附錄 A. 本章使用的函數

1. 給予一個品質參數 quality，quantizedTable(quality) 函數可傳回該品質參數下，用於亮度和彩度的量化表：

```
01  def quantizedTable(quality=50):
02      std_lumQT = np.array(    # 50% luminance quantized table
03          [[ 16,  11,  10,  16,  24,  40,  51,  61],
04           [ 12,  12,  14,  19,  26,  58,  60,  55],
05           [ 14,  13,  16,  24,  40,  57,  69,  56],
06           [ 14,  17,  22,  29,  51,  87,  80,  62],
07           [ 18,  22,  37,  56,  68, 109, 103,  77],
08           [ 24,  35,  55,  64,  81, 104, 113,  92],
09           [ 49,  64,  78,  87, 103, 121, 120, 101],
10           [ 72,  92,  95,  98, 112, 100, 103,  99]])
11      std_chrQT = np.array(    # 50% chrominance quantized table
12          [[ 17,  18,  24,  47,  99,  99,  99,  99],
13           [ 18,  21,  26,  66,  99,  99,  99,  99],
14           [ 24,  26,  56,  99,  99,  99,  99,  99],
15           [ 47,  66,  99,  99,  99,  99,  99,  99],
16           [ 99,  99,  99,  99,  99,  99,  99,  99],
17           [ 99,  99,  99,  99,  99,  99,  99,  99],
18           [ 99,  99,  99,  99,  99,  99,  99,  99],
19           [ 99,  99,  99,  99,  99,  99,  99,  99]])
20
21      qualityScale = 5000/quality if(quality < 50) else 200-quality*2
22      lumQT = np.floor((std_lumQT*qualityScale+50)/100).clip(1,255).astype(int)
23      chrQT = np.floor((std_chrQT*qualityScale+50)/100).clip(1,255).astype(int)
24      return lumQT,chrQT
```

在上面的程式中，std_lumQT 和 std_chrQT 分別是 quality=50 時的亮度和彩度的量化表。

2. 給予 16 進位的 huf_tables，builtHT(huf_tables, param='encode') 函數可傳回 $\mathcal{H}_L^{dc}$ 、$\mathcal{H}_L^{ac}$ 、 $\mathcal{H}_C^{dc}$ 和 $\mathcal{H}_C^{ac}$ 四個編碼用的 Huffman 表。builtHT(huf_tables, 'decode') 則可傳回 $\widehat{\mathcal{H}}_L^{dc}$ 、$\widehat{\mathcal{H}}_L^{ac}$、$\widehat{\mathcal{H}}_C^{dc}$ 和 $\widehat{\mathcal{H}}_C^{ac}$ 四個解碼用的 Huffman 表。你也可以針對特定圖像用自己設計的 Huffman 表傳入函數的 huf_tables 參數中。

```
ht_default= {'dc0':'000105010101010101000000000000000001020304050607080 90a0b',
             'ac0':'00020103030204030505040400000017d01020300041105122131 4106'\
                   '135161072271143281 91a1082342b1c11552d1f02433627282090a16'\
                   '1718191a25262728292a3435363738393a434445464748494a535455'\
                   '565758595a636465666768696a737475767778797a83848586878889'\
                   '8a92939495969798999aa2a3a4a5a6a7a8a9aab2b3b4b5b6b7b8b9ba'\
                   'c2c3c4c5c6c7c8c9cad2d3d4d5d6d7d8d9dae1e2e3e4e5e6e7e8e9ea'\
                   'f1f2f3f4f5f6f7f8f9fa',
              'dc1':'000301010101010101010100000000000000010203040506070 8090a0b',
              'ac1':'00020102040403040705040400010277000102031104052131 061241'\
                    '5107617113223281 08144291a1b1c109233352f0156272d10a162434'\
                    'e125f11718191a262728292a35363738393a434445464748494a5354'\
                    '55565758595a636465666768696a737475767778797a828384858687'\
                    '88898a92939495969798999aa2a3a4a5a6a7a8a9aab2b3b4b5b6b7b8'\
                    'b9bac2c3c4c5c6c7c8c9cad2d3d4d5d6d7d8d9dae2e3e4e5e6e7e8e9'\
                    'eaf2f3f4f5f6f7f8f9fa'
             }

def buildHT(huf_tables,param='encode'):  # biuld huffman table from Hex-digits
    HT=[]
    for ht in ['dc0','ac0','dc1','ac1']:
        dht=bytes.fromhex(huf_tables[ht])
        table = {}
        num_codes_by_length=list(dht[:16])
        code_ptr = 16
        code_val = 0b0
        for code_length, num_codes in enumerate(num_codes_by_length, 1):
          if num_codes != 0:
            for _ in range(num_codes):
              if param=='decode': # for decode
                  table.update({'{:0{}b}'.format(code_val, code_length):dht[code_ptr]})
              else:               # for encode
                  table.update({dht[code_ptr]:'{:0{}b}'.format(code_val, code_length)})
              code_ptr += 1
              code_val += 1
          code_val <<= 1
        HT.append(table)
    return HT
```

附錄 B. AC 編碼表

Table K.5 – Table for luminance AC coefficients

Run/Size	Code length	Code word
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
0/5	5	11010
0/6	7	1111000
0/7	8	11111000
0/8	10	1111110110
0/9	16	1111111110000010
0/A	16	1111111110000011
1/1	4	1100
1/2	5	11011
1/3	7	1111001
1/4	9	111110110
1/5	11	11111110110
1/6	16	1111111110000100
1/7	16	1111111110000101
1/8	16	1111111110000110
1/9	16	1111111110000111
1/A	16	1111111110001000
2/1	5	11100
2/2	8	11111001
2/3	10	1111110111
2/4	12	111111110100
2/5	16	1111111110001001
2/6	16	1111111110001010
2/7	16	1111111110001011
2/8	16	1111111110001100
2/9	16	1111111110001101
2/A	16	1111111110001110
3/1	6	111010
3/2	9	111110111
3/3	12	111111110101
3/4	16	1111111110001111
3/5	16	1111111110010000
3/6	16	1111111110010001
3/7	16	1111111110010010
3/8	16	1111111110010011
3/9	16	1111111110010100
3/A	16	1111111110010101
4/1	6	111011
4/2	10	1111111000
4/3	16	1111111110010110
4/4	16	1111111110010111
4/5	16	1111111110011000
4/6	16	1111111110011001
4/7	16	1111111110011010
4/8	16	1111111110011011
4/9	16	1111111110011100
4/A	16	1111111110011101
5/1	7	1111010
5/2	11	11111110111
5/3	16	1111111110011110
5/4	16	1111111110011111
5/5	16	1111111110100000
5/6	16	1111111110100001
5/7	16	1111111110100010
5/8	16	1111111110100011
5/9	16	1111111110100100
5/A	16	1111111110100101
6/1	7	1111011
6/2	12	111111110110
6/3	16	1111111110100110
6/4	16	1111111110100111
6/5	16	1111111110101000
6/6	16	1111111110101001
6/7	16	1111111110101010
6/8	16	1111111110101011
6/9	16	1111111110101100
6/A	16	1111111110101101
7/1	8	11111010
7/2	12	111111110111
7/3	16	1111111110101110
7/4	16	1111111110101111
7/5	16	1111111110110000
7/6	16	1111111110110001
7/7	16	1111111110110010
7/8	16	1111111110110011

Run/Size	Code length	Code word
7/9	16	1111111110110100
7/A	16	1111111110110101
8/1	9	111111000
8/2	15	111111111000000
8/3	16	1111111110110110
8/4	16	1111111110110111
8/5	16	1111111110111000
8/6	16	1111111110111001
8/7	16	1111111110111010
8/8	16	1111111110111011
8/9	16	1111111110111100
8/A	16	1111111110111101
9/1	9	111111001
9/2	16	1111111110111110
9/3	16	1111111110111111
9/4	16	1111111111000000
9/5	16	1111111111000001
9/6	16	1111111111000010
9/7	16	1111111111000011
9/8	16	1111111111000100
9/9	16	1111111111000101
9/A	16	1111111111000110
A/1	9	111111010
A/2	16	1111111111000111
A/3	16	1111111111001000
A/4	16	1111111111001001
A/5	16	1111111111001010
A/6	16	1111111111001011
A/7	16	1111111111001100
A/8	16	1111111111001101
A/9	16	1111111111001110
A/A	16	1111111111001111
B/1	10	1111111001
B/2	16	1111111111010000
B/3	16	1111111111010001
B/4	16	1111111111010010
B/5	16	1111111111010011
B/6	16	1111111111010100
B/7	16	1111111111010101
B/8	16	1111111111010110
B/9	16	1111111111010111
B/A	16	1111111111011000

Run/Size	Code length	Code word
C/1	10	1111111010
C/2	16	1111111111011001
C/3	16	1111111111011010
C/4	16	1111111111011011
C/5	16	1111111111011100
C/6	16	1111111111011101
C/7	16	1111111111011110
C/8	16	1111111111011111
C/9	16	1111111111100000
C/A	16	1111111111100001
D/1	11	11111111000
D/2	16	1111111111100010
D/3	16	1111111111100011
D/4	16	1111111111100100
D/5	16	1111111111100101
D/6	16	1111111111100110
D/7	16	1111111111100111
D/8	16	1111111111101000
D/9	16	1111111111101001
D/A	16	1111111111101010
E/1	16	1111111111101011
E/2	16	1111111111101100
E/3	16	1111111111101101
E/4	16	1111111111101110
E/5	16	1111111111101111
E/6	16	1111111111110000
E/7	16	1111111111110001
E/8	16	1111111111110010
E/9	16	1111111111110011
E/A	16	1111111111110100
F/0 (ZRL)	11	11111111001
F/1	16	1111111111110101
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

Table K.6–Table for chrominance AC coefficients

Run/Size	Code length	Code word
0/0(EOB)	2	00
0/1	2	01
0/2	3	100
0/3	4	1010
0/4	5	11000
0/5	5	11001
0/6	6	111000
0/7	7	1111000
0/8	9	111110100
0/9	10	1111110110
0/A	12	111111110100
1/1	4	1011
1/2	6	111001
1/3	8	11110110
1/4	9	111110101
1/5	11	11111110110
1/6	12	111111110101
1/7	16	1111111110001000
1/8	16	1111111110001001
1/9	16	1111111110001010
1/A	16	1111111110001011
2/1	5	11010
2/2	8	11110111
2/3	10	1111110111
2/4	12	111111110110
2/5	15	111111111000010
2/6	16	1111111110001100
2/7	16	1111111110001101
2/8	16	1111111110001110
2/9	16	1111111110001111
2/A	16	1111111110010000
3/1	5	11011
3/2	8	11111000
3/3	10	1111111000
3/4	12	111111110111
3/5	16	1111111110010001
3/6	16	1111111110010010
3/7	16	1111111110010011
3/8	16	1111111110010100
3/9	16	1111111110010101
3/A	16	1111111110010110
4/1	6	111010
4/2	9	111110110
4/3	16	1111111110010111
4/4	16	1111111110011000
4/5	16	1111111110011001
4/6	16	1111111110011010
4/7	16	1111111110011011
4/8	16	1111111110011100
4/9	16	1111111110011101
4/A	16	1111111110011110
5/1	6	111011
5/2	10	1111111001
5/3	16	1111111110011111
5/4	16	1111111110100000
5/5	16	1111111110100001
5/6	16	1111111110100010
5/7	16	1111111110100011
5/8	16	1111111110100100
5/9	16	1111111110100101
5/A	16	1111111110100110
6/1	7	1111001
6/2	11	11111110111
6/3	16	1111111110100111
6/4	16	1111111110101000
6/5	16	1111111110101001
6/6	16	1111111110101010
6/7	16	1111111110101011
6/8	16	1111111110101100
6/9	16	1111111110101101
6/A	16	1111111110101110
7/1	7	1111010
7/2	11	11111111000
7/3	16	1111111110101111
7/4	16	1111111110110000
7/5	16	1111111110110001
7/6	16	1111111110110010
7/7	16	1111111110110011
7/8	16	1111111110110100
7/9	16	1111111110110101
7/A	16	1111111110110110

Run/Size	Code length	Code word
8/1	8	11111001
8/2	16	1111111110110111
8/3	16	1111111110111000
8/4	16	1111111110111001
8/5	16	1111111110111010
8/6	16	1111111110111011
8/7	16	1111111110111100
8/8	16	1111111110111101
8/9	16	1111111110111110
8/A	16	1111111110111111
9/1	9	111110111
9/2	16	1111111111000000
9/3	16	1111111111000001
9/4	16	1111111111000010
9/5	16	1111111111000011
9/6	16	1111111111000100
9/7	16	1111111111000101
9/8	16	1111111111000110
9/9	16	1111111111000111
9/A	16	1111111111001000
A/1	9	111111000
A/2	16	1111111111001001
A/3	16	1111111111001010
A/4	16	1111111111001011
A/5	16	1111111111001100
A/6	16	1111111111001101
A/7	16	1111111111001110
A/8	16	1111111111001111
A/9	16	1111111111010000
A/A	16	1111111111010001
B/1	9	111111001
B/2	16	1111111111010010
B/3	16	1111111111010011
B/4	16	1111111111010100
B/5	16	1111111111010101
B/6	16	1111111111010110
B/7	16	1111111111010111
B/8	16	1111111111011000
B/9	16	1111111111011001
B/A	16	1111111111011010
C/1	9	111111010

Run/Size	Code length	Code word
C/2	16	1111111111011011
C/3	16	1111111111011100
C/4	16	1111111111011101
C/5	16	1111111111011110
C/6	16	1111111111011111
C/7	16	1111111111100000
C/8	16	1111111111100001
C/9	16	1111111111100010
C/A	16	1111111111100011
D/1	11	11111111001
D/2	16	1111111111100100
D/3	16	1111111111100101
D/4	16	1111111111100110
D/5	16	1111111111100111
D/6	16	1111111111101000
D/7	16	1111111111101001
D/8	16	1111111111101010
D/9	16	1111111111101011
D/A	16	1111111111101100
E/1	14	11111111100000
E/2	16	1111111111101101
E/3	16	1111111111101110
E/4	16	1111111111101111
E/5	16	1111111111110000
E/6	16	1111111111110001
E/7	16	1111111111110010
E/8	16	1111111111110011
E/9	16	1111111111110100
E/A	16	1111111111110101
F/0 (ZRL)	10	1111111010
F/1	15	111111111000011
F/2	16	1111111111110110
F/3	16	1111111111110111
F/4	16	1111111111111000
F/5	16	1111111111111001
F/6	16	1111111111111010
F/7	16	1111111111111011
F/8	16	1111111111111100
F/9	16	1111111111111101
F/A	16	1111111111111110

附錄 C. test16.bmp 編碼結果參考

test16.bmp, $q = 55$ 時的編碼結果

$\{\bar{y}_i^Q\}_{i=0}^3$

```
[[ 20,  26, -12,  -1,   3,  -1,   0,   1],
 [  4, -13,   3,   0,  -2,   1,   0,   0],
 [  6,  -7,   2,   1,  -1,   0,   0,  -1],
 [ -2,   3,   2,  -1,   1,   0,   0,   0],
 [  1,   1,   0,   1,   0,   0,   0,   0],
 [ -2,   1,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0]],

[[ 14,  22,  -9, -10,   0,   0,  -1,   0],
 [  4,  -3,   5,   0,   1,   0,   0,   0],
 [  2,  -2,   3,  -1,   0,   0,   0,   0],
 [  1,   1,   1,   1,   1,   0,   0,   0],
 [ -1,   0,  -1,   0,   0,   0,   0,   0],
 [ -1,   0,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0]],

[[ 34,   4, -19,   9,  -3,   0,   0,   0],
 [ 10,  -1,   1,   0,   0,   1,  -1,   0],
 [ -2,   1,  -5,   1,   0,   0,   0,   0],
 [ -3,   7,  -2,   0,   0,   0,   0,   0],
 [ -2,  -1,   1,   0,   0,   0,   0,   0],
 [  0,  -1,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0],
 [  0,   0,  -1,   0,   0,   0,   0,   0]],

[[ 22,   6, -17,  -4,  -6,  -1,  -1,   0],
 [  4,   5,  -1,  -1,   2,  -1,  -1,   0],
 [ -5,   3,   3,  -1,   1,   1,   0,   0],
 [ -7,   0,  -2,  -3,  -1,   0,   0,   0],
 [ -2,   1,   0,  -1,   0,   0,   0,   0],
 [  0,   1,   1,   0,   0,   0,   0,   0],
 [ -1,   0,   0,   0,   0,   0,   0,   0],
 [  0,   0,   0,   0,   0,   0,   0,   0]]
```

$\{cb_i^Q\}_{i=0}^3$

```
[[[-1,  0,  0,  0,  0,  0,  0,  0],
  [-1, -1,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0]],

 [[-4,  2, -1,  0,  0,  0,  0,  0],
  [-1,  1,  0,  0,  0,  0,  0,  0],
  [ 0,  1,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0],
  [ 0,  0,  0,  0,  0,  0,  0,  0]],

[[-1, -3,  2,  0,  0,  0,  0,  0],
 [ 0,  1, -1,  0,  0,  0,  0,  0],
 [ 1, -1,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]],

[[-3,  3,  1,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]]]
```

$\{cr_i^Q\}_{i=0}^3$

```
[[ 1, -1,  1,  0,  0,  0,  0,  0],
 [ 0,  1, -1,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]],

[[ 7, -3,  0,  1,  0,  0,  0,  0],
 [-1,  0, -1,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]],

[[ 2,  1, -1,  0,  0,  0,  0,  0],
 [-1, -1,  0,  0,  0,  0,  0,  0],
 [-1,  1,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]],

[[ 4, -1, -1,  0,  0,  0,  0,  0],
 [ 1,  0,  0,  0,  0,  0,  0,  0],
 [ 0, -1,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0]]]
```

```
Block number: 0----
Encode dc of Y[0]
p:0, '110'+'10100'
Encode ac of Y[0]
p:8, ac=26, code= '11010'+'11010'
p:18, ac=4, code= '100'+'100'
p:24, ac=6, code= '100'+'110'
p:30, ac=-13, code= '1011'+'0010'
p:38, ac=-12, code= '1011'+'0011'
p:46, ac=-1, code= '00'+'0'
p:49, ac=3, code= '01'+'11'
p:53, ac=-7, code= '100'+'000'
p:59, ac=-2, code= '01'+'01'
p:63, ac=1, code= '00'+'1'
p:66, ac=3, code= '01'+'11'
p:70, ac=2, code= '01'+'10'
p:74, ac=3, code= '11011'+'11'
p:81, ac=-1, code= '00'+'0'
p:84, ac=-2, code= '01'+'01'
p:88, ac=1, code= '00'+'1'
p:91, ac=2, code= '01'+'10'
p:95, ac=1, code= '00'+'1'
p:98, ac=-2, code= '01'+'01'
p:102, ac=1, code= '1100'+'1'
p:107, ac=-1, code= '1100'+'0'
p:112, ac=-1, code= '00'+'0'
p:115, ac=1, code= '00'+'1'
p:118, ac=1, code= '1100'+'1'
p:123, ac=1, code= '11100'+'1'
p:129, ac=1, code= '00'+'1'
p:132, ac=-1, code= '111111010'+'0'
p:142, ac=0, code= '1010' (EOB)
Encode dc of Cb[0]
p:146, '01'+'0'
Encode ac of Cb[0]
p:149, ac=-1, code= '1011'+'0'
p:154, ac=-1, code= '1011'+'0'
p:159, ac=0, code= '00' (EOB)
Encode dc of Cr[0]
p:161, '01'+'1'
Encode ac of Cr[0]
p:164, ac=-1, code= '01'+'0'
p:167, ac=1, code= '11010'+'1'
p:173, ac=1, code= '01'+'1'
p:176, ac=-1, code= '1011'+'0'
p:181, ac=0, code= '00' (EOB)
Block number: 1----
Encode dc of Y[1]
p:183, '100'+'001'
Encode ac of Y[1]
p:189, ac=22, code= '11010'+'10110'
p:199, ac=4, code= '100'+'100'
p:205, ac=2, code= '01'+'10'
p:209, ac=-3, code= '01'+'00'
p:213, ac=-9, code= '1011'+'0110'
p:221, ac=-10, code= '1011'+'0101'
p:229, ac=5, code= '100'+'101'
p:235, ac=-2, code= '01'+'01'
p:239, ac=1, code= '00'+'1'
p:242, ac=-1, code= '00'+'0'
p:245, ac=1, code= '00'+'1'
p:248, ac=3, code= '01'+'11'
p:252, ac=1, code= '111010'+'1'
p:259, ac=-1, code= '00'+'0'
p:262, ac=1, code= '00'+'1'
p:265, ac=-1, code= '1100'+'0'
p:270, ac=-1, code= '11100'+'0'
p:276, ac=1, code= '00'+'1'
p:279, ac=-1, code= '11100'+'0'
p:285, ac=1, code= '111010'+'1'
p:292, ac=0, code= '1010' (EOB)
Encode dc of Cb[1]
p:296, '10'+'00'
Encode ac of Cb[1]
p:300, ac=2, code= '100'+'10'
p:305, ac=-1, code= '01'+'0'
p:308, ac=1, code= '1011'+'1'
p:313, ac=-1, code= '01'+'0'
p:316, ac=1, code= '11010'+'1'
p:322, ac=0, code= '00' (EOB)
Encode dc of Cr[1]
p:324, '110'+'110'
Encode ac of Cr[1]
p:330, ac=-3, code= '100'+'00'
p:335, ac=-1, code= '01'+'0'
p:338, ac=1, code= '11011'+'1'
p:344, ac=-1, code= '01'+'0'
p:347, ac=0, code= '00' (EOB)
Block number: 2----
Encode dc of Y[2]
p:349, '110'+'10100'
Encode ac of Y[2]
```

```
p:357, ac=4, code= '100'+'100'
p:363, ac=10, code= '1011'+'1010'
p:371, ac=-2, code= '01'+'01'
p:375, ac=-1, code= '00'+'0'
p:378, ac=-19, code= '11010'+'01100'
p:388, ac=9, code= '1011'+'1001'
p:396, ac=1, code= '00'+'1'
p:399, ac=1, code= '00'+'1'
p:402, ac=-3, code= '01'+'00'
p:406, ac=-2, code= '01'+'01'
p:410, ac=7, code= '100'+'111'
p:416, ac=-5, code= '100'+'010'
p:422, ac=-3, code= '11011'+'00'
p:429, ac=1, code= '11100'+'1'
p:435, ac=-2, code= '01'+'01'
p:439, ac=-1, code= '00'+'0'
p:442, ac=-1, code= '11100'+'0'
p:448, ac=1, code= '00'+'1'
p:451, ac=1, code= '11100'+'1'
p:457, ac=-1, code= '11100'+'0'
p:463, ac=0, code= '11111111001'
(ZRL)
p:474, ac=-1, code= '11100'+'0'
p:480, ac=0, code= '1010' (EOB)
Encode dc of Cb[2]
p:484, '10'+'11'
Encode ac of Cb[2]
p:488, ac=-3, code= '100'+'00'
p:493, ac=1, code= '1011'+'1'
p:498, ac=1, code= '01'+'1'
p:501, ac=2, code= '100'+'10'
p:506, ac=-1, code= '1011'+'0'
p:511, ac=-1, code= '01'+'0'
p:514, ac=0, code= '00' (EOB)
Encode dc of Cr[2]
p:516, '110'+'010'
Encode ac of Cr[2]
p:522, ac=1, code= '01'+'1'
p:525, ac=-1, code= '01'+'0'
p:528, ac=-1, code= '01'+'0'
p:531, ac=-1, code= '01'+'0'
p:534, ac=-1, code= '01'+'0'
p:537, ac=1, code= '11010'+'1'
p:543, ac=0, code= '00' (EOB)
Block number: 3----
Encode dc of Y[3]
p:545, '101'+'0011'
Encode ac of Y[3]
p:552, ac=6, code= '100'+'110'
p:558, ac=4, code= '100'+'100'
p:564, ac=-5, code= '100'+'010'
p:570, ac=5, code= '100'+'101'
p:576, ac=-17, code= '11010'+'01110'
p:586, ac=-4, code= '100'+'011'
p:592, ac=-1, code= '00'+'0'
p:595, ac=3, code= '01'+'11'
p:599, ac=-7, code= '100'+'000'
p:605, ac=-2, code= '01'+'01'
p:609, ac=3, code= '11011'+'11'
p:616, ac=-1, code= '00'+'0'
p:619, ac=-6, code= '100'+'001'
p:625, ac=-1, code= '00'+'0'
p:628, ac=2, code= '01'+'10'
p:632, ac=-1, code= '00'+'0'
p:635, ac=-2, code= '01'+'01'
p:639, ac=1, code= '00'+'1'
p:642, ac=-1, code= '1100'+'0'
p:647, ac=1, code= '00'+'1'
p:650, ac=-3, code= '11011'+'00'
p:657, ac=1, code= '00'+'1'
p:660, ac=-1, code= '00'+'0'
p:663, ac=-1, code= '00'+'0'
p:666, ac=-1, code= '1100'+'0'
p:671, ac=1, code= '00'+'1'
p:674, ac=-1, code= '00'+'0'
p:677, ac=-1, code= '00'+'0'
p:680, ac=1, code= '00'+'1'
p:683, ac=0, code= '1010' (EOB)
Encode dc of Cb[3]
p:687, '10'+'01'
Encode ac of Cb[3]
p:691, ac=3, code= '100'+'11'
p:696, ac=1, code= '11011'+'1'
p:702, ac=0, code= '00' (EOB)
Encode dc of Cr[3]
p:704, '10'+'10'
Encode ac of Cr[3]
p:708, ac=-2, code= '100'+'01'
p:713, ac=1, code= '01'+'1'
p:716, ac=-1, code= '11010'+'0'
p:722, ac=-1, code= '11010'+'0'
p:728, ac=0, code= '00' (EOB)
```

Output code stream

```
1101010011010110101001001001101011001010
1100110000111100000001010010111011011011
1110000101001011000101011100111000000001
1100111100100111111101001010010101101011
0000110101101010111011000010000011101010
1101001000110010010110110101101011001010
1010010000010111111010100000111000111000
0001111000111010110101000100100101011101
0110101001101101000000101101110100011010
1001001001011101001010001101001100101110
0100100101000101100111100010110110011100
1010100011100000011110011110001111111100
1111000101010111000001011101110010101100
1000110010011010010010010110101001010011
1001101001001000101001011101001110100011
0000111100000001011101111000100001000011
0000010100111000001110110000100000011000
0010000000011010100110011110111001010100
01011110100110100000
```

附錄 D test16.bmp 解碼結果參考

```
參數說明
--------------------------------
p:CSf 的位置
j:要解碼的 ac 在一個區塊裡的位置
sk：講義上的 sk
(run,ac): run 和係數 ac
--------------------------------

Block number: 0----
Decode dc of Y[0]
p:0, '110'+'10100': delta=20
Decode ac of Y[0]
p:8, j=1, '11010'+'11010': sk=5, (0,26)
p:18, j=2, '100'+'100': sk=3, (0,4)
p:24, j=3, '100'+'110': sk=3, (0,6)
p:30, j=4, '1011'+'0010': sk=4, (0,-13)
p:38, j=5, '1011'+'0011': sk=4, (0,-12)
p:46, j=6, '00'+'0': sk=1, (0,-1)
p:49, j=7, '01'+'11': sk=2, (0,3)
p:53, j=8, '100'+'000': sk=3, (0,-7)
p:59, j=9, '01'+'01': sk=2, (0,-2)
p:63, j=10, '00'+'1': sk=1, (0,1)
p:66, j=11, '01'+'11': sk=2, (0,3)
p:70, j=12, '01'+'10': sk=2, (0,2)
p:74, j=14, '11011'+'11': sk=18, (1,3)
p:81, j=15, '00'+'0': sk=1, (0,-1)
p:84, j=16, '01'+'01': sk=2, (0,-2)
p:88, j=17, '00'+'1': sk=1, (0,1)
p:91, j=18, '01'+'10': sk=2, (0,2)
p:95, j=19, '00'+'1': sk=1, (0,1)
p:98, j=20, '01'+'01': sk=2, (0,-2)
p:102, j=22, '1100'+'1': sk=17, (1,1)
p:107, j=24, '1100'+'0': sk=17, (1,-1)
p:112, j=25, '00'+'0': sk=1, (0,-1)
p:115, j=26, '00'+'1': sk=1, (0,1)
p:118, j=28, '1100'+'1': sk=17, (1,1)
p:123, j=31, '11100'+'1': sk=33, (2,1)
p:129, j=32, '00'+'1': sk=1, (0,1)
p:132, j=43, '111111010'+'0': sk=161, (10,-1)
p:142, j=43, '1010'+'': sk=0, (0,0) EOB
Decode dc of Cb[0]
p:146, '01'+'0': delta=-1
Decode ac of Cb[0]
p:149, j=2, '1011'+'0': sk=17, (1,-1)
p:154, j=4, '1011'+'0': sk=17, (1,-1)
p:159, j=4, '00'+'': sk=0, (0,0) EOB
Decode dc of Cr[0]
p:161, '01'+'1': delta=1
Decode ac of Cr[0]
p:164, j=1, '01'+'0': sk=1, (0,-1)
p:167, j=4, '11010'+'1': sk=33, (2,1)
p:173, j=5, '01'+'1': sk=1, (0,1)
p:176, j=7, '1011'+'0': sk=17, (1,-1)
p:181, j=7, '00'+'': sk=0, (0,0) EOB
Block number: 1----
Decode dc of Y[1]
p:183, '100'+'001': delta=-6
Decode ac of Y[1]
p:189, j=1, '11010'+'10110': sk=5, (0,22)
p:199, j=2, '100'+'100': sk=3, (0,4)
p:205, j=3, '01'+'10': sk=2, (0,2)
p:209, j=4, '01'+'00': sk=2, (0,-3)
p:213, j=5, '1011'+'0110': sk=4, (0,-9)
p:221, j=6, '1011'+'0101': sk=4, (0,-10)
p:229, j=7, '100'+'101': sk=3, (0,5)
p:235, j=8, '01'+'01': sk=2, (0,-2)
p:239, j=9, '00'+'1': sk=1, (0,1)
p:242, j=10, '00'+'0': sk=1, (0,-1)
p:245, j=11, '00'+'1': sk=1, (0,1)
p:248, j=12, '01'+'11': sk=2, (0,3)
p:252, j=16, '111010'+'1': sk=49, (3,1)
p:259, j=17, '00'+'0': sk=1, (0,-1)
```

```
p:262, j=18, '00'+'1': sk=1, (0,1)
p:265, j=20, '1100'+'0': sk=17, (1,-1)
p:270, j=23, '11100'+'0': sk=33, (2,-1)
p:276, j=24, '00'+'1': sk=1, (0,1)
p:279, j=27, '11100'+'0': sk=33, (2,-1)
p:285, j=31, '111010'+'1': sk=49, (3,1)
p:292, j=31, '1010'+'': sk=0, (0,0) EOB
Decode dc of Cb[1]
p:296, '10'+'00': delta=-3
Decode ac of Cb[1]
p:300, j=1, '100'+'10': sk=2, (0,2)
p:305, j=2, '01'+'0': sk=1, (0,-1)
p:308, j=4, '1011'+'1': sk=17, (1,1)
p:313, j=5, '01'+'0': sk=1, (0,-1)
p:316, j=8, '11010'+'1': sk=33, (2,1)
p:322, j=8, '00'+'': sk=0, (0,0) EOB
Decode dc of Cr[1]
p:324, '110'+'110': delta=6
Decode ac of Cr[1]
p:330, j=1, '100'+'00': sk=2, (0,-3)
p:335, j=2, '01'+'0': sk=1, (0,-1)
p:338, j=6, '11011'+'1': sk=49, (3,1)
p:344, j=7, '01'+'0': sk=1, (0,-1)
p:347, j=7, '00'+'': sk=0, (0,0) EOB
Block number: 2----
Decode dc of Y[2]
p:349, '110'+'10100': delta=20
Decode ac of Y[2]
p:357, j=1, '100'+'100': sk=3, (0,4)
p:363, j=2, '1011'+'1010': sk=4, (0,10)
p:371, j=3, '01'+'01': sk=2, (0,-2)
p:375, j=4, '00'+'0': sk=1, (0,-1)
p:378, j=5, '11010'+'01100': sk=5, (0,-
19)
p:388, j=6, '1011'+'1001': sk=4, (0,9)
p:396, j=7, '00'+'1': sk=1, (0,1)
p:399, j=8, '00'+'1': sk=1, (0,1)
p:402, j=9, '01'+'00': sk=2, (0,-3)
p:406, j=10, '01'+'01': sk=2, (0,-2)
p:410, j=11, '100'+'111': sk=3, (0,7)
p:416, j=12, '100'+'010': sk=3, (0,-5)
p:422, j=14, '11011'+'00': sk=18, (1,-3)
p:429, j=17, '11100'+'1': sk=33, (2,1)
p:435, j=18, '01'+'01': sk=2, (0,-2)
p:439, j=19, '00'+'0': sk=1, (0,-1)
p:442, j=22, '11100'+'0': sk=33, (2,-1)
p:448, j=23, '00'+'1': sk=1, (0,1)
p:451, j=26, '11100'+'1': sk=33, (2,1)
p:457, j=29, '11100'+'0': sk=33, (2,-1)
p:463, j=45, '11111111001'+'': sk=240,
(15,0) ZRL
p:474, j=48, '11100'+'0': sk=33, (2,-1)
p:480, j=48, '1010'+'': sk=0, (0,0) EOB
Decode dc of Cb[2]
p:484, '10'+'11': delta=3
Decode ac of Cb[2]
p:488, j=1, '100'+'00': sk=2, (0,-3)
p:493, j=3, '1011'+'1': sk=17, (1,1)
p:498, j=4, '01'+'1': sk=1, (0,1)
p:501, j=5, '100'+'10': sk=2, (0,2)
p:506, j=7, '1011'+'0': sk=17, (1,-1)
p:511, j=8, '01'+'0': sk=1, (0,-1)
p:514, j=8, '00'+'': sk=0, (0,0) EOB
Decode dc of Cr[2]
p:516, '110'+'010': delta=-5
Decode ac of Cr[2]
p:522, j=1, '01'+'1': sk=1, (0,1)
p:525, j=2, '01'+'0': sk=1, (0,-1)
p:528, j=3, '01'+'0': sk=1, (0,-1)
p:531, j=4, '01'+'0': sk=1, (0,-1)
p:534, j=5, '01'+'0': sk=1, (0,-1)
p:537, j=8, '11010'+'1': sk=33, (2,1)
p:543, j=8, '00'+'': sk=0, (0,0) EOB
Block number: 3----
Decode dc of Y[3]
p:545, '101'+'0011': delta=-12
Decode ac of Y[3]
p:552, j=1, '100'+'110': sk=3, (0,6)
p:558, j=2, '100'+'100': sk=3, (0,4)
p:564, j=3, '100'+'010': sk=3, (0,-5)
p:570, j=4, '100'+'101': sk=3, (0,5)
p:576, j=5, '11010'+'01110': sk=5, (0,-
17)
p:586, j=6, '100'+'011': sk=3, (0,-4)
p:592, j=7, '00'+'0': sk=1, (0,-1)
p:595, j=8, '01'+'11': sk=2, (0,3)
p:599, j=9, '100'+'000': sk=3, (0,-7)
p:605, j=10, '01'+'01': sk=2, (0,-2)
p:609, j=12, '11011'+'11': sk=18, (1,3)
p:616, j=13, '00'+'0': sk=1, (0,-1)
p:619, j=14, '100'+'001': sk=3, (0,-6)
p:625, j=15, '00'+'0': sk=1, (0,-1)
p:628, j=16, '01'+'10': sk=2, (0,2)
p:632, j=17, '00'+'0': sk=1, (0,-1)
p:635, j=18, '01'+'01': sk=2, (0,-2)
p:639, j=19, '00'+'1': sk=1, (0,1)
p:642, j=21, '1100'+'0': sk=17, (1,-1)
p:647, j=22, '00'+'1': sk=1, (0,1)
```

```
p:650, j=24, '11011'+'00': sk=18, (1,-3)
p:657, j=25, '00'+'1': sk=1, (0,1)
p:660, j=26, '00'+'0': sk=1, (0,-1)
p:663, j=27, '00'+'0': sk=1, (0,-1)
p:666, j=29, '1100'+'0': sk=17, (1,-1)
p:671, j=30, '00'+'1': sk=1, (0,1)
p:674, j=31, '00'+'0': sk=1, (0,-1)
p:677, j=32, '00'+'0': sk=1, (0,-1)
p:680, j=33, '00'+'1': sk=1, (0,1)
p:683, j=33, '1010'+'': sk=0, (0,0) EOB
Decode dc of Cb[3]
p:687, '10'+'01': delta=-2
Decode ac of Cb[3]
p:691, j=1, '100'+'11': sk=2, (0,3)
p:696, j=5, '11011'+'1': sk=49, (3,1)
p:702, j=5, '00'+'': sk=0, (0,0) EOB
Decode dc of Cr[3]
p:704, '10'+'10': delta=2
Decode ac of Cr[3]
p:708, j=1, '100'+'01': sk=2, (0,-2)
p:713, j=2, '01'+'1': sk=1, (0,1)
p:716, j=5, '11010'+'0': sk=33, (2,-1)
p:722, j=8, '11010'+'0': sk=33, (2,-1)
p:728, j=8, '00'+'': sk=0, (0,0) EOB
```