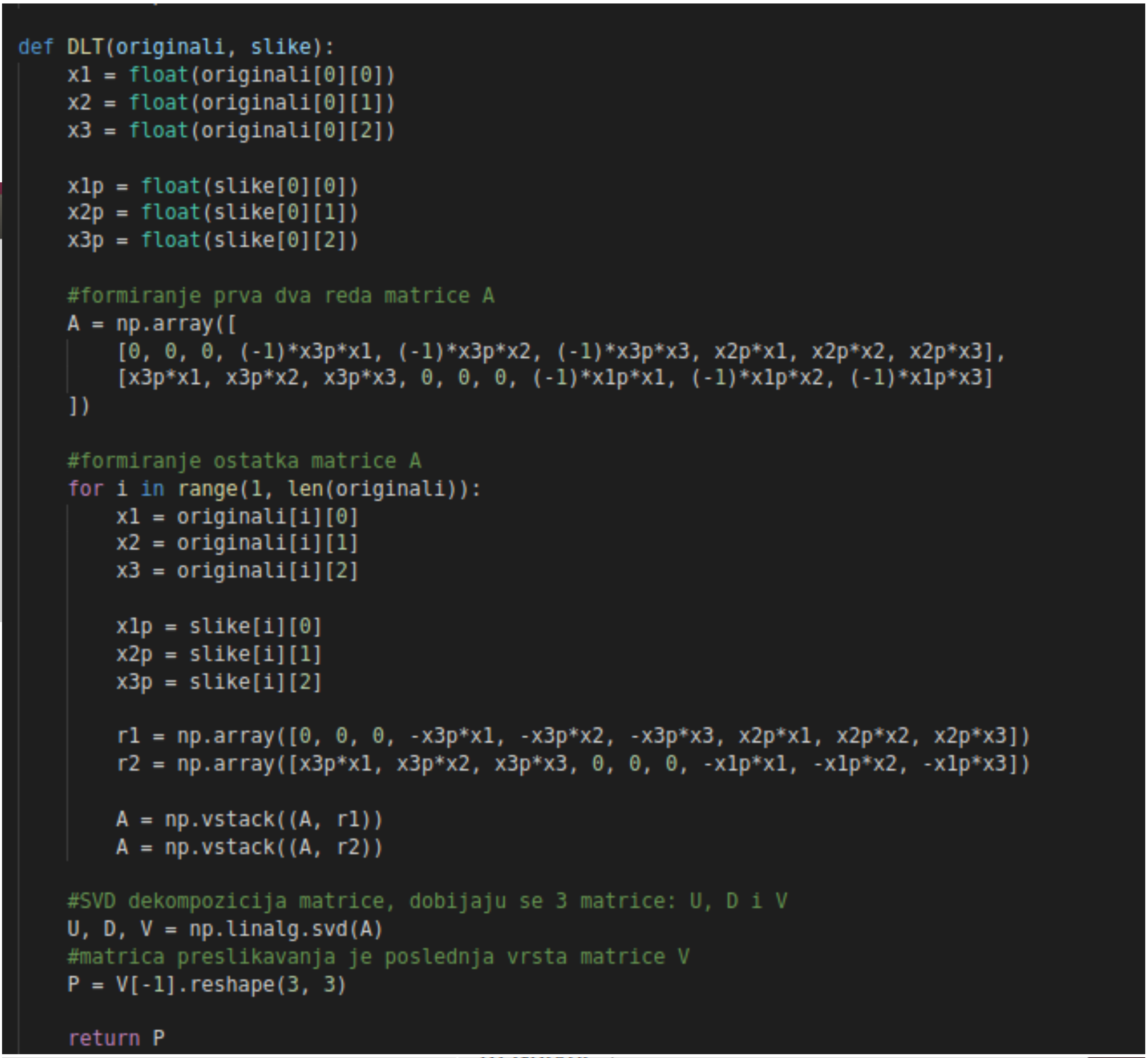
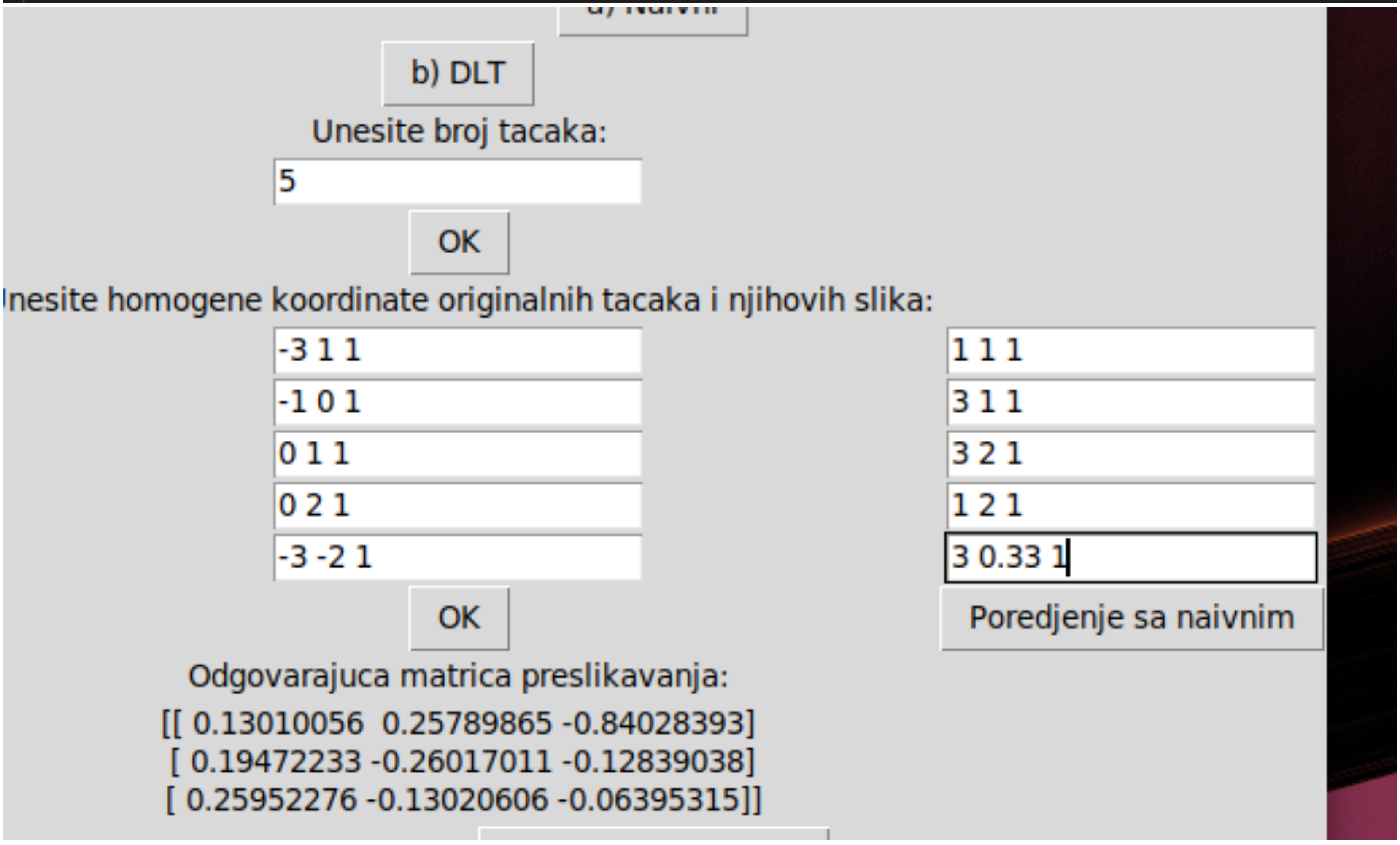


```
def DLT(originali, slike):
    x1 = float(originali[0][0])
    x2 = float(originali[0][1])
    x3 = float(originali[0][2])

    x1p = float(slike[0][0])
    x2p = float(slike[0][1])
    x3p = float(slike[0][2])

    #formiranje prva dva reda matrice A
    A = np.array([
        [0, 0, 0, (-1)*x3p*x1, (-1)*x3p*x2, (-1)*x3p*x3, x2p*x1, x2p*x2, x2p*x3],
        [x3p*x1, x3p*x2, x3p*x3, 0, 0, 0, (-1)*x1p*x1, (-1)*x1p*x2, (-1)*x1p*x3]
    ])

    #formiranje ostatka matrice A
    for i in range(1, len(originali)):
        x1 = originali[i][0]
        x2 = originali[i][1]
        x3 = originali[i][2]

        x1p = slike[i][0]
        x2p = slike[i][1]
        x3p = slike[i][2]

        r1 = np.array([0, 0, 0, -x3p*x1, -x3p*x2, -x3p*x3, x2p*x1, x2p*x2, x2p*x3])
        r2 = np.array([x3p*x1, x3p*x2, x3p*x3, 0, 0, 0, -x1p*x1, -x1p*x2, -x1p*x3])

        A = np.vstack((A, r1))
        A = np.vstack((A, r2))

    #SVD dekompozicija matrice, dobijaju se 3 matrice: U, D i V
    U, D, V = np.linalg.svd(A)
    #matrica preslikavanja je poslednja vrsta matrice V
    P = V[-1].reshape(3, 3)

    return P
```



Unesite broj tacaka:

5

nesite homogene koordinate originalnih tacaka i njihovih slika:

-3 1 1	1 1 1
-1 0 1	3 1 1
0 1 1	3 2 1
0 2 1	1 2 1
-3 -2 1	3 0.33 1

Odgovarajuca matrica preslikavanja:

```
[[ 0.13010056  0.25789865 -0.84028393]
 [ 0.19472233 -0.26017011 -0.12839038]
 [ 0.25952276 -0.13020606 -0.06395315]]
```