

```
1: #!/usr/bin/python
2: import numpy as np
3: import pylab
4:
5:
6: def gaussian(x, y, Lx, Ly):
7:     return sum([
8:         sum([
9:             np.exp(-(x + (n * Lx))**2 + (y + (m * Ly))**2) / (2 * (25**2)))
10:            for n in range(-1, 1)
11:        ])
12:        for m in range(-1, 1)
13:    ])
14:
15: def part_a():
16:     img_data = np.loadtxt('blur.txt')
17:     pylab.imshow(img_data)
18:     pylab.set_cmap('Greys_r')
19:     pylab.show()
20:
21:
22: def part_b():
23:     img_data = np.loadtxt('blur.txt')
24:     shape = img_data.shape
25:     blur = [[gaussian(x, y, shape[0], shape[1])
26:             for x in range(shape[0])]
27:            for y in range(shape[1])]
28:     pylab.imshow(blur)
29:     pylab.set_cmap('Greys_r')
30:     pylab.show()
31:
32:
33: def part_c():
34:     img_data = np.loadtxt('blur.txt')
35:     shape = img_data.shape
36:     blur = [[gaussian(x, y, shape[0], shape[1])
37:             for x in range(shape[0])]
38:            for y in range(shape[1])]
39:     fft_img = np.fft.rfft2(img_data)
40:     fft_blur = np.fft.rfft2(blur)
41:     fft_unblured = fft_img
42:     for i, row in enumerate(fft_unblured):
43:         for j, elem in enumerate(row):
44:             if fft_blur[i][j] > 1e-3:
45:                 fft_unblured[i][j] /= (fft_blur[i][j])
46:     unblured = np.fft.irfft2(fft_unblured)
47:     pylab.imshow(unblured)
48:     pylab.set_cmap('Greys_r')
49:     pylab.show()
50:
51:
52: def main():
53:     part_a()
54:     part_b()
55:     part_c()
56:
57:
58: if __name__ == "__main__":
59:     main()
```