1 Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.linalg import solve


f = lambda x: 1 / (1 + x**2)

def composite_trap(a, b, n, f):
    h = (b-a) / n
    x = np.linspace(a, b, n)
    I = h * (0.5 * (f(x[0]) + f(x[-1])) + np.sum(f(x[1:-1]))
        )
    return I

def composite_simpsons(a, b, n, f):
    h = (b-a) / n
    x = np.linspace(a, b, n)
    print(x.size)
    I = (h / 3) * (f(x[0]) + f(x[-1]) + 4*np.sum(f(x[1::2]))
         + 2*np.sum(f(x[2:-1:2])))
    return I

a, b = -5, 5

n_trap = 1291 # calculated from error formulas
n_simps = 70

I_trap = composite_trap(a, b, n_trap, f)
I_simps = composite_simpsons(a, b, n_simps, f)
I_scipy, err = quad(f, a, b)
I_scipy_lowtol, err = quad(f, a, b, epsabs=1e-4)

print(f'The approximate integral using the composite
    Trapezoidal method is {I_trap}')
print(f'The approximate integral using the composite
    Simpsons method is {I_simps}')
print(f'The approximate integral using the built in scipy
    quad is {I_scipy}')
print(f'The approximate integral using the built in scipy
    with tol={1e-4} is {I_scipy_lowtol}')

f2 = lambda x: x*np.cos(1/x)
a2 = 0
b2 = 1
n2 = 5

I2_simps = composite_simpsons(a2, b2, n2, f2)
print(f'The approximate integral with composite Simpsons
    method is {I2_simps}')
```

```
'''question 3'''

A = np.array([[1, 2*np.sqrt(2), 8],
              [1, 4, 16],
              [1, 1, 1]])
B = np.array([0,
              0,
              1])

X = solve(A, B)
print(X)
```