```
import random as rd
import matplotlib.pyplot as plt

nbPoints, nMax = 30, 100
tab = [[rd.randint(-nMax, nMax) for k in range(nbPoints)] for k in
range(2)]

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 _1_plusBas(tab, n):
    indice_min = 0
    ordonnee_min = tab[1][0]
    abscisse = tab[0][0]
    for j in range(1, n):
        if tab[1][j] < ordonnee_min or tab[1][j] ==  ordonnee_min and
tab[0][j] < abscisse:
            indice_min = j
            abscisse = tab[0][j]
            ordonnee_min = tab[1][j]
    return indice_min

# Q2  +1 pour le premier, -1 pour le 2e

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 _5_prochainPoint(tab, n, i):
    if i == 0:
        max_provisoire = 1
    else:
        max_provisoire = 0
    for k in range(1, n):
        if _3_orient(tab, i, max_provisoire, k) < 0:
          max_provisoire = k
    return max_provisoire

# Q5  variante
def _5_prochainPoint_bis(tab, n, i):
    prochain = i-n+1
    for k in range(i-n+2, i):
        if _3_orient(tab, i, prochain, k) <= 0:
            prochain = k
    return prochain % n

def _7_convJarvis(tab, n):
    depart = _1_plusBas(tab, n)
    reponse = [depart]
    prochain = _5_prochainPoint_bis(tab, n, depart)
    while prochain != depart:
        reponse.append(prochain)
        prochain = _5_prochainPoint(tab, n, prochain)
    return reponse

def tracer(enveloppe):
    enveloppe.append(enveloppe[0])
    x = [tab[0][k] for k in enveloppe]
    y = [tab[1][k] for k in enveloppe]
    plt.plot(x, y)

dessiner(tab, nbPoints)
enveloppe = _7_convJarvis(tab, nbPoints)
tracer(enveloppe)
print(enveloppe)
```