Algorithm And Data Structure

print ("Linear Search")

def linear\_search (list, target):

*"""*

*Return the index position of the target if found, else return None*

*"""*

*for* i *in* range (0, len(list)):

*if* list[i] == target:

*return* i

*return* None

def verify1(index):

*if* index is not None:

print ("Target found at index: ", index)

*else*:

print ("Target not found in list")

num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

result = linear\_search (num, 12)

verify1(result)

result = linear\_search (num, 3)

verify1(result)

print ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

print ("binary Search")

def binary\_search (list, target):

first = 0

last = len(list) - 1

*while* first <= last:

midpoint = (first + last)//2

*if* list[midpoint] == target:

*return* midpoint

*elif* list[midpoint] < target:

first = midpoint + 1

*else*:

last = midpoint - 1

*return* None

def verify(index):

*if* index is not None:

print ("Target found at index: ", index)

*else*:

print ("Target not found in list")

num = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

result = binary\_search (num, 12)

verify(result)

result = binary\_search (num, 3)

verify(result)

print ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

print ("Recursive binary Search")

def recursive\_binary\_search (list, target):

*if* len(list) == 0:

*return* False

*else*:

midpoint= (len(list)) // 2

*if* list[midpoint] == target:

*return* True

*else*:

*if* list[midpoint] < target:

*return* recursive\_binary\_search(list[midpoint+1:], target)

*else*:

*return* recursive\_binary\_search (list[: midpoint], target)

def verify(result):

print ("Target found: ", result)

num = [1,2,3,4,5,6,7,8]

result = recursive\_binary\_search (num, 12)

verify(result)

result = recursive\_binary\_search (num, 6)

verify(result)