In [1]:

```
from matplotlib import pyplot as plt
import numpy as np
import cv2 as cv

def create_blank (height, width, img_type):
    """Create new image(numpy array) """
    # Create black blank image
    image = np.zeros((height, width, 3 ), img_type)
    return image

img = cv.imread( "./images for PA1/berry.jpg" , cv.IMREAD_COLOR )

# The shape of an image is accessed by img.shape.
# It returns a tuple of the number of rows,
# columns, and channels (if the image is color):
print (img.shape)

rows = img.shape[0]
cols = img.shape[1]
channels = img.shape[ 2 ]
img2 = create_blank(rows, cols, np.uint8)

#1 channel for grey images
#turn the image into a grey image
for i in range ( 0 , rows- 1 ):
    for j in range ( 0 , cols- 1 ):
        px = img[i, j]
        #You can access a pixel value by its row and column coordinates.
        b = px[ 0 ]
        g = px[ 1 ]
        r = px[ 2 ]

        # don't worry about the warning about overflow. In this case, the value will never
        img2[i, j] = int ( ((b + g + r) / 3 ) )
```

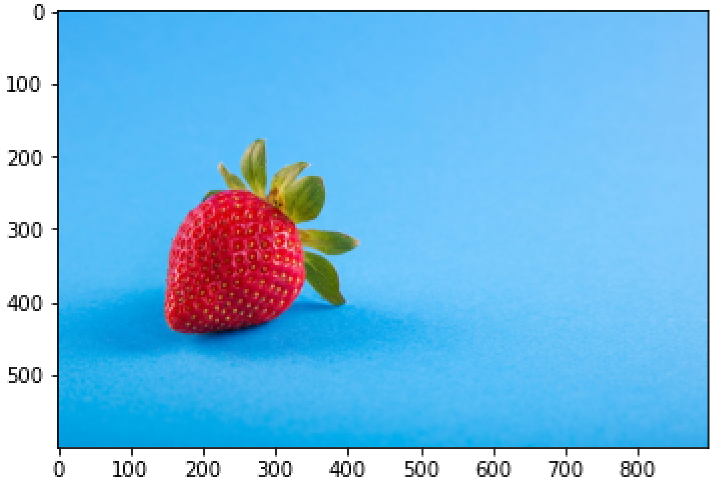
```
(600, 897, 3)

<ipython-input-1-e8ff55144896>:34: RuntimeWarning: overflow encountered in ubyte_scalar
s
  img2[i, j] = int ( ((b + g + r) / 3 ) )
```

Plot color image

In [2]:
```
timg = cv.cvtColor(img, cv.COLOR_BGR2RGB)
plt.imshow(timg)
```

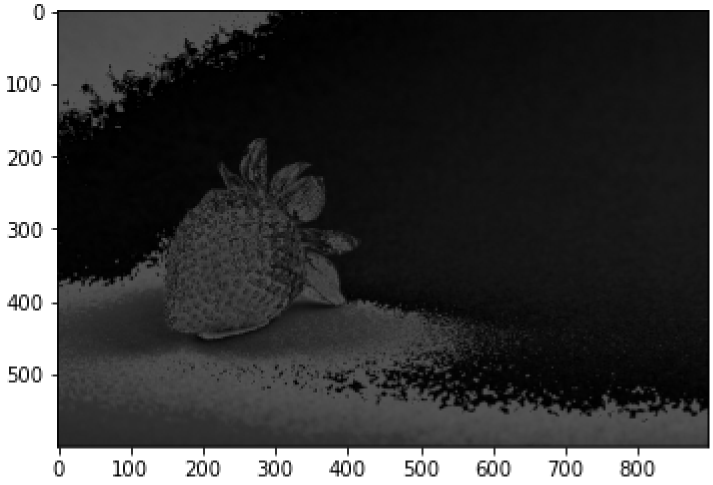
Out[2]:
```
<matplotlib.image.AxesImage at 0x29a39c6bc40>
```



Plot grey image

In [3]:
```
plt.imshow(img2)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x29a39f04160>
```

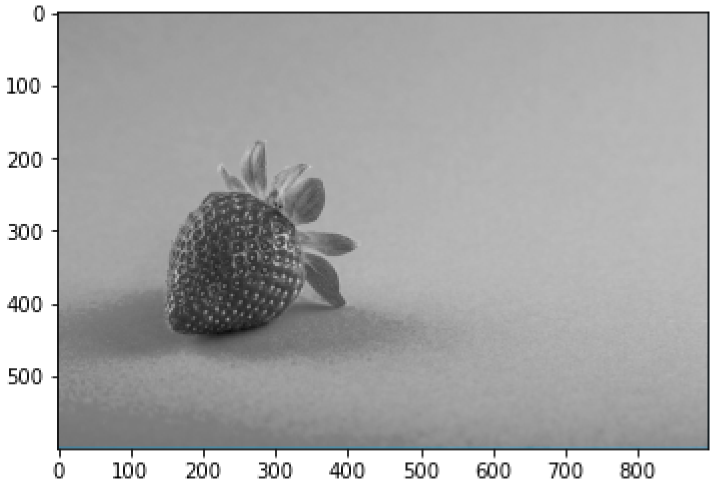


Task 1

```
In  [4]: task1_img = cv.cvtColor(img, cv.COLOR_BGR2RGB)
         for i in range ( 0 , rows- 1 ):
             for j in range ( 0 , cols- 1 ):
                 px = img[i, j]
                 b = px[ 0 ]
                 g = px[ 1 ]
                 r = px[ 2 ]

                 task1_img[i, j] = int ( 0.1140 * b + 0.5870 * g + 0.2990 * r )
         plt.imshow(task1_img)
```

```
Out[4]: <matplotlib.image.AxesImage at 0x29a3a105a90>
```

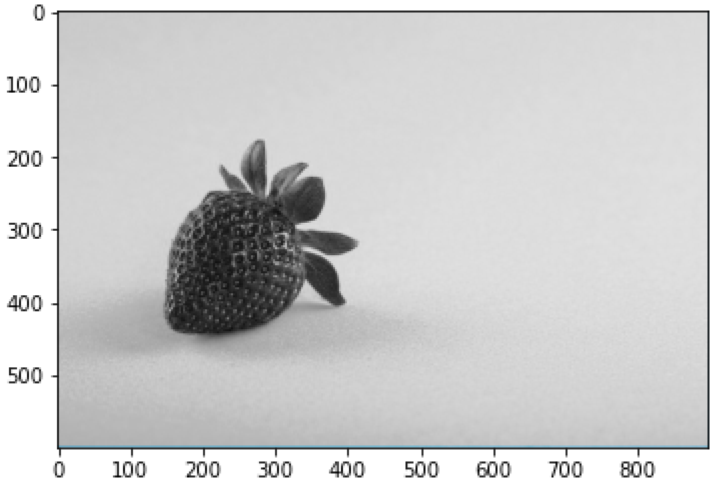


Task 2

In [5]:
```
task2_img = cv.cvtColor(img, cv.COLOR_BGR2RGB)
for i in range ( 0 , rows- 1 ):
    for j in range ( 0 , cols- 1 ):
        px = img[i, j]
        b = px[ 0 ]
        g = px[ 1 ]
        r = px[ 2 ]

        task2_img[i, j] = int ( 0.1140*r + 0.5870*b + 0.2990*g )
plt.imshow(task2_img)
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x29a3a6372e0>
```



Task 3

In [6]:
```
def BoxFilterGray3(image):

    rows = img.shape[0]
    cols = img.shape[1]
    Filter_image = image

    for x in range(rows):
        for y in range(cols):
            total = 0
            for x_i in range(max(0,x - 1),min(rows -1, x + 1)+1):
                for y_i in range(max(0, y - 1), min(cols -1,y + 1) +1):
                    total += image[x_i,y_i]
            Filter_image[x,y] = total / 9

    return Filter_image
```

In [7]:

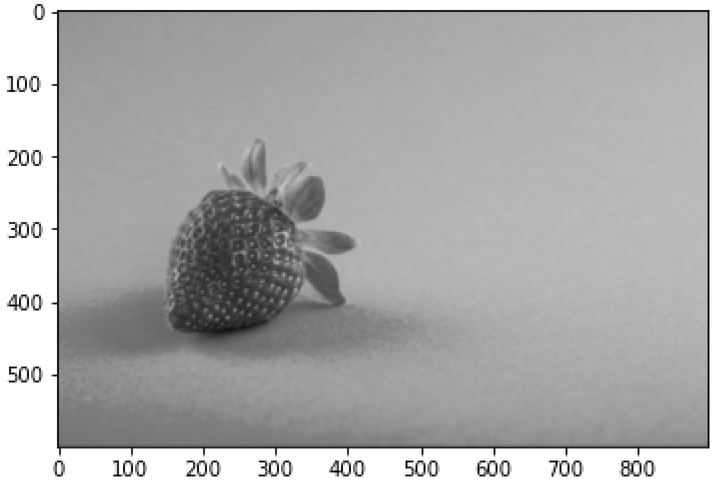
```
image = cv.imread( "./images for PA1/berry.jpg" , cv.IMREAD_GRAYSCALE )

final_image = BoxFilterGray3(image)

plt.imshow(image,cmap = "gray", vmin=0,vmax=255)
```

Out[7]:

```
<matplotlib.image.AxesImage at 0x29a3a84cc10>
```



In [ ]:

```
```