```python
import numpy as np
from matplotlib import pyplot as plt

def f1(x):
    return x**5 - 6*x**3 + np.pi*x - 1

def f2(x):
    return np.exp(-x/3) + x**4 - 7*(x**2)*np.log(x)

def f3(x):
    return x**12 + 2**x - 2.5

def ROOT_FALSEPOSITION(b,a,f,err):
    #Find point where the line between the two bounds crosses x axis
    c = b - ((f(b))*(a-b))/((f(a))-(f(b)))
    z = 0
    clist = []
    #Start loop that depends on how close f(c) is to 0
    while np.abs(f(c)) > err:
        #Loop choosing to replace a or b with c value depending on
        #which side has both a positive and negative f(x)
        if f(a)*f(b) < 0:
            b=c
        else:
            a=c
        #Create a list containing all of the c values to analyze later
        clist = np.append(clist,c)
        c = b - ((f(b))*(a-b))/((f(a))-(f(b)))
        z += 1
    iterations = np.arange(1,z+1,1)
    rel_err = np.abs(clist/c - 1)*100

    plt.close('all')
    plt.figure(1)
    plt.subplot(211)
    plt.plot(iterations, clist, 'bo--')
    plt.xlabel("Number of Iterations", fontsize=10, fontstyle='italic')
    plt.ylabel("Estimate of Root", fontsize=10, fontstyle='italic')
    plt.title("How False Position Method Approaches Actual Root",fontsize=14)
    plt.subplots_adjust(hspace = .7)

    plt.subplot(212)
    plt.plot(iterations, rel_err, 'ro--')
    plt.xlabel("Number of Iterations", fontsize=10, fontstyle='italic')
    plt.ylabel("Relative Error", fontsize=10, fontstyle='italic')
    plt.title("Relative Error Progression",fontsize=14)
    plt.show()
    plt.savefig("function4")
    print(c, z)
    return c, z
```

```
#ROOT_FALSEPOSITION(1.5,2.5,f1,0.0001) #root = 2.3467, iterations = 9
#ROOT_FALSEPOSITION(0,3,f1,0.0001) #root = 2.3467, iterations = 139
#ROOT_FALSEPOSITION(0,3,f2,0.0001) #Blows up because of the log value
#ROOT_FALSEPOSITION(0,3,f3,0.0001) #Bounces around as there are multiple roots and the sl
```