```python
import numpy as np
import matplotlib.pyplot as plt
import sys

# 2.4e

# mu = 0.060, gamma = 0.06531584901232429
# mu = 0.061, gamma = 0.059478472832076835
# mu = 0.062, gamma = 0.05033007132854761
# mu = 0.063, gamma = 0.040706178156078014
# mu = 0.064, gamma = 0.03071758583638221
# mu = 0.065, gamma = 0.019671518608965637

# mu = 0.06591, gamma = 0.004898948029599108
# mu = 0.06592, gamma = 0.004620674420854347
# mu = 0.06593, gamma = 0.0045464915511753004
# mu = 0.06594, gamma = 0.004219778084911094
# mu = 0.06595, gamma = 0.003887937972174111
# mu = 0.06596, gamma = 0.0037182289028526564
# mu = 0.06597, gamma = 0.003512188654658859
# mu = 0.06598, gamma = 0.003105918259005926
# mu = 0.06599, gamma = 0.002830588306209588

mu1 = np.array([0.060, 0.061, 0.062, 0.063, 0.064, 0.065])
gammas = np.array([0.06531584901232429, \
        0.059478472832076835, \
        0.05033007132854761, \
        0.040706178156078014, \
        0.03071758583638221, \
        0.019671518608965637])

mu_c = 0.066
mu_array = np.abs(mu1-mu_c)
coef = np.polyfit(np.log(mu_array),np.log(gammas),1)
poly = np.poly1d(coef)
# print(poly)

u = (np.sqrt(((mu1*(mu1+4)+9)*mu1**2+5)) + 2*mu1-1)/(mu1 + 2)
Tmu_estimate = np.log(1/gammas)/u

# fig, ax = plt.subplots(figsize=(7,7))
# ax.plot(np.log(mu_array),np.log(gammas))
# ax.plot(mu_array,gammas)
# ax.set_xscale('log')
# ax.set_yscale('log')
# plt.show()

# Answer: a = 0.6813, A = 0.7669


# 2.4f
a = 0.6813
A = 0.7669

mu2 = np.linspace(0.06,0.065,50)
mu_c = 0.066
mu_abs = np.abs(mu2-mu_c)
gammas = mu2**a + A
u = (np.sqrt(((mu2*(mu2+4)+9)*mu2**2+5)) + 2*mu2-1)/(mu2 + 2)
```

```python
Tmu_theory = np.log(1/gammas)/u

fig, ax = plt.subplots(figsize=(6,6))
ax.plot(mu_abs, Tmu_theory, label='Estimate')
# ax.plot(mu_array, Tmu_estimate, label='Numerical')
ax.set_ylabel('$T_{\mu}$')
ax.set_xlabel('$|\mu-\mu_{c}|$')
# ax.set_xscale('log')
# ax.set_yscale('log')

plt.legend(loc="upper left")
plt.savefig('24f.png', bbox_inches='tight')
plt.show()
```