# Write a Python program to interchange first and last elements in a list by using all the approaches

# Given a list, write a Python program to swap first and last element of the list.

# Examples:

# Input : [12, 35, 9, 56, 24]

# Output : [24, 35, 9, 56, 12]

# Input : [1, 2, 3]

# Output : [3, 2, 1]

# Approach #1: Find the length of the list and simply swap the first element with (n-1)th element.

def appr\_1(list\_1):

    for i in list\_1:

        first = list\_1[0]

        len\_of\_list = len(list\_1)

        last = list\_1[len\_of\_list - 1]

    first, last = last, first

    return list\_1

if \_\_name\_\_ == '\_\_main\_\_':

    print()

    print("Approach\_1: Original= [12, 35, 9, 56, 24]", "Swapped=", appr\_1([12, 35, 9, 56, 24]))

    print("Approach\_1: Original= [1, 2, 3]", "Swapped=", appr\_1([1, 2, 3]))

    print()

# Approach #2: The last element of the list can be referred as list[-1].

# Therefore, we can simply swap list[0] with list[-1].

def appr\_2(list\_2):

    first = list\_2[0]

    last = list\_2[-1]

    list\_2[0] = last

    list\_2[-1] = first

    return list\_2

if \_\_name\_\_ == '\_\_main\_\_':

    print("Approach\_2: Original= [1, 2, 3, 4]", "Swapped=", appr\_2([1, 2, 3, 4]))

    print()

# Approach #3: Swap the first and last element is using tuple variable.

# Store the first and last element as a pair in a tuple variable,

# say get, and unpack those elements with first and last element in that list.

# Now, the First and last values in that list are swapped.

def appr\_3(list\_3):

    first = list\_3[0]

    last = list\_3[-1]

    first\_last = first, last

    # uppack

    (i1, i2) = first\_last

    list\_3[0], list\_3[-1] = i2, i1

    return list\_3

if \_\_name\_\_ == "\_\_main\_\_":

    print("Approach\_3: Original= [12, 35, 9, 56, 24]", "Swapped=", appr\_3([12, 35, 9, 56, 24]))

    print("Approach\_3: Original= [1, 2, 3]", "Swapped=", appr\_3([1, 2, 3]))

    print()

# Approach #4: Using \* operand.

# This operand proposes a change to iterable unpacking syntax, allowing to specify

# a “catch-all” name which will be assigned a list of all items not assigned to a “regular” name.

def appr\_4(list\_4):

    (first, \*middle, last) = list\_4

    # print(first)

    # print(middle)

    # print(last)

    list\_4 = (last, \*middle, first)

    return list\_4

if \_\_name\_\_ == "\_\_main\_\_":

    print("Approach\_4: Original= [6, 7, 8, 9]", "Swapped=", appr\_4([6, 7, 8, 9]))

    print()

# Approach #5: Swap the first and last elements is to use the inbuilt function list.pop().

# Pop the first element and store it in a variable. Similarly, pop the last element and

# store it in another variable. Now insert the two popped element at each other’s original position.

def appr\_5(list\_5):

    first = list\_5.pop(0)

    last = list\_5.pop()

    list\_5.insert(0, last)

    list\_5.insert(3, first)

    # print(first, last)

    # print(list\_5)

    # print(list\_5)

    return list\_5

if \_\_name\_\_ == "\_\_main\_\_":

    print("Approach\_5: Original= [6, 7, 8, 9]", "Swapped=", appr\_5([6,7,8,9]))

    print()

Results

