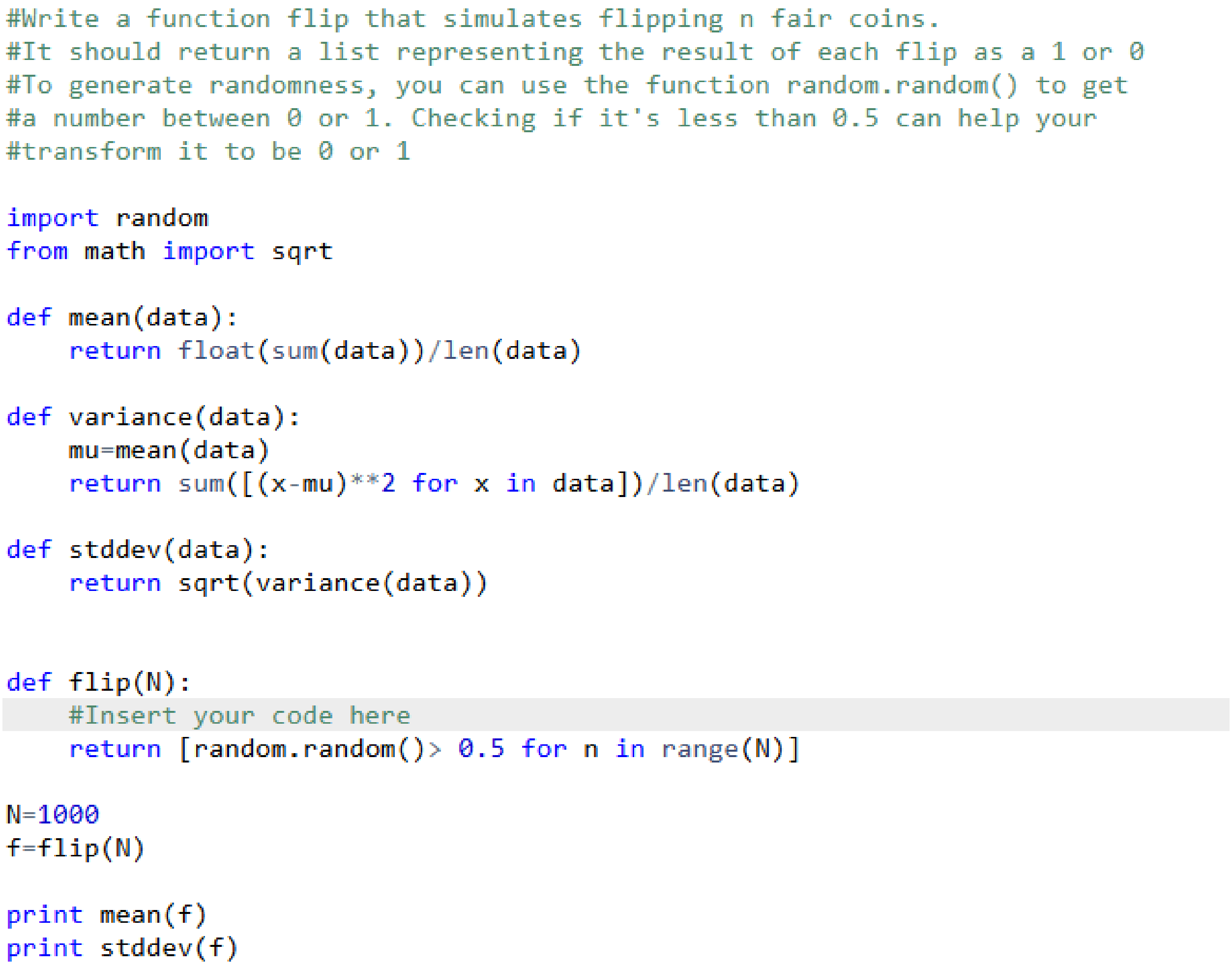
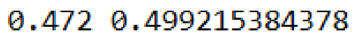
Central Limit Theorem Programming

Programming Flips



```
#Write a function flip that simulates flipping n fair coins.
#It should return a list representing the result of each flip as a 1 or 0
#To generate randomness, you can use the function random.random() to get
#a number between 0 or 1. Checking if it's less than 0.5 can help your
#transform it to be 0 or 1

import random
from math import sqrt

def mean(data):
    return float(sum(data))/len(data)

def variance(data):
    mu=mean(data)
    return sum([(x-mu)**2 for x in data])/len(data)

def stddev(data):
    return sqrt(variance(data))


def flip(N):
    #Insert your code here
    return [random.random()> 0.5 for n in range(N)]

N=1000
f=flip(N)

print mean(f)
print stddev(f)
```

Output:



```
 0.472 0.499215384378
```

Sets Of Flips

```
#Write a function sample that simulates N sets of coin flips and
#returns a list of the proportion of heads in each set of N flips
#It may help to use the flip and mean functions that you wrote before
import random
from math import sqrt
from plotting import *

def mean(data):
    return float(sum(data))/len(data)

def variance(data):
    mu=mean(data)
    return sum([(x-mu)**2 for x in data])/len(data)

def stddev(data):
    return sqrt(variance(data))

def flip(N):
    return [random.random()>0.5 for x in range(N)]

def sample(N):
    #Insert your code here
    return [mean(flip(N)) for n in range(N)]

N=1000
outcomes=sample(N)
histplot(outcomes,nbins=30)

print mean(outcomes)
print stddev(outcomes)
```

Output:

```
0.500088 0.0160242396387
```