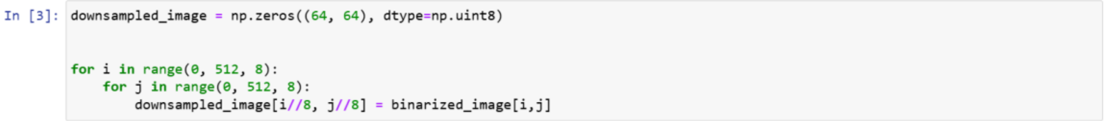
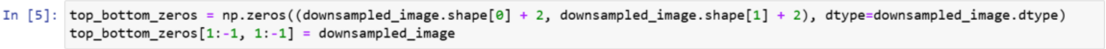


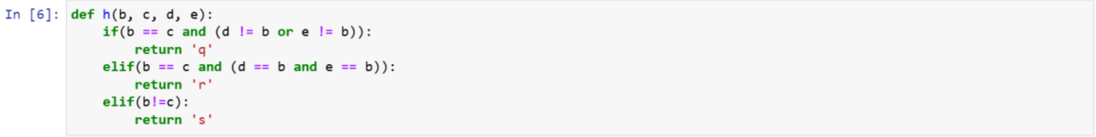
In [3]:
```python
downsampled_image = np.zeros((64, 64), dtype=np.uint8)


for i in range(0, 512, 8):
    for j in range(0, 512, 8):
        downsampled_image[i//8, j//8] = binarized_image[i,j]
```

將照片二值化後，將圖片從 512*512 改為 64*64，取左上角

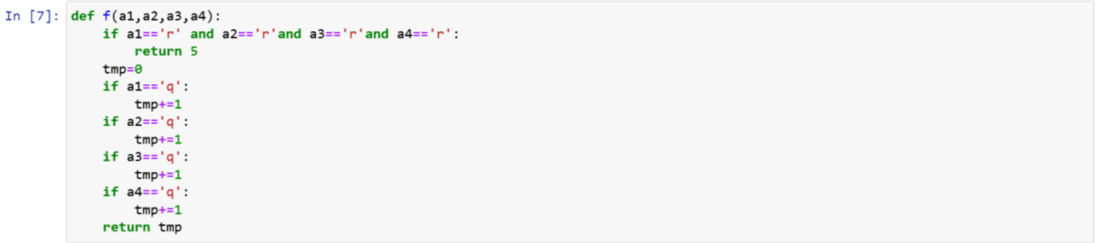


In [5]:
```python
top_bottom_zeros = np.zeros((downsampled_image.shape[0] + 2, downsampled_image.shape[1] + 2), dtype=downsampled_image.dtype)
top_bottom_zeros[1:-1, 1:-1] = downsampled_image
```

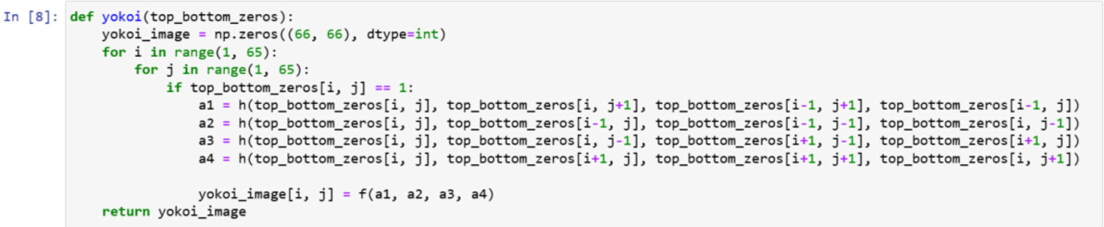
保留上下左右各一行避免超出邊界



In [6]:
```python
def h(b, c, d, e):
    if(b == c and (d != b or e != b)):
        return 'q'
    elif(b == c and (d == b and e == b)):
        return 'r'
    elif(b!=c):
        return 's'
```

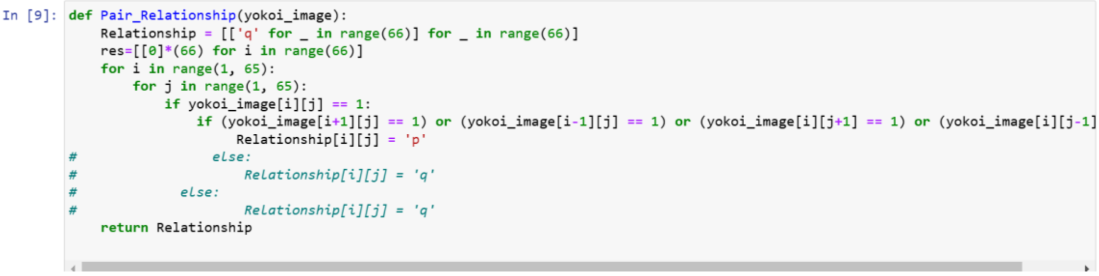


In [7]:
```python
def f(a1,a2,a3,a4):
    if a1=='r' and a2=='r'and a3=='r'and a4=='r':
        return 5
    tmp=0
    if a1=='q':
        tmp+=1
    if a2=='q':
        tmp+=1
    if a3=='q':
        tmp+=1
    if a4=='q':
        tmp+=1
    return tmp
```



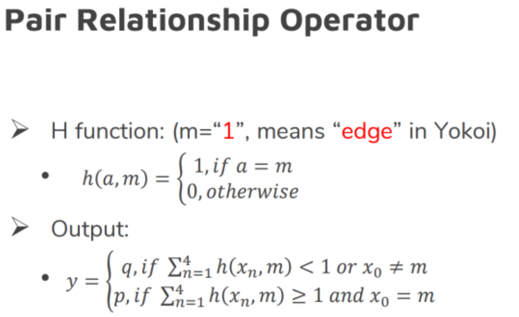
In [8]:
```python
def yokoi(top_bottom_zeros):
    yokoi_image = np.zeros((66, 66), dtype=int)
    for i in range(1, 65):
        for j in range(1, 65):
            if top_bottom_zeros[i, j] == 1:
                a1 = h(top_bottom_zeros[i, j], top_bottom_zeros[i, j+1], top_bottom_zeros[i-1, j+1], top_bottom_zeros[i-1, j])
                a2 = h(top_bottom_zeros[i, j], top_bottom_zeros[i-1, j], top_bottom_zeros[i-1, j-1], top_bottom_zeros[i, j-1])
                a3 = h(top_bottom_zeros[i, j], top_bottom_zeros[i, j-1], top_bottom_zeros[i+1, j-1], top_bottom_zeros[i+1, j])
                a4 = h(top_bottom_zeros[i, j], top_bottom_zeros[i+1, j], top_bottom_zeros[i+1, j+1], top_bottom_zeros[i, j+1])

                yokoi_image[i, j] = f(a1, a2, a3, a4)
    return yokoi_image
```

使用上週實作的 yokoi function



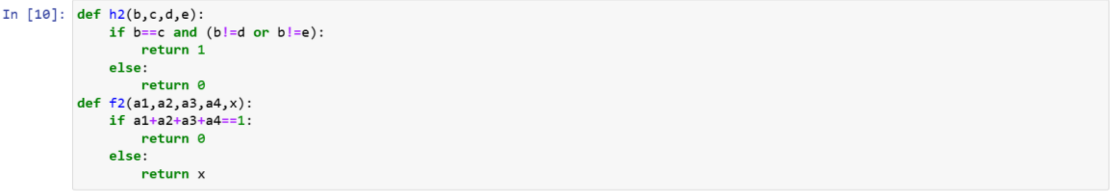
In [9]:
```python
def Pair_Relationship(yokoi_image):
    Relationship = [['q' for _ in range(66)] for _ in range(66)]
    res=[[0]*(66) for i in range(66)]
    for i in range(1, 65):
        for j in range(1, 65):
            if yokoi_image[i][j] == 1:
                if (yokoi_image[i+1][j] == 1) or (yokoi_image[i-1][j] == 1) or (yokoi_image[i][j+1] == 1) or (yokoi_image[i][j-1]
                    Relationship[i][j] = 'p'
#                 else:
#                     Relationship[i][j] = 'q'
#             else:
#                     Relationship[i][j] = 'q'
    return Relationship
```

實作 ppt 中的 pair_relation

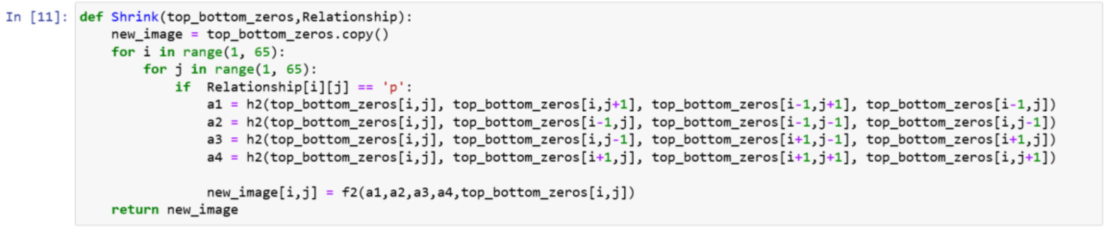


Pair Relationship Operator

- H function: (m="1", means "edge" in Yokoi)
  - $h(a,m) = \begin{cases} 1, if\ a = m \\ 0, otherwise \end{cases}$
- Output:
  - $y = \begin{cases} q, if\ \sum_{n=1}^{4} h(x_n, m) < 1\ or\ x_0 \neq m \\ p, if\ \sum_{n=1}^{4} h(x_n, m) \geq 1\ and\ x_0 = m \end{cases}$

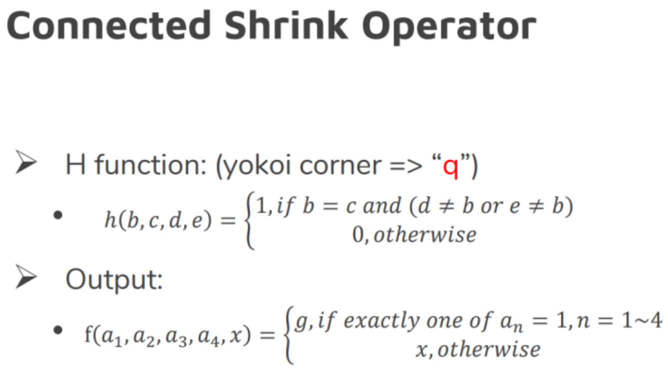


In [10]:
```python
def h2(b,c,d,e):
    if b==c and (b!=d or b!=e):
        return 1
    else:
        return 0
def f2(a1,a2,a3,a4,x):
    if a1+a2+a3+a4==1:
        return 0
    else:
        return x
```



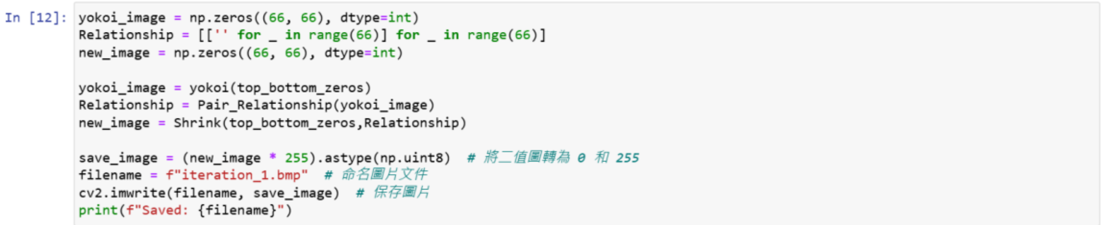
```
In [11]: def Shrink(top_bottom_zeros,Relationship):
             new_image = top_bottom_zeros.copy()
             for i in range(1, 65):
                 for j in range(1, 65):
                     if  Relationship[i][j] == 'p':
                         a1 = h2(top_bottom_zeros[i,j], top_bottom_zeros[i,j+1], top_bottom_zeros[i-1,j+1], top_bottom_zeros[i-1,j])
                         a2 = h2(top_bottom_zeros[i,j], top_bottom_zeros[i-1,j], top_bottom_zeros[i-1,j-1], top_bottom_zeros[i,j-1])
                         a3 = h2(top_bottom_zeros[i,j], top_bottom_zeros[i,j-1], top_bottom_zeros[i+1,j-1], top_bottom_zeros[i+1,j])
                         a4 = h2(top_bottom_zeros[i,j], top_bottom_zeros[i+1,j], top_bottom_zeros[i+1,j+1], top_bottom_zeros[i,j+1])

                         new_image[i,j] = f2(a1,a2,a3,a4,top_bottom_zeros[i,j])
             return new_image
```

實作 ppt 中 shrink 的部分



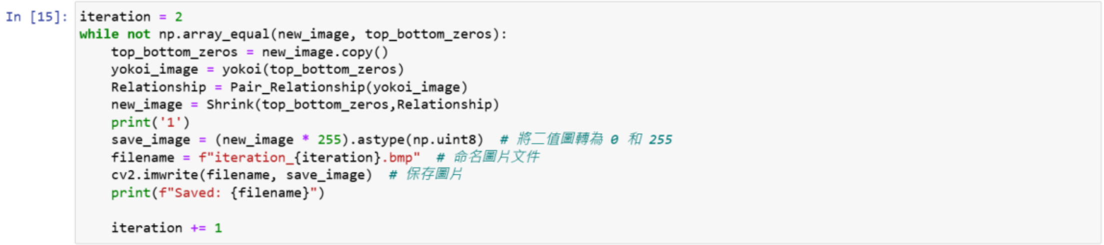
Connected Shrink Operator

- H function: (yokoi corner => “q”)
  - $h(b,c,d,e) = \begin{cases} 1, if\ b = c\ and\ (d \neq b\ or\ e \neq b) \\ 0, otherwise \end{cases}$
- Output:
  - $f(a_1, a_2, a_3, a_4, x) = \begin{cases} g, if\ exactly\ one\ of\ a_n = 1, n = 1{\sim}4 \\ x, otherwise \end{cases}$



```
In [12]: yokoi_image = np.zeros((66, 66), dtype=int)
         Relationship = [['' for _ in range(66)] for _ in range(66)]
         new_image = np.zeros((66, 66), dtype=int)

         yokoi_image = yokoi(top_bottom_zeros)
         Relationship = Pair_Relationship(yokoi_image)
         new_image = Shrink(top_bottom_zeros,Relationship)

         save_image = (new_image * 255).astype(np.uint8)  # 將二值圖轉為 0 和 255
         filename = f"iteration_1.bmp"  # 命名圖片文件
         cv2.imwrite(filename, save_image)  # 保存圖片
         print(f"Saved: {filename}")
```

執行確定結果



```
In [15]: iteration = 2
         while not np.array_equal(new_image, top_bottom_zeros):
             top_bottom_zeros = new_image.copy()
             yokoi_image = yokoi(top_bottom_zeros)
             Relationship = Pair_Relationship(yokoi_image)
             new_image = Shrink(top_bottom_zeros,Relationship)
             print('1')
             save_image = (new_image * 255).astype(np.uint8)  # 將二值圖轉為 0 和 255
             filename = f"iteration_{iteration}.bmp"  # 命名圖片文件
             cv2.imwrite(filename, save_image)  # 保存圖片
             print(f"Saved: {filename}")

             iteration += 1
```

反覆執行直到不再改變