```
# hw6
# Cam Brown
# April 18, 2020

import qc
import math
import numpy as np
import matplotlib.pyplot as plt
import cmath

def check_error():
    errors = []
    deciles =[]
    norms = []
    iterations = 1000000
    for i in range(0, iterations):
        state = qc.uniform(2)
        errors.append(state[0]*state[3] - state[1]*state[2])

    for item in errors:
      norm = math.sqrt(math.pow(item.real,2) + math.pow(item.imag, 2))
      norms.append(norm)
    norms.sort()

    # reports breakpoints
    for i in range(0, 10):
        deciles.append(norms[math.floor(i*(iterations / 10))])
    for item in deciles:
      print(item)

    def mean(lst):
        return sum(lst)/len(lst)
    # plots norms histogram
    mu = mean(norms)
    sigma = np.std(norms)
    num_bins = 1000

    plt.hist(norms, density=1, bins=num_bins)
    plt.show()

def main():
    # this code displays a histogram of 1 million normed errors, plotted from 0 to .
5
    # if you want to run, you might consider lowering the number of iterations
    # The curve looks cool, see attached to hw6.pdf
    check_error()

if __name__ == '__main__':
    main()
```