

```

import numpy as np

def ConjugateGradientAlgo(lite, Q, b, c, x):
    """
    f(x) = 0.5 * X.T * Q * X - X.T * b + c
    """
    g = np.matmul(Q, x) - b
    d = g
    for _ in range(lite):
        print(x, d, g)
        alpha = - (g.T * d) / (np.matmul(np.matmul(d.T, Q), d))
        x = x + alpha * d
        g = np.matmul(Q, x) - b
        beta = (np.matmul(g.T, Q) * d) / (np.matmul(np.matmul(d.T, Q), d))
        d = - g + np.matmul(beta, d)

Q = np.array([[5, -3], [-3, 2]])
b = np.array([0, 1])
c = -7
x = np.array([0, 0])
lite = 3
ConjugateGradientAlgo(lite, Q, b, c, x)

```

python3 10_9.py

[0 0] [0 -1] [0 -1]

[0. 0.5] [3.75 2.25] [-1.5 0.]

[0.70754717 0.5] [4.01299395 8.17337131] [2.03773585 -2.12264151]