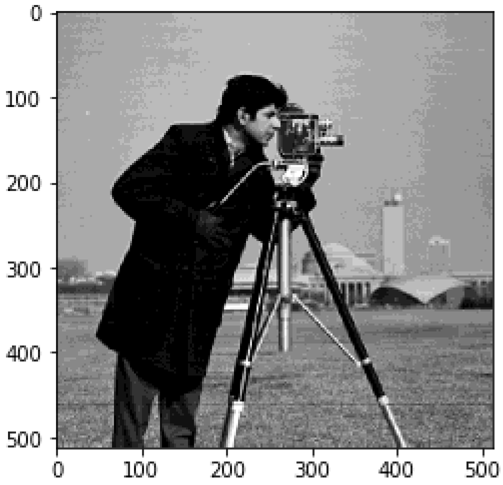


In [24]:

```
#8+7+6+5 bit
eight_seven_six_5_bit=eight_bit_slice+seven_bit_slice+six_bit_slice+five_bit_slice
plt.imshow(eight_seven_six_5_bit,plt.cm.gray)
```

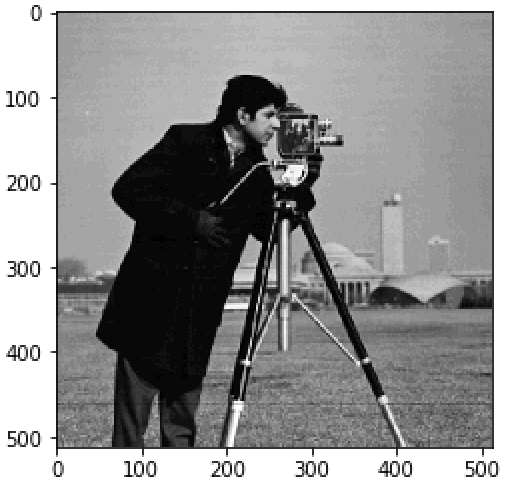
Out[24]:

```
<matplotlib.image.AxesImage at 0x19a31253c88>
```



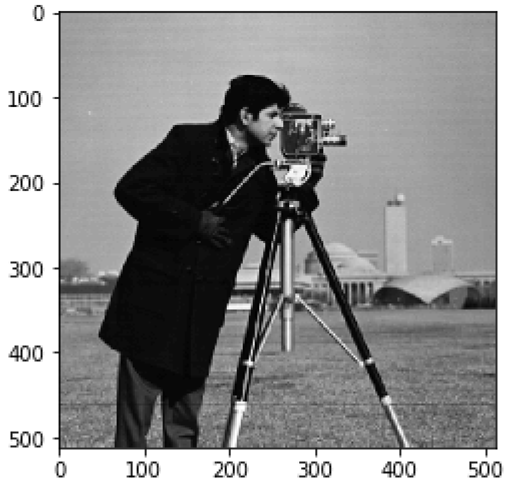
In [25]:
```
#8+7+6+5+4 bit
eight_seven_six_5_4_bit=eight_bit_slice+seven_bit_slice+six_bit_slice+five_bit_slice+four_bit_slice
plt.imshow(eight_seven_six_5_4_bit,plt.cm.gray)
```

Out[25]:
```
<matplotlib.image.AxesImage at 0x19a314d5fc8>
```



In [26]:
```
#8+7+6+5+4+3 bit
eight_seven_six_5_4_3_bit=eight_bit_slice+seven_bit_slice+six_bit_slice+five_bit_slice+four_bit_slice+three_bit_
plt.imshow(eight_seven_six_5_4_3_bit,plt.cm.gray)
```

Out[26]:
```
<matplotlib.image.AxesImage at 0x19a31537f48>
```

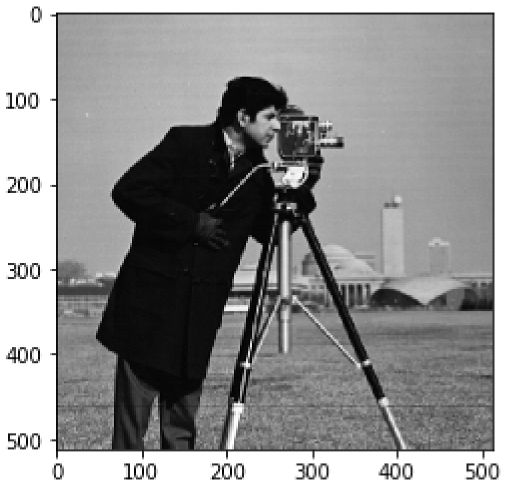


In [27]:

```
#8+7+6+5+4+3+2 bit
eight_seven_six_5_4_3_2_bit=eight_bit_slice+seven_bit_slice+six_bit_slice+five_bit_slice+four_bit_slice+three_bi
plt.imshow(eight_seven_six_5_4_3_2_bit,plt.cm.gray)
```

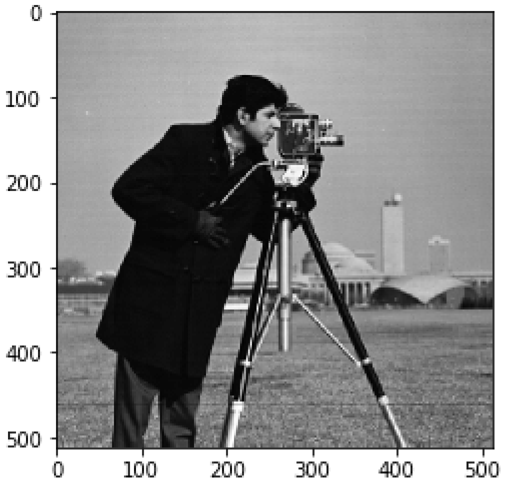
Out[27]:

```
<matplotlib.image.AxesImage at 0x19a3159cd88>
```



In [28]:
```
#8+7+6+5+4+3+2+1 bit
all_bits=eight_bit_slice+seven_bit_slice+six_bit_slice+five_bit_slice+four_bit_slice+three_bit_slice+two_bit_sli
plt.imshow(all_bits,plt.cm.gray)
```

Out[28]:
```
<matplotlib.image.AxesImage at 0x19a31823d08>
```



In [29]:

```
# As you can see original image and image made by bit plane slicing is not equal because in bit plane slicing, i
np.equal(processed_image,all_bits)
```

Out[29]:

```
array([[ True, False,  True, ...,  True,  True,  True],
       [ True, False, False, ...,  True,  True,  True],
       [ True, False,  True, ...,  True,  True,  True],
       ...,
       [False, False,  True, ..., False, False, False],
       [False, False,  True, ..., False, False, False],
       [False, False,  True, ..., False, False, False]])
```