

```

1 import sys
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 # Function to know if we have a CCW turn
6 def CCW(p1, p2, p3):
7     if (p3[1]-p1[1])*(p2[0]-p1[0]) >= (p2[1]-p1[1])*(p3[0]-p1[0]):
8         return True
9     return False
10
11 # Main function:
12 def GiftWrapping(S):
13     plt.figure() # Define figure
14     index = 0
15     n = len(S)
16     P = [None] * n
17     l = np.where(S[:,0] == np.min(S[:,0]))
18     pointOnHull = S[l[0][0]]
19     i = 0
20     while True:
21         P[i] = pointOnHull
22         endpoint = S[0]
23         for j in range(1,n):
24             if (endpoint[0] == pointOnHull[0] and endpoint[1] == pointOnHull[1]) or not CCW(S[j],P[i],endpoint):
25                 endpoint = S[j]
26         i = i + 1
27         pointOnHull = endpoint
28     J = np.array([P[k] for k in range(n) if P[k] is not None])
29     plt.clf() # Clear plot
30     plt.plot(J[:,0],J[:,1], 'b-', picker=5) # Plot lines
31     plt.plot(S[:,0],S[:,1],".r") # Plot points
32     plt.axis('off') # No axis
33     plt.show(block=False) # Close plot
34     plt.pause(0.0000001) # Mini-pause before closing plot
35     index += 1
36     if endpoint[0] == P[0][0] and endpoint[1] == P[0][1]:
37         break
38     for i in range(n):
39         if P[-1] is None:
40             del P[-1]
41     P = np.array(P)
42
43     # Plot final hull
44     plt.clf()
45     plt.plot(P[:,0],P[:,1], 'b-', picker=5)
46     plt.plot([P[-1,0],P[0,0]],[P[-1,1],P[0,1]], 'b-', picker=5)
47     plt.plot(S[:,0],S[:,1],".r")
48     plt.axis('off')
49     plt.show(block=False)
50     plt.pause(0.0000001)
51     return P
52
53 def main():
54     try:
55         N = int(sys.argv[1])
56     except:
57         N = int(input("Introduce N: "))
58
59     # By default we build a random set of N points with coordinates in [0,300)x[0,300):
60     P = np.array([(np.random.randint(0,300),np.random.randint(0,300)) for i in range(N)])
61     L = GiftWrapping(P)
62
63     # We use the predefined figure
64     plt.plot(L[:,0],L[:,1], 'b-', picker=5)
65     plt.plot([L[-1,0],L[0,0]],[L[-1,1],L[0,1]], 'b-', picker=5)
66     plt.plot(P[:,0],P[:,1],".r")
67     plt.axis('off')
68     plt.show()
69
70 if __name__ == '__main__':
71     main()
72

```

Introduce N: 10

