

```

import random as rd
import matplotlib.pyplot as plt

nbPoints = 30
nMax = 2 * nbPoints
liste_ordonnees = [rd.randint(0, nMax) for k in range(nbPoints)]
abscisse = 0
liste_abscisses = []
for k in range(nbPoints):
    abscisse += rd.randint(0, 4)
    liste_abscisses.append(abscisse)
tab = [liste_abscisses, liste_ordonnees]

def dessiner(tab, n):
    for k in range(n):
        plt.plot([tab[0][k]], [tab[1][k]], marker = 'o')
        plt.annotate(k, xy = (tab[0][k], tab[1][k]))

def _3_orient(tab, i, j, k):
    xi, xj, xk = tab[0][i], tab[0][j], tab[0][k]
    yi, yj, yk = tab[1][i], tab[1][j], tab[1][k]
    determinant = (xj - xi)*(yk - yi) - (xk - xi)*(yj - yi)
    if determinant == 0:
        return 0
    if determinant > 0:
        return 1
    else:
        return -1

def _10_majES(tab, es, i):
    depiler = True
    while depiler:
        j = es.pop()
        if es == []:
            depiler = False
        else:
            k = es[-1]
            if _3_orient(tab, i, j, k) > 0:
                depiler = False
    es.append(j)
    es.append(i)

def _11_majEI(tab, ei, i):
    depiler = True
    while depiler:
        j = ei.pop()
        if ei == []:
            depiler = False
        else:
            k = ei[-1]
            if _3_orient(tab, i, j, k) < 0:
                depiler = False
    ei.append(j)
    ei.append(i)

def _12_convGraham(tab, n):
    es, ei = [0], [0]
    for i in range(1, nbPoints):
        _10_majES(tab, es, i)
        _11_majEI(tab, ei, i)
    while es != []:
        ei.append(es.pop())
    return ei

def tracer(enveloppe):
    x = [tab[0][k] for k in enveloppe]
    y = [tab[1][k] for k in enveloppe]

```

```
plt.plot(x, y)

dessiner(tab, nbPoints)
enveloppe = _12_convGraham(tab, nbPoints)
tracer(enveloppe)
print(enveloppe)
```