```python
"""
CU Roll No.          :
CU Registration No. : xxxx
Description         : Monte Carlo Method (Estimation of Pi)
Author              : AKB
"""

import numpy as np
import matplotlib.pyplot as plt
import time

# store starting time
begin = time.time()

# Count points under circle and Total Points
n, N = 0, 100

# MC Step
for i in np.arange(N):

    x = np.random.uniform(-0.5, 0.5) # Create (x,y) pair
    y = np.random.uniform(-0.5, 0.5)

    if(x**2 + y**2 <= 0.5**2): # Check whether inside circle
       n += 1
       plt.plot(x,y,'co', ms=4)
    else:
       plt.plot(x,y,'mx', ms=4)

    #plt.pause(0.01)

# Estimate Pi (pi/4 = n/N)
print ('Points within circle = ',n,' Total Points = ',N,' Pi = ', 4*float(n)/N)

# store starting time
end = time.time()

# total time taken
print("Total runtime of the program is ", round(abs(end-begin)))

plt.title('N = '+str(N)+r', $\pi = $'+str(4*float(n)/N), fontsize=14)
plt.grid()
plt.xlabel('X', fontsize=12); plt.xticks(fontsize = 12)
plt.ylabel('Y', fontsize=12); plt.yticks(fontsize = 12)
plt.show()


"""
Result:
Points within circle =  79      Total Points =  100      Pi =  3.16
Points within circle =  776     Total Points =  1000     Pi =  3.104
Points within circle =  7825    Total Points =  10000    Pi =  3.13
Points within circle =  78707   Total Points =  100000   Pi =  3.14828
Points within circle =  784992  Total Points =  1000000  Pi =  3.139968
Points within circle =  7853966 Total Points =  10000000 Pi =  3.1415864
"""
```