

Appendix 1

Example Output

The following code...

```
# include your function definitions here

def main():
    print()
    print('GIF Image Viewer')
    print()

    file_name = 'squares.gif'

    data, info = load_file(file_name)
    for i in range(len(data)):
        print(hex(data[i]))
    print(type(data))
    print()

    # extract GIF signature
    signature = extract_header(data)
    print(signature)
    print()

    # extract screen descriptor
    scn_w, scn_h, scn_gc_fl, scn_cr, scn_sort_fl, scn_gc_size, scn_bcolour_i,
scn_px_ratio = extract_screen_descriptor(data)
    print('screen width: ', end='')
    print(scn_w)
    print('screen height: ', end='')
    print(scn_h)
    print('global color table flag: ', end='')
    print(scn_gc_fl)
    print('colour resolution: ', end='')
    print(scn_cr)
    print('sort flag: ', end='')
    print(scn_sort_fl)
    print('global colour size: ', end='')
    print(scn_gc_size)
    print('background colour index: ', end='')
    print(scn_bcolour_i)
    print('pixel aspect ratio: ', end='')
    print(scn_px_ratio)
    print()

    # extract global color map
    gc_table = extract_global_colour_table(data)

    for i in range(2**(scn_gc_size+1)):
        print("#",end='')
        print(i,end='\t')
        print(gc_table[i][0],end='\t')
        print(gc_table[i][1],end='\t')
        print(gc_table[i][2])
    print(type(gc_table))
    print(type(gc_table[0][0]))
    print()

    # extract image descriptor
```

```

    img_left, img_top, img_w, img_h, img_lc_fl, img_itl_fl, img_sort_fl, img_res,
    img_lc_size = extract_image_descriptor(data)

    print('image left: ', end='')
    print(img_left)
    print('image top: ', end='')
    print(img_top)
    print('image width: ', end='')
    print(img_w)
    print('image height: ', end='')
    print(img_h)
    print('local colour table flag (0: global, 1: local) : ', end='')
    print(img_lc_fl)
    print('interlace flag (0: sequential, 1: interlaced): ', end='')
    print(img_itl_fl)
    print('sort flag (0: unordered, 1: ordered): ', end='')
    print(img_sort_fl)
    print('reserved values: ', end='')
    print(img_res)
    print('local colour table size: ', end='')
    print(img_lc_size)
    print()

    # extract image data
    img = extract_image(data)

    # print image red channel
    print('img red channel:')
    for i in range(len(img)):
        for j in range(len(img[0])):
            print(img[i][j][0], end='\t')
        print()
    print()

    # print image green channel
    print('img green channel:')
    for i in range(len(img)):
        for j in range(len(img[0])):
            print(img[i][j][1], end='\t')
        print()
    print()

    # print image blue channel
    print('img blue channel:')
    for i in range(len(img)):
        for j in range(len(img[0])):
            print(img[i][j][2], end='\t')
        print()
    print()

if __name__ == '__main__':
    main()

```

Should produce the following output...

GIF Image Viewer

0x47
0x49
0x46
0x38
0x37
0x61
0xa
0x0
0xa
0x0
0xa1
0x0
0x0
0xff
0xff
0xff
0xff
0x0
0x0
0x0
0x0
0xff
0x0
0xff
0x0
0x2c
0x0
0x0
0x0
0x0
0xa
0x0
0xa
0x0
0x0
0x2
0x1a
0x84
0x8f
0x10
0x1b
0xe2
0x62
0x58
0x7b
0x91
0x3d
0xa8
0x2c
0x5
0xa3
0xf
0x69
0x78
0x1f
0x13
0x7a
0x20
0x67

```

0x92
0x49
0x52
0x0
0x0
0x3b
<class 'bytes'>
GIF87a

screen width: 10
screen height: 10
global color table flag: 1
colour resolution: 2
sort flag: 0
global colour size: 1
background colour index: 0
pixel aspect ratio: 0

#0  255  255  255
#1  255  0    0
#2  0    0    255
#3  0    255  0
<class 'list'>
<class 'int'>

image left: 0
image top: 0
image width: 10
image height: 10
local colour table flag (0: global, 1: local) : 0
interlace flag (0: sequential, 1: interlaced): 0
sort flag (0: unordered, 1: ordered): 0
reserved values: 0
local table size: 0

img red channel:
255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 0  0  0  0  255 255 255 255 255
255 0  0  0  0  255 255 255 255 255
255 0  0  0  0  255 255 255 255 255
255 0  0  0  0  255 255 255 255 255
255 255 255 255 255 255 255 255 255 255

img green channel:
255 255 255 255 255 255 255 255 255 255
255 0  0  0  0  0  0  0  0  255
255 0  0  0  0  0  0  0  0  255
255 0  0  0  0  0  0  0  0  255
255 0  0  0  0  0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 0  0  0  0  255
255 255 255 255 255 255 255 255 255 255

img blue channel:
255 255 255 255 255 255 255 255 255 255
255 0  0  0  0  255 255 255 255 255
255 0  0  0  0  255 255 255 255 255
255 0  0  0  0  255 255 255 255 255

```

[illegible]