Derivative of Matrix

import library

In [ ]:
```
import numpy as np
import matplotlib.image as img
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import util
```

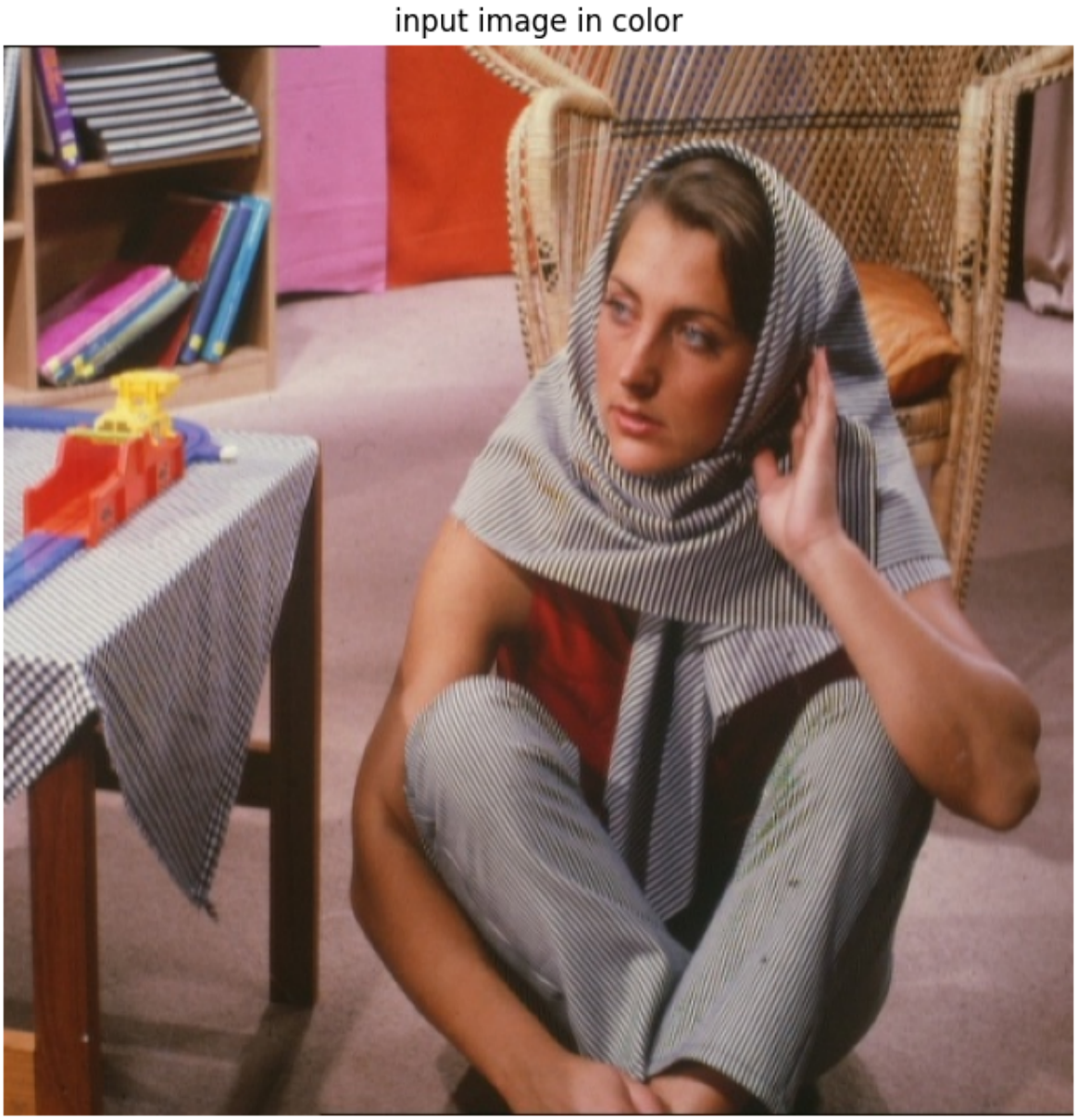
load image

In [ ]:
```
I_color = img.imread('03_image.jpeg')
```

visualize the input image in color

In [ ]:
```
plt.figure(figsize=(8,8))
plt.imshow(I_color)
plt.title('input image in color')
plt.axis('off')
```

Out[ ]:
```
(-0.5, 511.5, 511.5, -0.5)
```



check the size of image

In [ ]:
```
dim_height  = I_color.shape[0]
dim_width   = I_color.shape[1]
dim_channel = I_color.shape[2]
print('height=', dim_height, ', width=', dim_width, ', channel=', d
im_channel)
```

```
height= 512 , width= 512 , channel= 3
```

convert the input image into gray scale by taking the average over the channel

In [ ]:
```
I_gray = util.convert_color_to_gray(I_color)
```

normalize the gray-scale image into the range [0, 1]

In [ ]:
```
I_norm = util.normalize_01(I_gray)
```

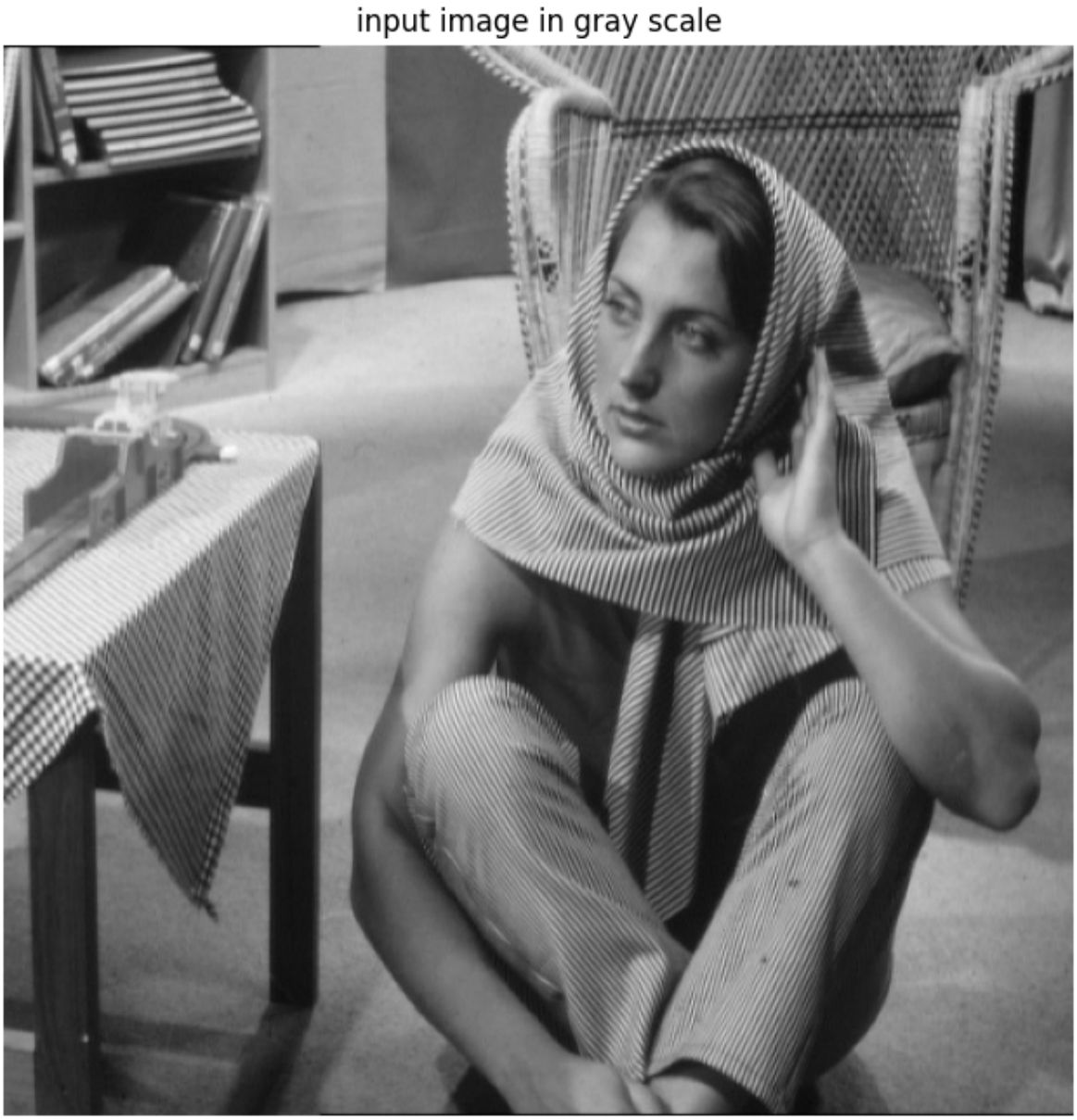
In [ ]:
```
val_min = np.min(I_norm)
val_max = np.max(I_norm)
print('min=', val_min, ', max=', val_max)
```

```
min= 0.0 , max= 1.0
```

visualize the input image in gray

In [ ]:
```
def plot_01():
    plt.figure(figsize=(8,8))
    plt.imshow(I_norm, cmap='gray', vmin=0, vmax=1)
    plt.title('input image in gray scale')
    plt.axis('off')
    plt.show()
    print('sum =', I_norm.sum())
```

In [ ]:
```
plot_01()
```



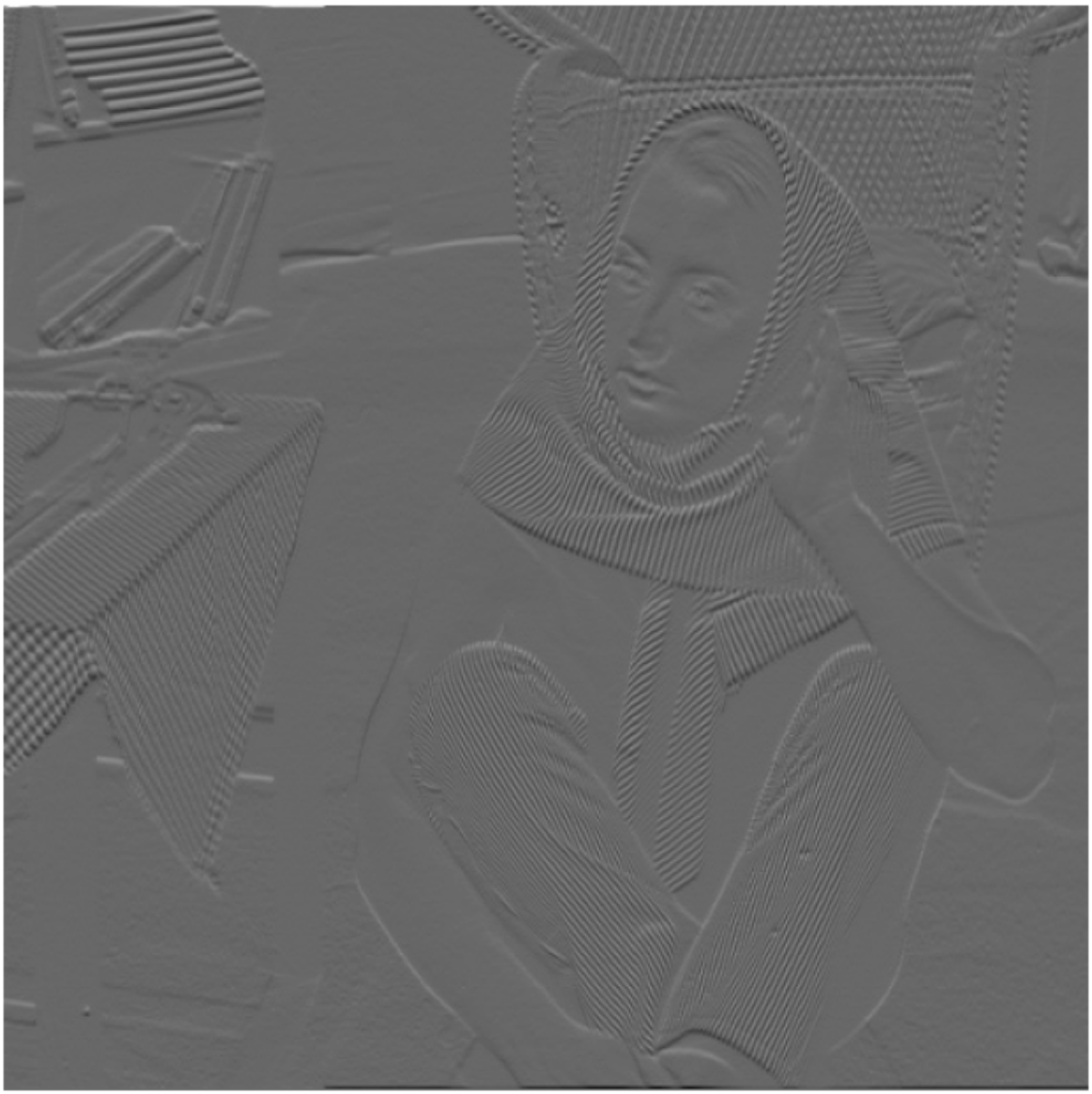

```
sum = 115933.95704697986
```

compute the first-order derivative using the forward difference scheme in row-direction

In [ ]:
```
I_out2 = util.compute_derivative_first_row_forward(I_norm)
```

In [ ]:
```
def plot_02():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out2, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out2.sum())
```

In [ ]:
```
plot_02()
```



```
sum = -125.03176733780757
```

compute the first-order derivative using the forward difference scheme in column-direction

In [ ]:
```
I_out3 = util.compute_derivative_first_col_forward(I_norm)
```

In [ ]:
```
def plot_03():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out3, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out3.sum())
```

In [ ]:
```
plot_03()
```

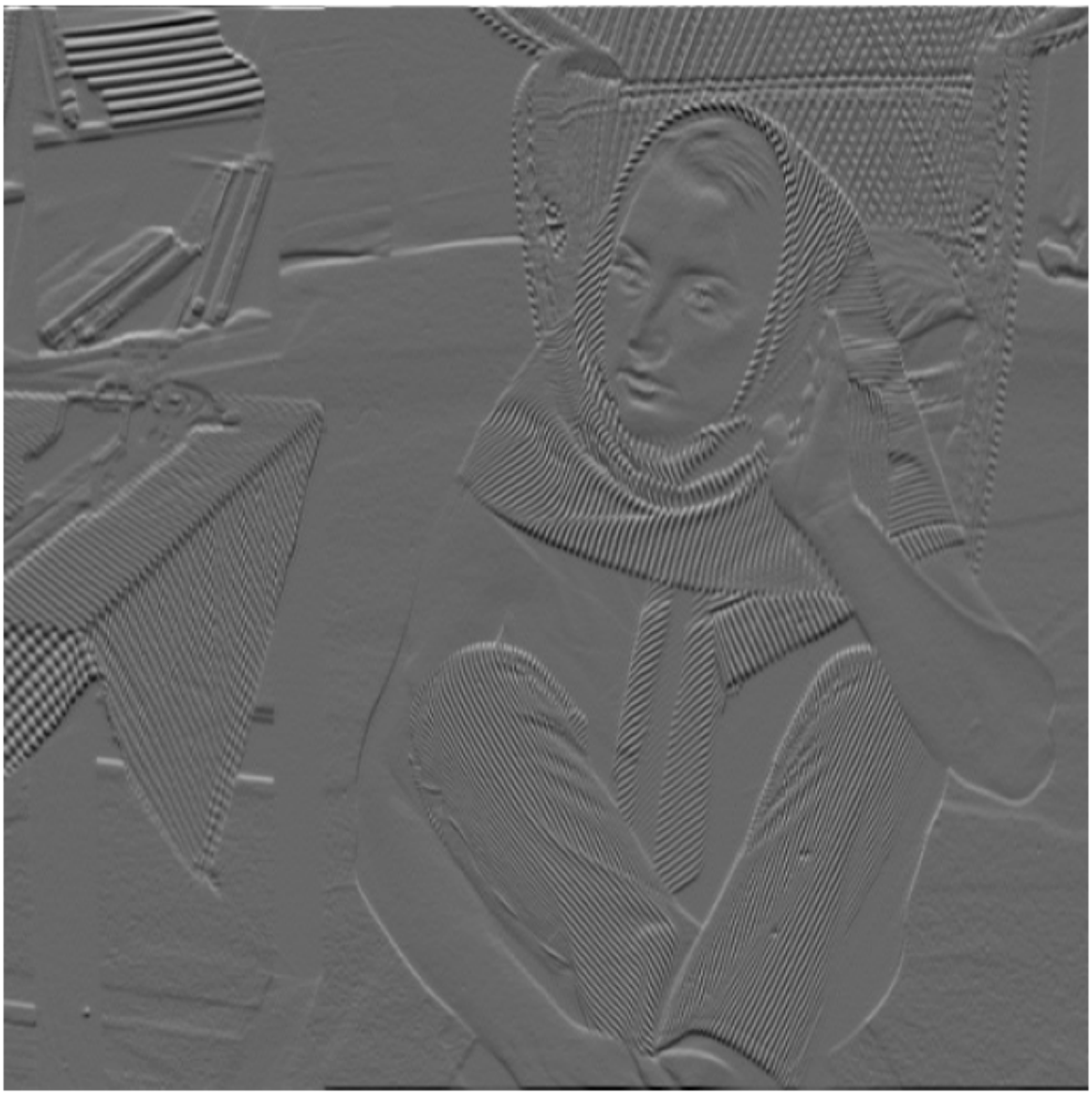
```
sum = -37.925727069351225
```

compute the first-order derivative using the central difference scheme in row-direction

In [ ]:
```
I_out4 = util.compute_derivative_first_row_central(I_norm)
```

In [ ]:
```
def plot_04():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out4, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out4.sum())
```

In [ ]:
```
plot_04()
```



```
sum = -62.515883668903726
```

compute the first-order derivative using the central difference scheme in column-direction

In [ ]:
```
I_out5 = util.compute_derivative_first_col_central(I_norm)
```

In [ ]:
```
def plot_05():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out5, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out5.sum())
```

In [ ]:
```
plot_05()
```

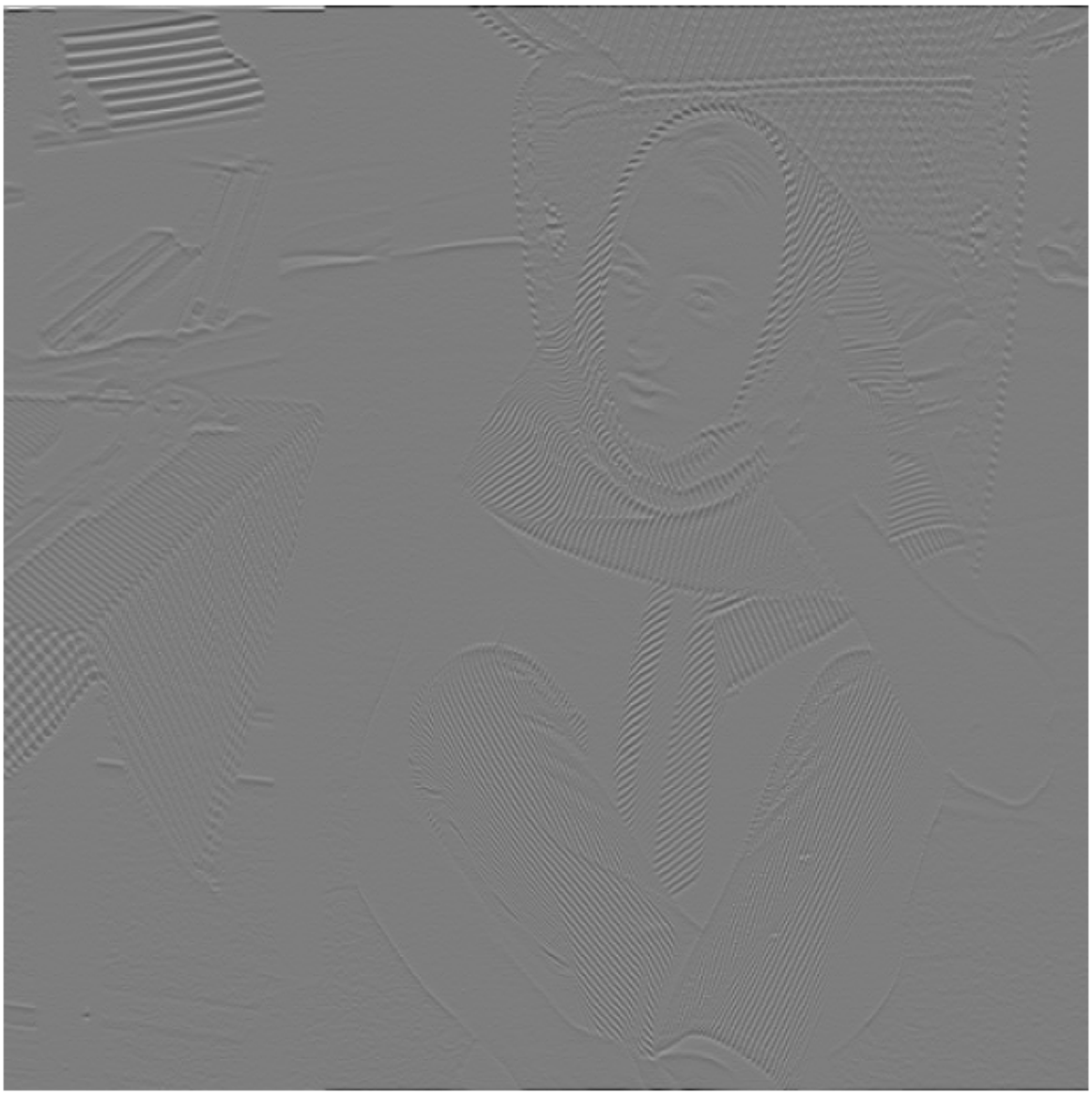
```
sum = -18.962863534675616
```

compute the second-order derivative in row-direction

In [ ]:
```
I_out6 = util.compute_derivative_second_row(I_norm)
```

In [ ]:
```
def plot_06():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out6, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out6.sum())
```

In [ ]:
```
plot_06()
```



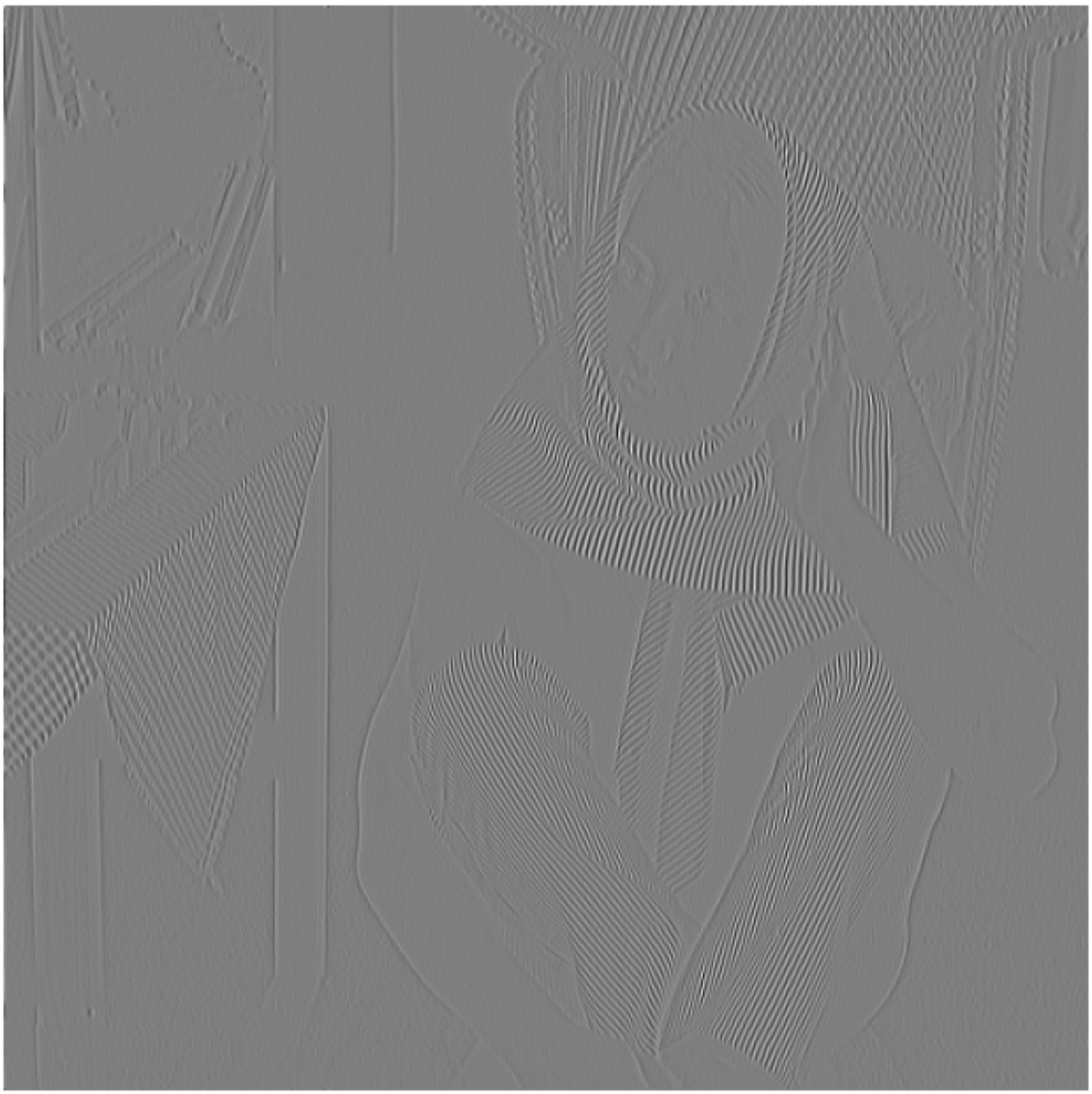
```
sum = -230.53557046979867
```

compute the second-order derivative in column-direction

In [ ]:
```
I_out7 = util.compute_derivative_second_col(I_norm)
```

In [ ]:
```
def plot_07():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out7, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out7.sum())
```

In [ ]:
```
plot_07()
```



```
sum = -35.123042505592856
```

compute the L_2^2-norm of the gradient using the central difference scheme

In [ ]:
```
I_out8 = util.compute_norm_gradient_central(I_norm)
```

In [ ]:
```
def plot_08():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out8, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out8.sum())
```

In [ ]:
```
plot_08()
```

```
sum = 20624.38376528384
```

compute the Laplacian

In [ ]:
```
def compute_laplacian(I):
    # ================================================
    # complete the code
    # ================================================
    I_pad       = np.pad(I, (1, 1), 'symmetric')
    I_der1      = I_pad[2:, 1:-1]
    I_der2      = I_pad[:-2, 1:-1]
    I_der3      = I_pad[1:-1, 2:]
    I_der4      = I_pad[1:-1, :-2]
    I_laplace   = I_der1 + I_der2 + I_der3 + I_der4 - 4.0 * I
    return I_laplace
```

In [ ]:
```
I_out9 = util.compute_laplacian(I_norm)
```

In [ ]:
```
def plot_09():
    plt.figure(figsize=(8,8))
    plt.imshow(I_out9, cmap='gray')
    plt.axis('off')
    plt.show()
    print('sum =', I_out9.sum())
```

In [ ]:
```
plot_09()
```

```
sum = -3.930189507173054e-14
```

results

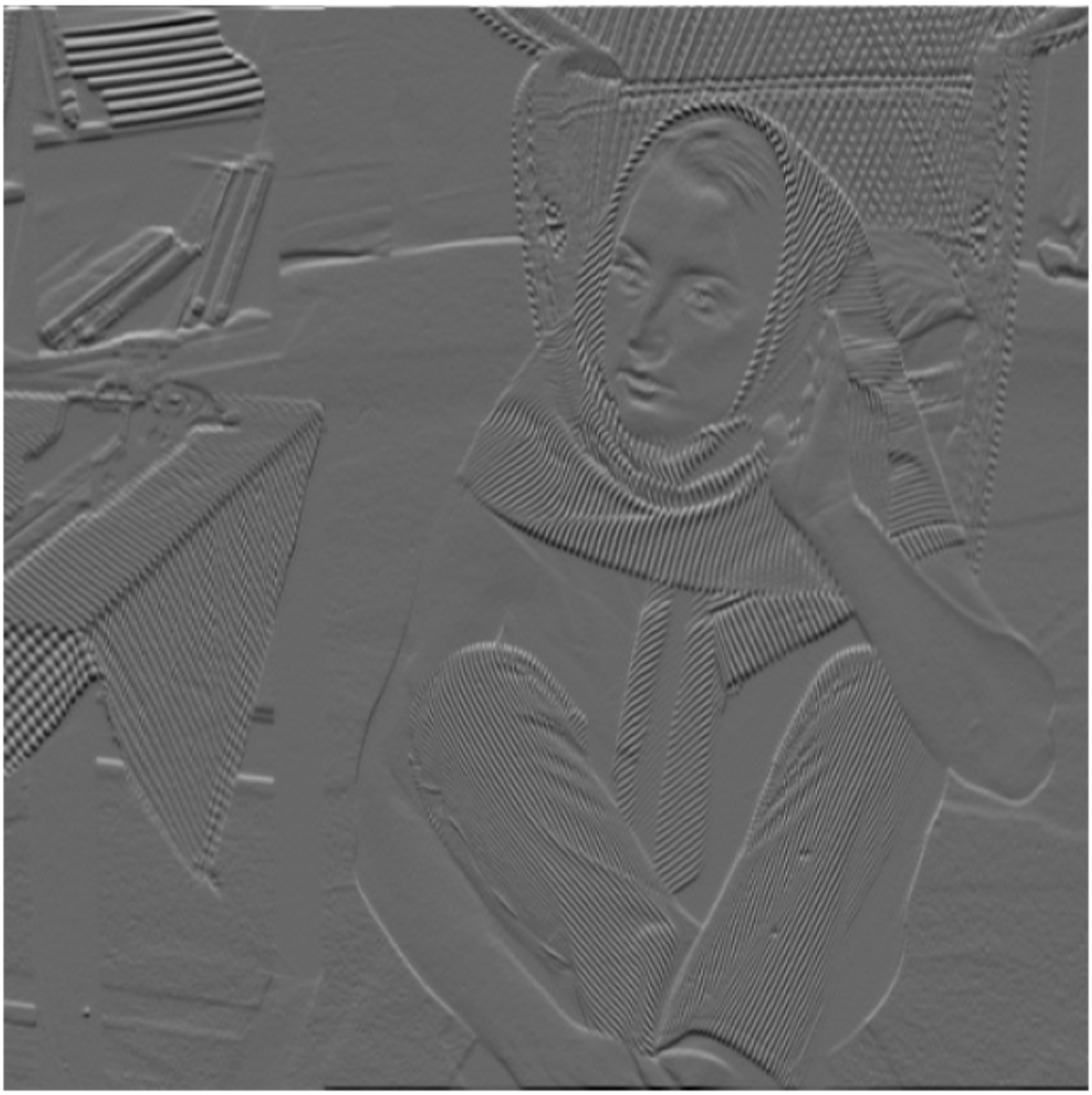
In [ ]:
```
number_result = 9

for i in range(number_result):
    title           = '# RESULT # {:02d}'.format(i+1)
    name_function   = 'plot_{:02d}()'.format(i+1)

    print('')
    print('##########################################################
########################')
    print(title)
    print('##########################################################
########################')
    print('')

    eval(name_function)
```

```
##########################################################################
##############
# RESULT # 01
##########################################################################
##############
```


```
sum = 115933.95704697986

##################################################################
##############
# RESULT # 02
##################################################################
##############
```

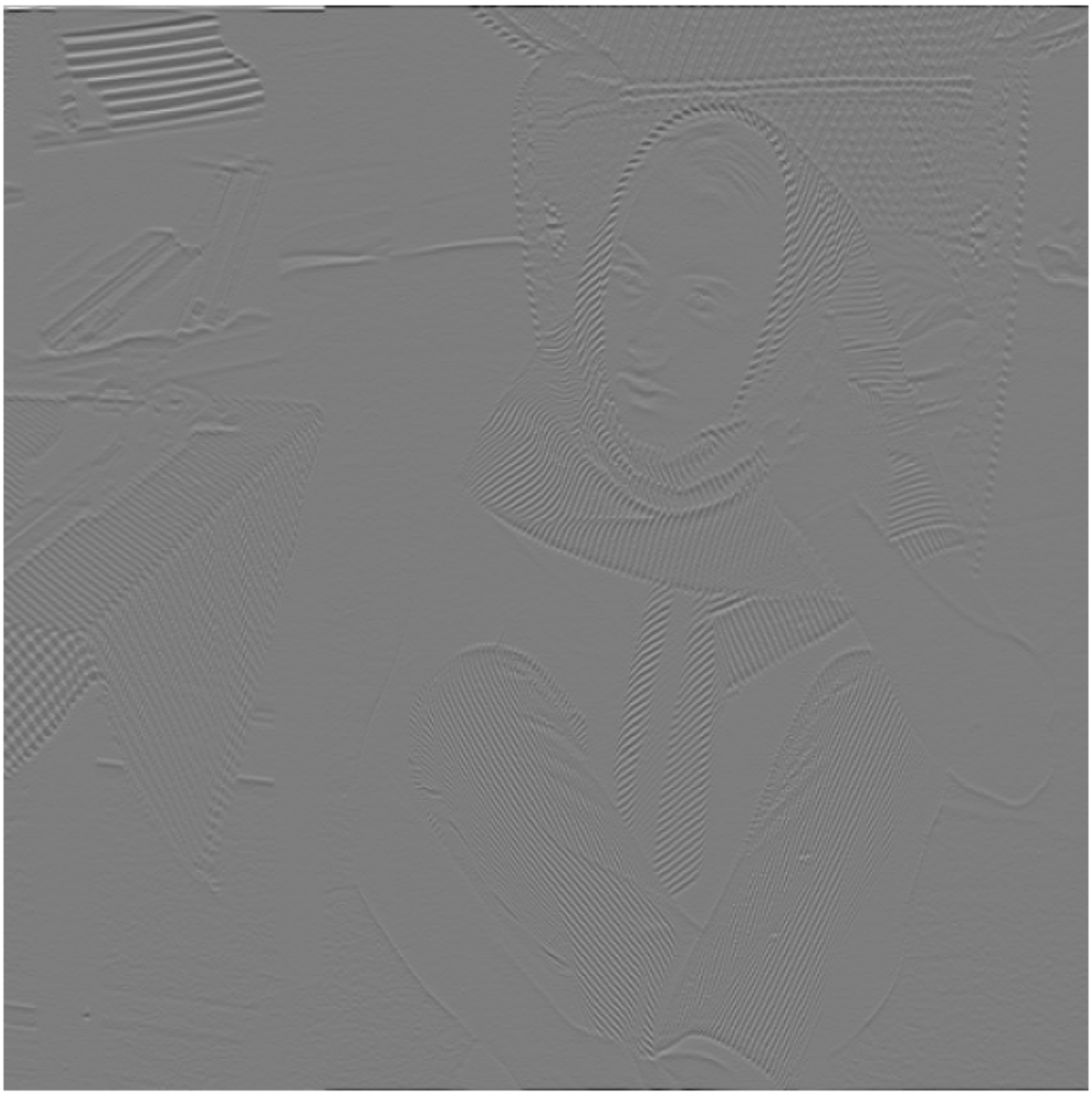
```
sum = -125.03176733780757

##################################################################
##############
# RESULT # 03
##################################################################
##############
```

```
sum = -37.925727069351225

##################################################################
##############
# RESULT # 04
##################################################################
##############
```

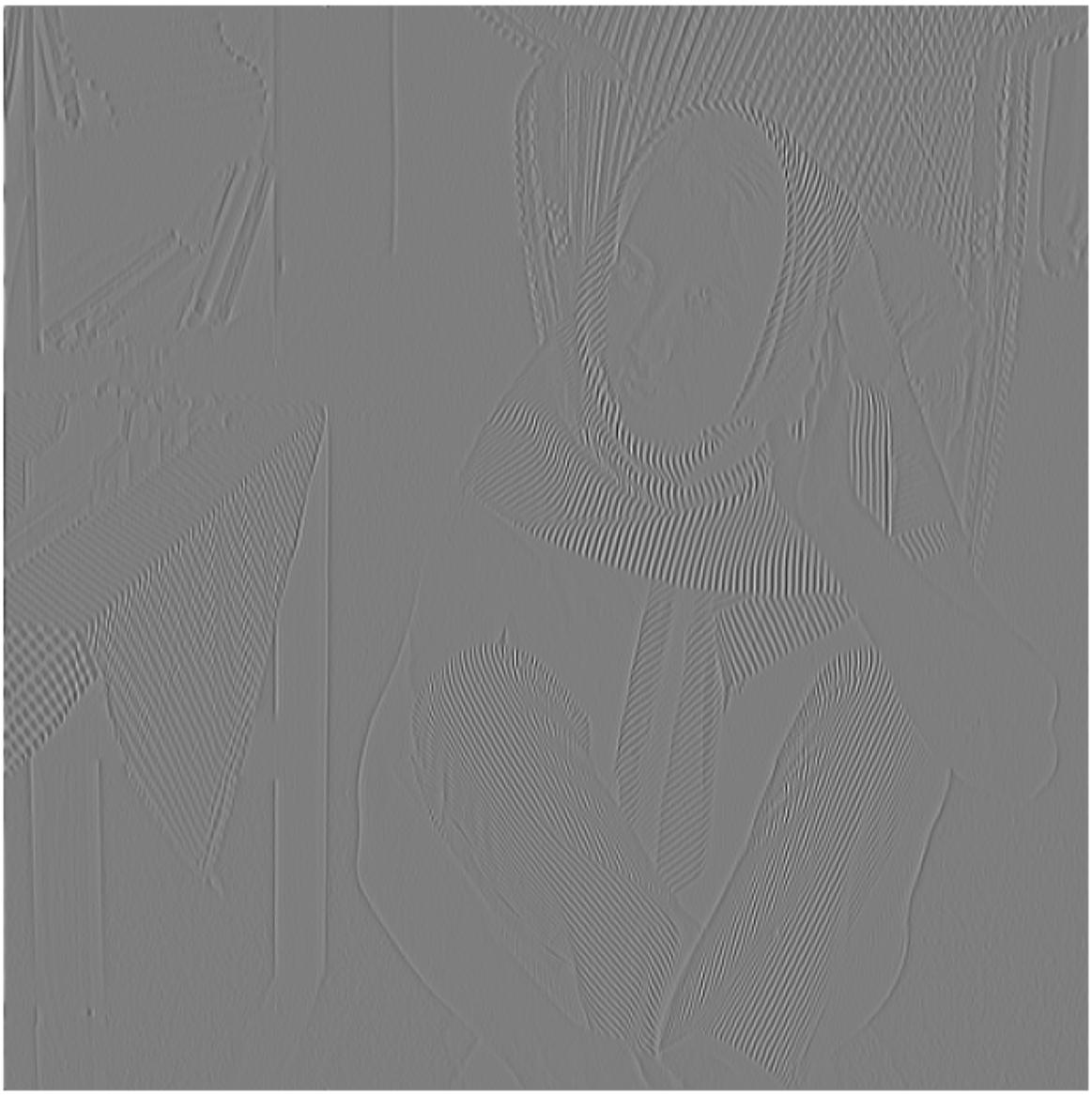


```
sum = -62.515883668903726

##################################################################
##############
# RESULT # 05
##################################################################
##############
```

```
sum = -18.962863534675616

##################################################################
##############
# RESULT # 06
##################################################################
##############
```



```
sum = -230.53557046979867

##################################################################
##############
# RESULT # 07
##################################################################
##############
```



```
sum = -35.123042505592856

##################################################################
##############
# RESULT # 08
##################################################################
##############
```

```
sum = 20624.38376528384

##################################################################
##############
# RESULT # 09
##################################################################
##############
```

```
sum = -3.930189507173054e-14
```

In [ ]: