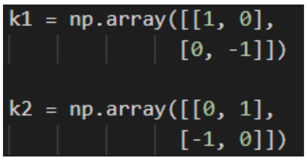
Computer Version HW9

You are to implement following edge detectors with thresholds :
(a) Robert's Operator: 12
(b) Prewitt's Edge Detector: 24
(c) Sobel's Edge Detector: 38
(d) Frei and Chen's Gradient Operator: 30
(e) Kirsch's Compass Operator: 135
(f) Robinson's Compass Operator: 43
(g) Nevatia-Babu 5x5 Operator: 12500

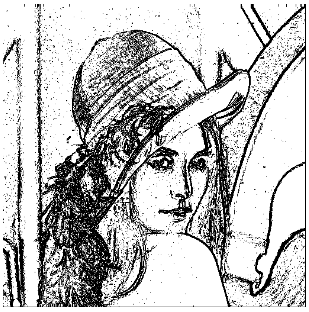
根據講義中提供的算法，實現以上各種 edge detectors，結果如下：

(a) Robert's Operator:

Threshold = 12

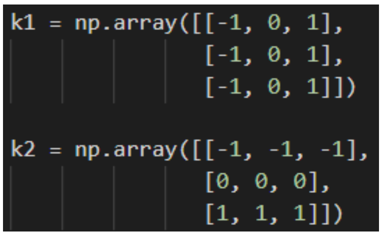


```
k1 = np.array([[1, 0],
               [0, -1]])

k2 = np.array([[0, 1],
               [-1, 0]])
```

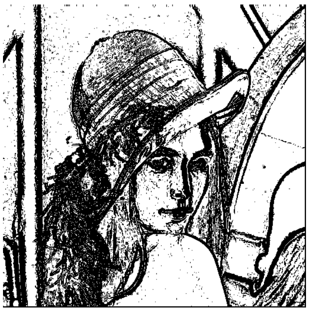


(b) Prewitt's Edge Detector:

Threshold = 24

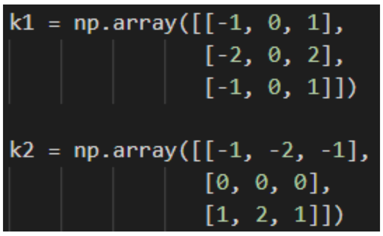


```
k1 = np.array([[-1, 0, 1],
               [-1, 0, 1],
               [-1, 0, 1]])

k2 = np.array([[-1, -1, -1],
               [0, 0, 0],
               [1, 1, 1]])
```

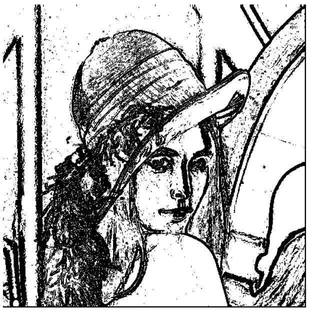


(c) Sobel's Edge Detector:

Threshold = 38

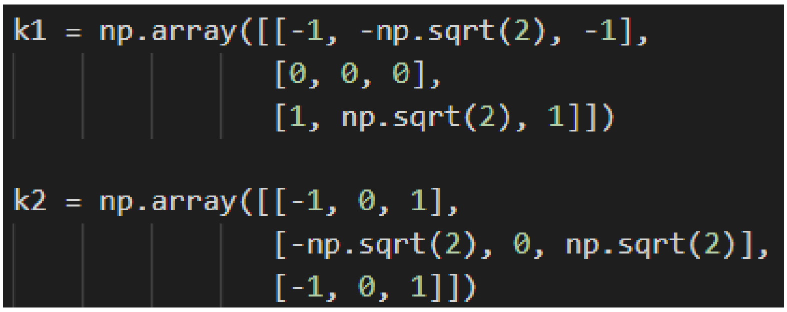


```
k1 = np.array([[-1, 0, 1],
               [-2, 0, 2],
               [-1, 0, 1]])

k2 = np.array([[-1, -2, -1],
               [0, 0, 0],
               [1, 2, 1]])
```

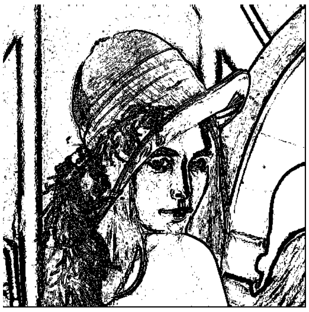


(d) Frei and Chen's Gradient Operator:

Threshold = 30

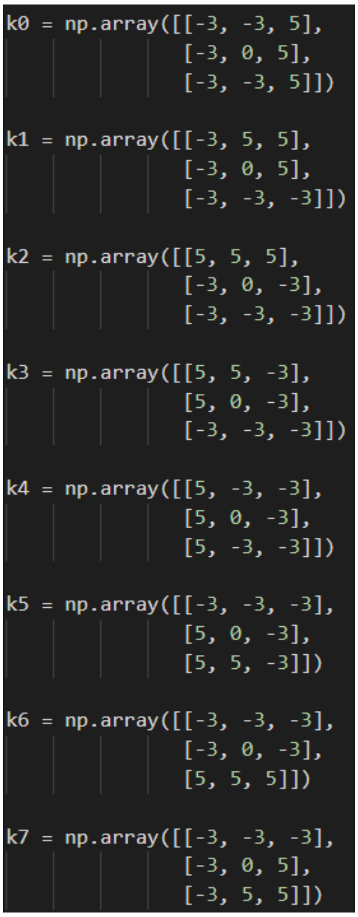


```
k1 = np.array([[-1, -np.sqrt(2), -1],
               [0, 0, 0],
               [1, np.sqrt(2), 1]])

k2 = np.array([[-1, 0, 1],
               [-np.sqrt(2), 0, np.sqrt(2)],
               [-1, 0, 1]])
```

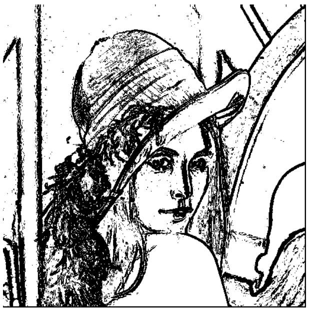


(e) Kirsch's Compass Operator:

Threshold = 135

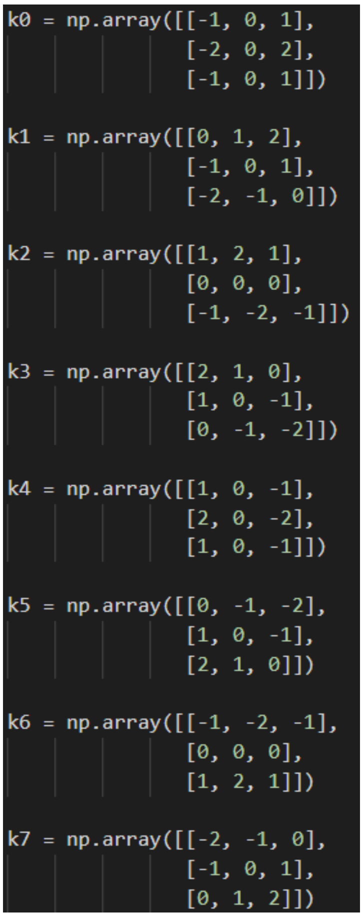


```
k0 = np.array([[-3, -3, 5],
               [-3, 0, 5],
               [-3, -3, 5]])

k1 = np.array([[-3, 5, 5],
               [-3, 0, 5],
               [-3, -3, -3]])

k2 = np.array([[5, 5, 5],
               [-3, 0, -3],
               [-3, -3, -3]])

k3 = np.array([[5, 5, -3],
               [5, 0, -3],
               [-3, -3, -3]])

k4 = np.array([[5, -3, -3],
               [5, 0, -3],
               [5, -3, -3]])

k5 = np.array([[-3, -3, -3],
               [5, 0, -3],
               [5, 5, -3]])

k6 = np.array([[-3, -3, -3],
               [-3, 0, -3],
               [5, 5, 5]])

k7 = np.array([[-3, -3, -3],
               [-3, 0, 5],
               [-3, 5, 5]])
```

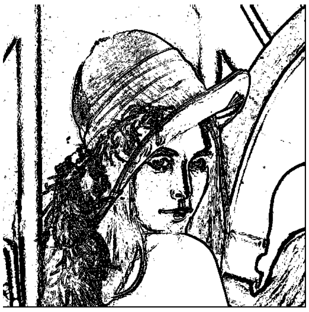


(f) Robinson's Compass Operator:

Threshold = 43

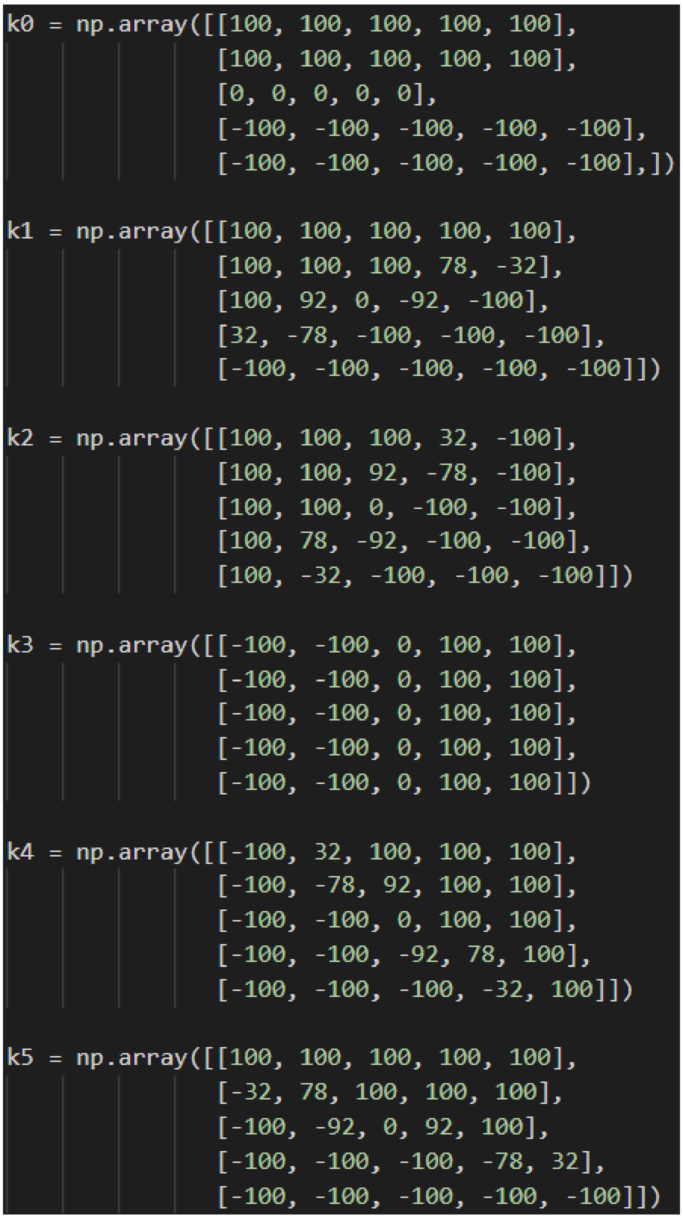


```
k0 = np.array([[-1, 0, 1],
               [-2, 0, 2],
               [-1, 0, 1]])

k1 = np.array([[0, 1, 2],
               [-1, 0, 1],
               [-2, -1, 0]])

k2 = np.array([[1, 2, 1],
               [0, 0, 0],
               [-1, -2, -1]])

k3 = np.array([[2, 1, 0],
               [1, 0, -1],
               [0, -1, -2]])

k4 = np.array([[1, 0, -1],
               [2, 0, -2],
               [1, 0, -1]])

k5 = np.array([[0, -1, -2],
               [1, 0, -1],
               [2, 1, 0]])

k6 = np.array([[-1, -2, -1],
               [0, 0, 0],
               [1, 2, 1]])

k7 = np.array([[-2, -1, 0],
               [-1, 0, 1],
               [0, 1, 2]])
```



(g) Nevatia-Babu 5x5 Operator:

Threshold = 12500



```
k0 = np.array([[100, 100, 100, 100, 100],
               [100, 100, 100, 100, 100],
               [0, 0, 0, 0, 0],
               [-100, -100, -100, -100, -100],
               [-100, -100, -100, -100, -100],])

k1 = np.array([[100, 100, 100, 100, 100],
               [100, 100, 100, 78, -32],
               [100, 92, 0, -92, -100],
               [32, -78, -100, -100, -100],
               [-100, -100, -100, -100, -100]])

k2 = np.array([[100, 100, 100, 32, -100],
               [100, 100, 92, -78, -100],
               [100, 100, 0, -100, -100],
               [100, 78, -92, -100, -100],
               [100, -32, -100, -100, -100]])

k3 = np.array([[-100, -100, 0, 100, 100],
               [-100, -100, 0, 100, 100],
               [-100, -100, 0, 100, 100],
               [-100, -100, 0, 100, 100],
               [-100, -100, 0, 100, 100]])

k4 = np.array([[-100, 32, 100, 100, 100],
               [-100, -78, 92, 100, 100],
               [-100, -100, 0, 100, 100],
               [-100, -100, -92, 78, 100],
               [-100, -100, -100, -32, 100]])

k5 = np.array([[100, 100, 100, 100, 100],
               [-32, 78, 100, 100, 100],
               [-100, -92, 0, 92, 100],
               [-100, -100, -100, -78, 32],
               [-100, -100, -100, -100, -100]])
```