**Використання модуля NUMPY**

Ви будете писати програму modelMatrix(inMatrix, sequence) для розрахунку матриці трансформації. Наприклад, припустимо, що в просторі заданий вектор , яким буде результат застосування наступних трансформацій? (R(θ) означає обертання 2D на θ градусів і Т(х, у) означає переміщення 2D на осі х і у, відповідно, S(х, у) - масштабування).

Вигляд матриць трансформацій для окремих операцій наступний (для простору ):

Тобто результатом застосування виразів для вектору буде вираз:

inMatrix – словник, ключами якого є відповідні позначення для окремих матриць трансформацій (R – обертання, Т – переміщення, S – масштабування, V – вхідний вектор) наступного вигляду:

inMatrix = {'R':(33),'T':(1.0,2.0),'S':(2.0,1.2),'V':(1,2)}

sequence – рядок, в якому представлена послідовність застосування окремих матриць трансформацій, наприклад 'SRT' (для наведеного вище прикладу).

Функція повертає результат у вигляді списку значень

def modelMatrix(inMatrix, sequence):

"""

Finds a transformation vector in 2D.

inMatrix: dictionary of transformation matrix

inMatrix = {'R':(angle),'T':(,),'S':(,),'V':(x,y)}

sequence – string of operation, like 'TRS', T – first, S - last

returns: tuple of transformed vector

"""

### TODO

>>> inMatrix = {'R':(33),'T':(1.0,2.0),'S':(2.0,1.2),'V':(1,2)}

>>> modelMatrix(inMatrix, 'SRT')

(-1.0, 5.33, 1)

>>> inMatrix = {'R':(16),'T':(1.0,2.0),'S':(1.0,2.0),'V':(1,2)}

>>> modelMatrix(inMatrix, 'TRS')

(0.86, 6.12, 1)

**Обов’язково** використайте numpy для матричних операцій.