

## 1. Script output

The script returned the following text:

```
1 C:/code/WUST-BDA/Statistical_physics/Drunkard_walk/main.py
2 [3.1667126108947747, 3.4557336934434053, 3.7434852423911065,
   4.117885981908679, 4.605993990443323, 5.026982792888792,
   5.529735150981464, 6.079065388692574, 6.63340442306965,
   7.282400634406213, 8.107910288107535, 8.80181254060776,
   9.673467870417516, 10.639056191222979, 11.708831702608078,
   12.983019185074017, 14.223607102278944, 15.62758983336842,
   17.307680232775276, 18.856038734580498, 20.58884253084665,
   22.755861795150718, 25.165925565335364, 27.24604965421593,
   30.37445084277245, 32.955837358501455, 36.510220638062435,
   39.843805737404146, 44.420286212045056, 48.13841516128257,
   52.14624354639556, 58.221294291350134, 64.05963570424046,
   70.61593638804203, 77.33900326846732, 84.01074768956649,
   92.19038995470189, 102.5165493432158, 111.95203434882279,
   120.85019314837689, 136.89695316273477, 148.44490721287815,
   164.93681710412628, 180.643733149091, 197.6451350758728,
   215.95873810327748, 239.9906450369264, 263.83502723019933,
   286.148879939377, 313.120321978373]
3 [ 0.50068648 -0.0077662 ]
4 Execution time: 316.87809705734253
5
6 Process finished with exit code 0
```

The most noteworthy is the line number 3: the A and B values of linear regression of the form  $y = Ax + B$

The script have also generated the attached plots.

```

1  from random import uniform
2  from multiprocessing import Pool
3  from time import time
4  from numpy import logspace, std, polyfit, log, arange
5  import matplotlib.pyplot as plt
6  import winsound
7
8
9  def drunkard(steps=30000):
10     position = 0
11     for i in range(steps):
12         if uniform(0, 1) > .5:
13             position += 1
14         else:
15             position -= 1
16     return position
17
18
19  def generate_std_report(steps=30000, drunkards=10000):
20     pool = Pool()
21     args = [steps]*drunkards #we need an iterable of arguments for
22         drunkard function
23     data = pool.map(drunkard, args)
24     pool.close()
25     pool.join()
26     return std(data)
27
28
29  def generate_hist(lower=1000, upper=5000, drunkards=50000):
30     pool = Pool()
31     l_result = pool.map(drunkard, [lower]*drunkards)
32     u_result = pool.map(drunkard, [upper]*drunkards)
33     plt.hist(x=l_result, bins=upper, histtype='step')
34     plt.hist(x=u_result, bins=upper, histtype='step')
35     plt.show()
36
37  if __name__ == '__main__':
38     TS = time()
39     space = logspace(1, 5, 50)
40     results = []
41     for i in space:
42         results.append(generate_std_report(steps=int(i)))
43     print(results)
44     linreg = polyfit(log(space), log(results), 1)
45     print(linreg)
46     plt.scatter(log(space), log(results))
47     plt.plot(log(space), log(space)*linreg[0]+linreg[1])
48     plt.title('İ for N steps')
49     plt.xlabel('log N')
50     plt.ylabel('log İ')
51     plt.show()
52     generate_hist()
53     print("Execution time: "+str(time()-TS))
54     # winsound.Beep(500, 300)

```