In [1]:

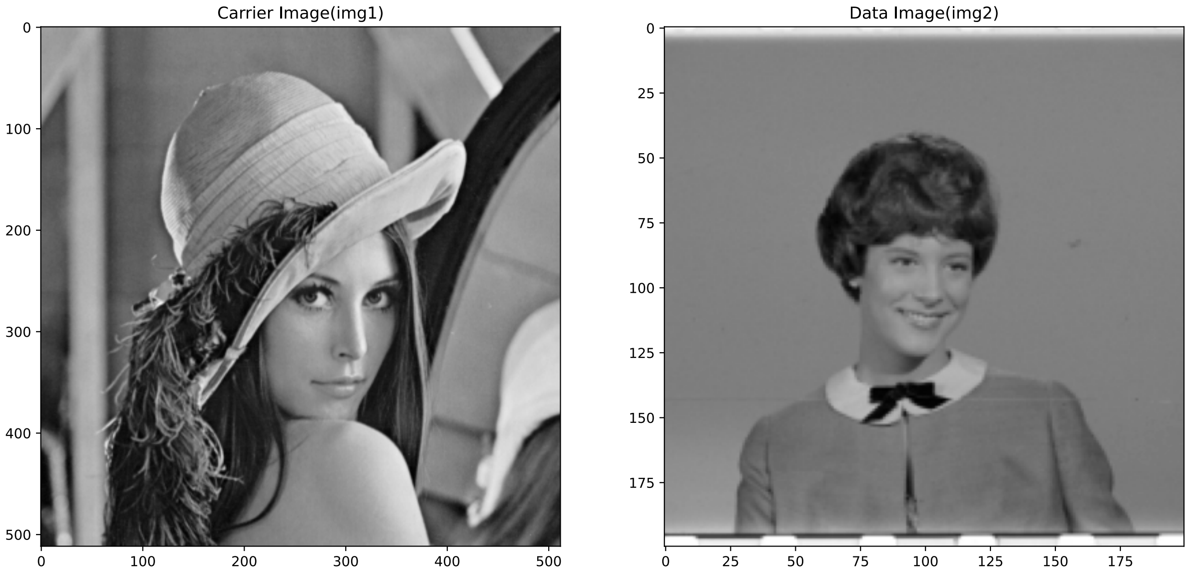
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
import math
from skimage import metrics
```

In [2]:

```
img1= cv2.imread('Lenna.png',0)
img2 =cv2.imread('4.1.03.tiff',0)
# img1=cv2.resize(img1,(5,5))
img2=cv2.resize(img2,(200,200))
# print(img1)
# print(img2)
plt.figure(figsize=(15,15))
plt.subplot(1,2,1)
plt.imshow(img1,'gray')
plt.title('Carrier Image(img1)')
plt.subplot(1,2,2)
plt.imshow(img2,'gray')
plt.title('Data Image(img2)')
```

Out[2]:

```
Text(0.5, 1.0, 'Data Image(img2)')
```



In [3]:

```
img2_flat=img2.flatten()
msg=""
for i in range (img2.shape[0]*img2.shape[1]):
    msg+=format(img2_flat[i],'08b')
msg_copy=msg+'0000000'

# print(msg)
# print(msg_copy)
```

In [4]:

```
diff_arr=[]
flat=[]
flat_copy=[]
for i in range(img1.shape[0]):
    if(i%2==0):
        for j in range (img1.shape[1]):
            flat.append(img1[i][j])
            flat_copy.append(img1[i][j])
    else:
        for j in range (img1.shape[1]-1,-1,-1):
            flat.append(img1[i][j])
            flat_copy.append(img1[i][j])

t=int(len(flat)/2)
for i in range(t):
    diff_arr.append(abs(int(flat[(i*2)+1])-int(flat[(i*2)])))
# print(flat)
# print(diff_arr)
```

In [5]:

```
#Encryption
num=0
i=0
while i<len(diff_arr) and num<len(msg):
    if(diff_arr[i]<8):
        temp=int(msg_copy[num:num+3],2)
        d=temp
        num+=3

    elif(diff_arr[i]<16 and diff_arr[i]>7):
        temp=int(msg_copy[num:num+3],2)
        d=8+temp
        num+=3
    elif(diff_arr[i]>15 and diff_arr[i]<32):
        temp=int(msg_copy[num:num+4],2)
        d=16+temp
        num+=4
    elif(diff_arr[i]>31 and diff_arr[i]<64):
        temp=int(msg_copy[num:num+5],2)
        d=32+temp
        num+=5
    elif(diff_arr[i]>63 and diff_arr[i]<128):
        temp=int(msg_copy[num:num+6],2)
        d=64+temp
        num+=6
    elif(diff_arr[i]>127 and diff_arr[i]<256):
        temp=int(msg_copy[num:num+7],2)
        d=128+temp
        num+=7

    if(flat[2*i]>=flat[2*i +1] and d>diff_arr[i]):
        flat[2*i]+=math.ceil(abs(d-diff_arr[i])/2)
        flat[2*i +1]-=math.floor(abs(d-diff_arr[i])/2)
    elif(flat[2*i]<flat[2*i +1] and d>diff_arr[i]):
        flat[2*i]-=math.ceil(abs(d-diff_arr[i])/2)
        flat[2*i +1]+=math.floor(abs(d-diff_arr[i])/2)
    elif(flat[2*i]>=flat[2*i +1] and d<=diff_arr[i]):
        flat[2*i]-=math.ceil(abs(d-diff_arr[i])/2)
        flat[2*i +1]+=math.floor(abs(d-diff_arr[i])/2)
    elif(flat[2*i]<flat[2*i +1] and d<=diff_arr[i]):
        flat[2*i]+=math.ceil(abs(d-diff_arr[i])/2)
        flat[2*i +1]-=math.floor(abs(d-diff_arr[i])/2)


    i+=1
```

In [6]:

```
carrier_img= np.zeros((img1.shape[0],img1.shape[1]),np.uint8)
index=0
for i in range(img1.shape[0]):
    if(i%2==0):
        for j in range (img1.shape[1]):
            carrier_img[i][j]=flat[index]
            index+=1
    else:
        for j in range (img1.shape[1]-1,-1,-1):
            carrier_img[i][j]=flat[index]
            index+=1
print(carrier_img)
```

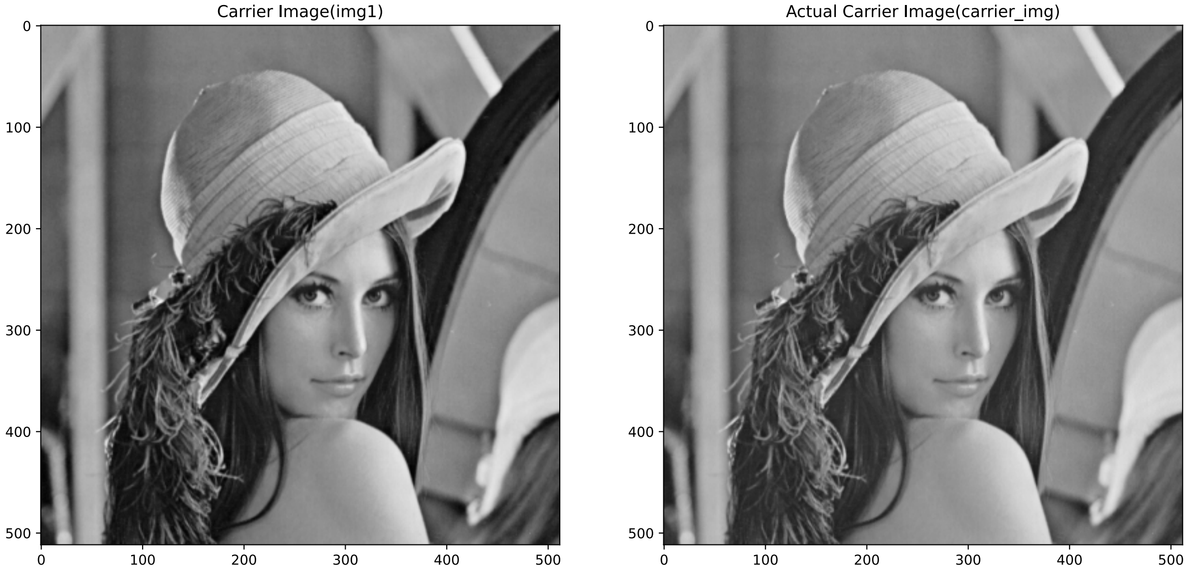
```
[[173 166 171 ... 174 165 136]
 [166 173 169 ... 173 165 134]
 [173 166 167 ... 173 158 141]
 ...
 [ 53  53  59 ... 115 112 114]
 [ 53  53  64 ... 117 118 122]
 [ 53  53  64 ... 117 118 122]]
```

In [7]:

```
plt.figure(figsize=(15,15))
plt.subplot(1,2,1)
plt.imshow(img1,'gray')
plt.title('Carrier Image(img1)')
plt.subplot(1,2,2)
plt.imshow(carrier_img,'gray')
plt.title('Actual Carrier Image(carrier_img)')
```

Out[7]:

```
Text(0.5, 1.0, 'Actual Carrier Image(carrier_img)')
```



Carrier Image(img1)

Actual Carrier Image(carrier_img)

In [8]:

```
#Decryption

flat_carr=[]
for i in range(img1.shape[0]):
    if(i%2==0):
        for j in range (img1.shape[1]):
            flat_carr.append(carrier_img[i][j])
    else:
        for j in range (img1.shape[1]-1,-1,-1):
            flat_carr.append(carrier_img[i][j])

# print(flat_carr)
```

In [9]:

```
msg_dec=""
i=0
while i<len(flat_carr) and len(msg_dec)<=len(msg):
    temp=abs(int(flat_carr[i+1])-int(flat_carr[i]))
    if(temp<8):
        s=format(temp,'03b')
        msg_dec+=s
    if(temp<16 and temp>7):
        temp-=8
        s=format(temp,'03b')
        msg_dec+=s
    elif(temp>15 and temp<32):
        temp-=16
        s=format(temp,'04b')
        msg_dec+=s
    elif(temp>31 and temp<64):
        temp-=32
        s=format(temp,'05b')
        msg_dec+=s
    elif(temp>63 and temp<128):
        temp-=64
        s=format(temp,'06b')
        msg_dec+=s
    elif(temp>127 and temp<256):
        temp-=128
        s=format(temp,'07b')
        msg_dec+=s

    i+=2

# print(msg_dec)
```

In [10]:

```
resultant_data_img= np.zeros((img2.shape[0],img2.shape[1]),np.uint8)
num=0
for i in range(img2.shape[0]):
    for j in range(img2.shape[1]):
        resultant_data_img[i][j]=int(msg_dec[num:num+8],2)
        num+=8
print(resultant_data_img)
```

```
[[254 254 254 ... 248 249 249]
 [241 241 240 ... 222 222 222]
 [227 227 228 ... 218 218 217]
 ...
 [254 254 255 ... 255 255 255]
 [254 255 255 ... 255 255 254]
 [254 254 254 ... 255 255 255]]
```

In [11]:

```
print("Carrier Image: ")
print("MSE: ",metrics.mean_squared_error(img1,carrier_img))
print("PSNR: ",metrics.peak_signal_noise_ratio(img1,carrier_img))
print("SSIM: ",metrics.structural_similarity(img1,carrier_img))
print("Data Image: ")
print("MSE: ",metrics.mean_squared_error(img2,resultant_data_img))
print("PSNR: ",metrics.peak_signal_noise_ratio(img2,resultant_data_img))
print("SSIM: ",metrics.structural_similarity(img2,resultant_data_img))

plt.figure(figsize=(20,20))
plt.subplot(1,4,1)
plt.imshow(img1,'gray')
plt.title('Carrier Image without hidden data')
plt.subplot(1,4,2)
plt.imshow(carrier_img,'gray')
plt.title('Carrier Image with hidden data')
plt.subplot(1,4,3)
plt.imshow(img2,'gray')
plt.title('Data image before Encryption')
plt.subplot(1,4,4)
plt.imshow(resultant_data_img,'gray')
plt.title('Data image after Encryption')
```

```
Carrier Image:
MSE:  3.353443145751953
PSNR:  42.875894131539674
SSIM:  0.9828129712766597
Data Image:
MSE:  0.0
PSNR:  inf
SSIM:  1.0
```

Out[11]:

```
Text(0.5, 1.0, 'Data image after Encryption')
```