```
# let's import our library
import scipy.linalg
import numpy as np


# Encoding this states to numbers as it
# is easier to deal with numbers instead
# of words.
state = ["A", "E"]

# Assigning the transition matrix to a variable
# i.e a numpy 2d matrix.
MyMatrix = np.array([[0.6, 0.4], [0.7, 0.3]])

# Simulating a random walk on our Markov chain
# with 20 steps. Random walk simply means that
# we start with an arbitrary state and then we
# move along our markov chain.
n = 20

# decide which state to start with
StartingState = 0
CurrentState = StartingState

# printing the stating state using state
# dictionary
print(state[CurrentState], "--->", end=" ")

while n-1:
    # Deciding the next state using a random.choice()
    # function,that takes list of states and the probability
    # to go to the next states from our current state
    CurrentState = np.random.choice([0, 1], p=MyMatrix[CurrentState])

    # printing the path of random walk
    print(state[CurrentState], "--->", end=" ")
    n -= 1
print("stop")

# Let us find the stationary distribution of our
# Markov chain by Finding Left Eigen Vectors
# We only need the left eigen vectors
MyValues, left = scipy.linalg.eig(MyMatrix, right=False, left=True)

print("left eigen vectors = \n", left, "\n")
print("eigen values = \n", MyValues)

# Pi is a probability distribution so the sum of
# the probabilities should be 1 To get that from
# the above negative values we just have to normalize
pi = left[:, 0]
pi_normalized = [(x/np.sum(pi)).real for x in pi]
pi_normalized
```