

```
#normalizacija tacaka
def normalize(tacke):
    x = 0.0
    y = 0.0

    udaljenost = 0.0

    for i in range(len(tacke)):
        x = x + float(tacke[i][0]) / float(tacke[i][2])
        y = y + float(tacke[i][1]) / float(tacke[i][2])

    #x i y su affine koordinate tezista tacaka
    x = x / float(len(tacke))
    y = y / float(len(tacke))

    for i in range(len(tacke)):
        tmp1 = tacke[i][0]/tacke[i][2] - x
        tmp2 = tacke[i][1]/tacke[i][2] - y

        udaljenost = udaljenost + math.sqrt(tmp1**2 + tmp2**2)

    udaljenost = udaljenost / float(len(tacke))

    k = math.sqrt(2) / udaljenost

    S = np.array([[k, 0, -k*x], [0, k, -k*y], [0, 0, 1]])

    return S
```

```
def dlt_normalize(originali, slike):

    #matrice T i Tp su matrice normalizacije originalnih tacaka i njihovih slika
    T = normalize(originali)
    Tp = normalize(slike)

    #kopije originalnih tacaka i njihovih slika
    originali_copy = copy.deepcopy(originali)
    slike_copy = copy.deepcopy(slike)

    for i in range(len(originali)):
        #normalizacija originalnih tacaka
        [x, y, z] = np.dot(T, [originali[i][0], originali[i][1], originali[i][2]])

        originali_copy[i][0] = float(x)
        originali_copy[i][1] = float(y)
        originali_copy[i][2] = float(z)

    for i in range(len(slike)):
        #normalizacija slika
        [x, y, z] = np.dot(Tp, [float(slike[i][0]), float(slike[i][1]), float(slike[i][2])])

        slike_copy[i][0] = float(x)
        slike_copy[i][1] = float(y)
        slike_copy[i][2] = float(z)

    #primena DLT algoritma na normalizovane originalne tacke i njihove slike
    Pp = DLT(originali_copy, slike_copy)

    #odgovarajuca matrica preslikavanja
    P = np.dot(np.linalg.inv(Tp), Pp)
    P = np.dot(P, T)

    return P
```

D) DLT

c) Modifikovani DLT

Unesite broj tacaka:

5

OK

Unesite homogene koordinate originalnih tacaka i njihovih slika:

-3 1 1

-1 0 1

0 1 1

0 2 1

-3 -2 1

OK

1 1 1

3 1 1

3 2 1

1 2 1

3 0.33 1

Poredjenje sa DLT-om

Odgovarajuca matrica preslikavanja:

[[-0.10289062 -0.20395668 0.66453256]

[-0.15399627 0.20575539 0.10153571]

[-0.20524339 0.10297387 0.05057573]]