# Problem



# Solution

import numpy

def arrays(arr):

    # complete this function

    # use numpy.array

    return(numpy.array(arr[::-1], float))

arr = input().strip().split(' ')

result = arrays(arr)

print(result)

# Note

arr[::] = equivalent to arr[start: stop : step] means from which index you want to start array index and where you want to stop looping step defines by how many places you want to jump through array

ex. arr = [0,1,2,3,4,5,6,7,8,9] arr[::] will print the complete array because start is by default 0 stop is by default is end of array and step is by default +1

arr[::2] will print arr = [0 2 4 6 8] because we're jumping two step each time

similarly arr[::-1] will reverse the array because in python we can point the last element in array by index -1 here -1 means start from last index and decrease each time by -1 till 0th index

Slice notation:

X[a:b:c]

means "count in increments of c starting at a inclusive, up to b exclusive". If a is not specified, it is defaulted to 0, if b is not specified, it is defaulted to len(X). If c is not specified, it is defaulted to 1.

If a and b both are not specified, you are talking about the complete string. When c is -1, you are going backward. So effective double colon here means skipping a and b in slice notation. You can read more details and perform some experiments with different values of a, b and c and skipping some of them to understand things better.