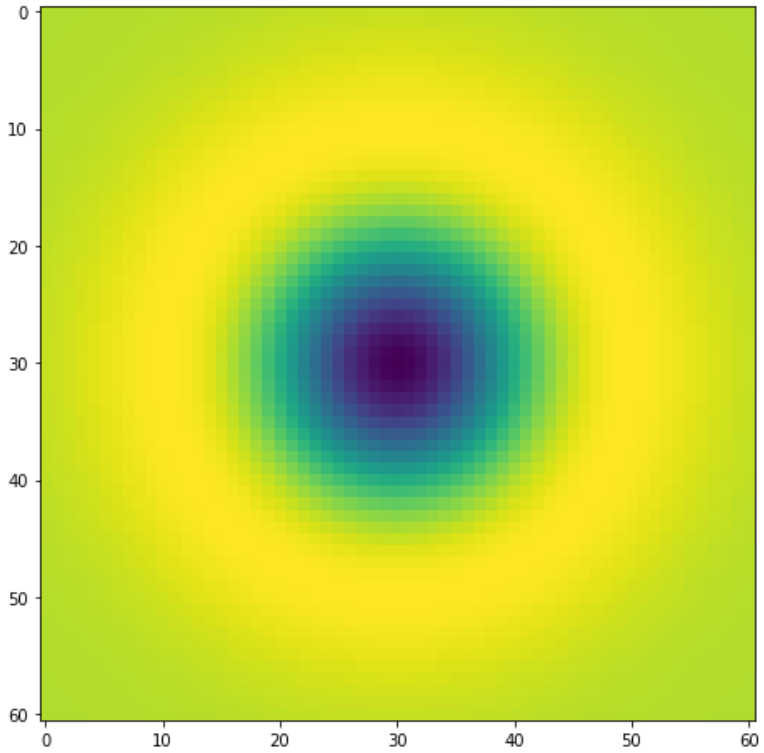
In [1]:
```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from skimage.feature import peak_local_max
%matplotlib inline
```

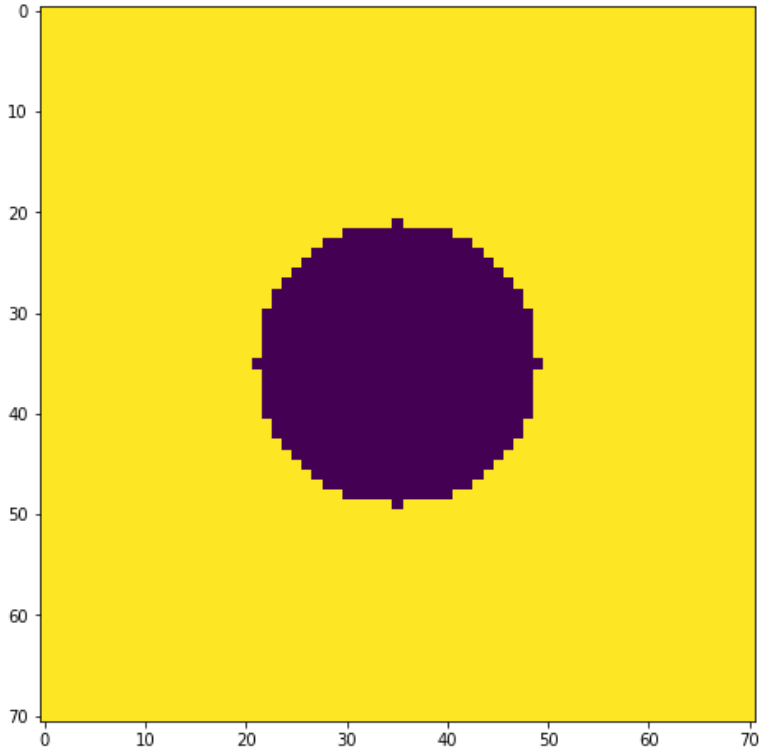
Question 1

In [11]:
```
sigma = 10
hw = 3 * sigma

X,Y = np.meshgrid(np.arange(-hw, hw + 1, 1), np.arange(-hw, hw + 1, 1))

l_o_g = 1/(2 * np.pi * sigma ** 2) * (X ** 2 / (sigma ** 2)  + Y ** 2 / (sigma ** 2) - 2) * np.exp(-(X**2 +

_, ax = plt.subplots(1,1, figsize=(8,8))
ax.imshow(l_o_g)
plt.show()
```

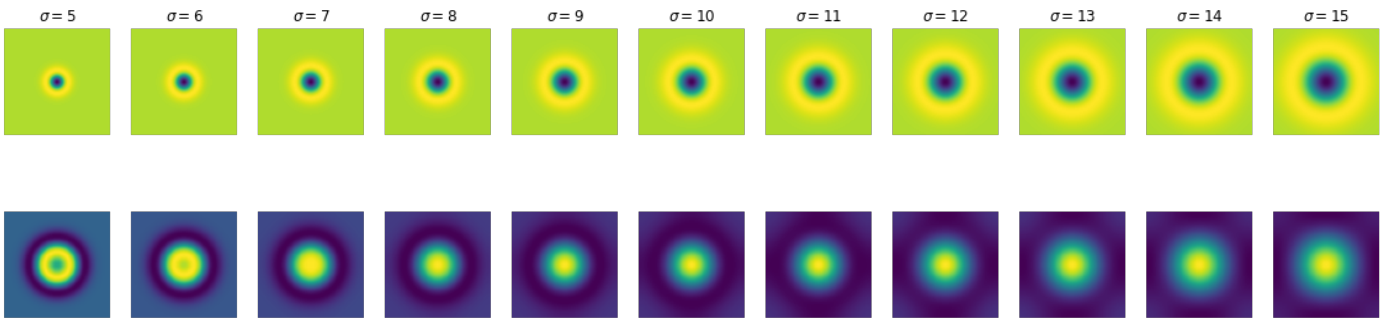


Question 2

In [13]:
```
w, h = 71, 71
hw, hh = w//2, h//2

f = np.ones((h,w), dtype = np.float32) * 255
X,Y = np.meshgrid(np.arange(-hh, hh+1, 1), np.arange(-hw, hw+1))



r = w //5
f *= X ** 2 + Y ** 2 > r ** 2

_, ax = plt.subplots(1,1, figsize=(8,8))
ax.imshow(f)
plt.show()
```



In [4]:
```
s = 11
fig, ax = plt.subplots(2,s, figsize = (20, 5))
scale_space = np.empty((h,w,s), dtype = np.float32)

sigmas = np.arange(5, 16, 1)

for i , sigma in enumerate(sigmas):
    log_hw = 3 * np.max(sigmas)
    X,Y = np.meshgrid(np.arange(-log_hw, log_hw + 1, 1), np.arange(-log_hw, log_hw + 1, 1))
    l_o_g = 1/(2 * np.pi * sigma ** 2) * (X ** 2 / (sigma ** 2)  + Y ** 2 / (sigma ** 2) - 2) * np.exp(-(X**
    f_log = cv.filter2D(f, -1, l_o_g)
    scale_space[:,:,i] = f_log

    ax[0, i].imshow(l_o_g)
    ax[0, i].axis('off')
    ax[0, i].set_title(r'$\sigma = {}$'.format(sigma))
    ax[1, i].imshow(f_log)
    ax[1, i].axis('off')


indicies = np.unravel_index(np.argmax(scale_space, axis = None), scale_space.shape)
```

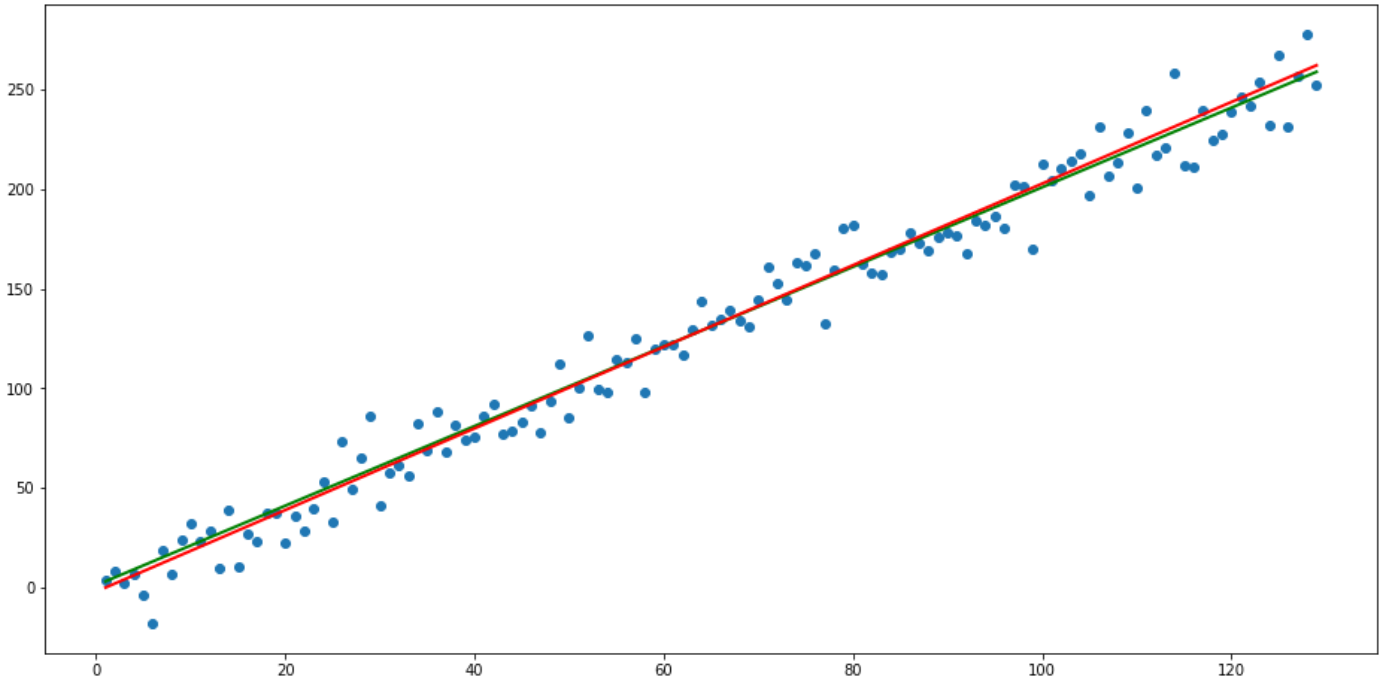


Question 3

In [22]:
```
#SIFT MACHING

image1 = cv.imread('img1.ppm', cv.IMREAD_GRAYSCALE)
image2 = cv.imread('img2.ppm', cv.IMREAD_GRAYSCALE)


sift = cv.xfeatures2d.SIFT_create()

kp1, desc1 = sift.detectAndCompute(image1 ,None)
kp2, desc2 = sift.detectAndCompute(image2,None)

bf = cv.BFMatcher(cv.NORM_L1, crossCheck=True)

matches = bf.match(desc1,desc2)
matches = sorted(matches, key = lambda x:x.distance)
connections = cv.drawMatches(image1 , kp1, image2, kp2, matches[:50], image2, flags=2)

_, ax = plt.subplots(1,1, figsize=(16,8))
plt.imshow(connections)
plt.show()
```

Question 4

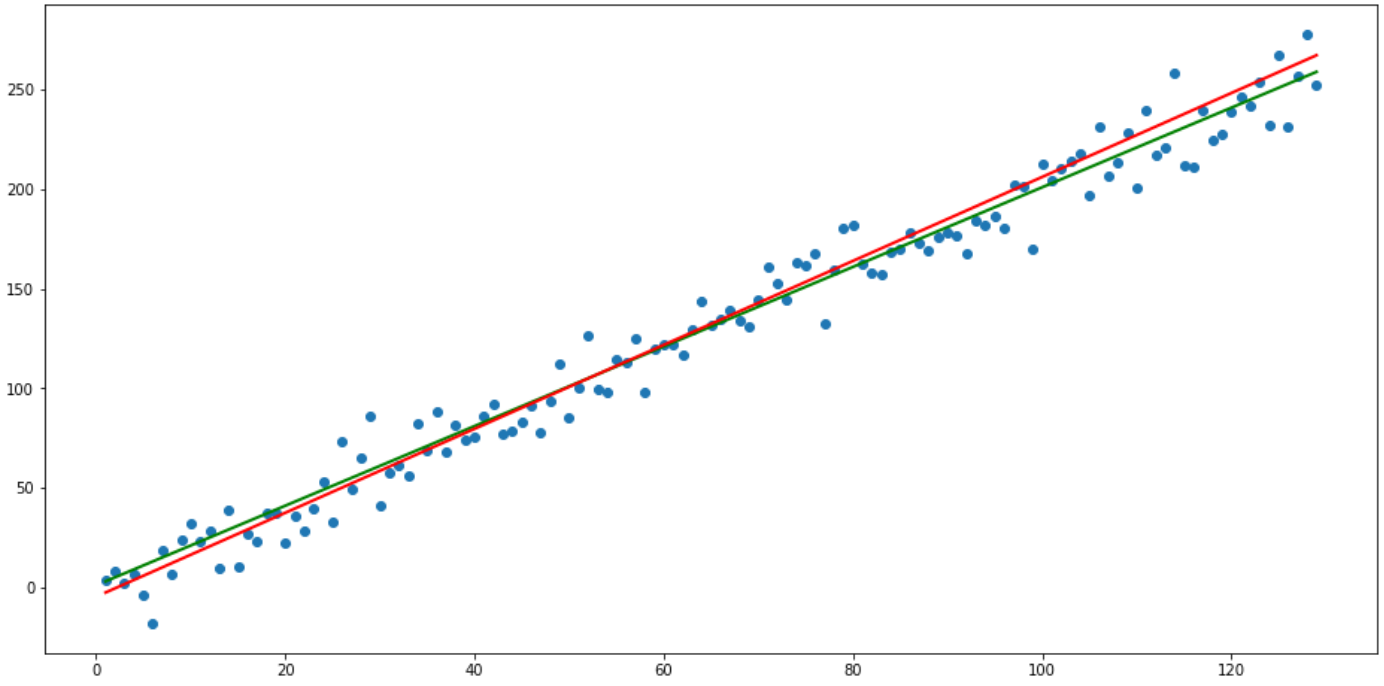
In [17]:
```
# Least squares line fitting

m = 2 # Line equation : y = m*x + c . m i s the s l o p e . c i s the i n t e r c e p t .
c = 1
x = np.arange (1 , 130 , 1)
sigma = 12
np.random.seed(45)
noise = sigma * np.random.randn(len(x))
o = np.zeros(x.shape)

y = m*x +c +noise +0


n = len(x)
X = np.concatenate([x.reshape(n,1), np.ones((n, 1))], axis = 1)
B = np.linalg.pinv(X.T @ X) @ X.T  @ y
mstar = B[0]
cstar = B[1]


_, ax = plt.subplots(1,1, figsize=(16,8))

ax.plot(x,y,'o', label='noise points')
ax.plot([x[0], x[-1]], [m*x[0] + c, m*x[-1] + c], color = 'g', linewidth = 2, label="True Line")
ax.plot([x[0], x[-1]], [mstar*x[0] + cstar, mstar*x[-1] + c], color = 'r', linewidth = 2, label="Estimated L

plt.show()
```



Question 5

In [18]:

```
# Total Least squares line fitting

m = 2 # Line equation : y = m*x + c . m i s the s l o p e . c i s the i n t e r c e p t .
c = 1
x = np.arange (1 , 130 , 1)
sigma = 12
np.random.seed(45)
noise = sigma * np.random.randn(len(x))
o = np.zeros(x.shape)

y = m*x +c +noise + o

n = len(x)

u11 = np.sum((x - np.mean(x)) ** 2)
u12 = np.sum((x - np.mean(x)) * (y - np.mean(y)))
u21 = u12
u22 = np.sum((y - np.mean(y)) ** 2)

U = np.array([[u11, u12], [u21, u22]])
W,V = np.linalg.eig(U)

ev_correspointing_to_smalest_ev = V[:, np.argmin(W)]

a = ev_correspointing_to_smalest_ev[0]
b = ev_correspointing_to_smalest_ev[1]
d = a*np.mean(x) + b*np.mean(y)


mstar = -a/b
cstar = d/b

_, ax = plt.subplots(1,1, figsize=(16,8))

ax.plot(x,y,'o', label='noise points')
ax.plot([x[0], x[-1]], [m*x[0] + c, m*x[-1] + c], color = 'g', linewidth = 2, label="True Line")
ax.plot([x[0], x[-1]], [mstar*x[0] + cstar, mstar*x[-1] + c], color = 'r', linewidth = 2, label="Estimated L

plt.show()
```



In [ ]: