**Binary Search (Without Using Recursion)**

def binary\_search(arr, low, high, ele):  
 mid = 0  
  
 while low <= high:  
  
 mid = (high + low) // 2  
  
 # If x is greater, ignore left half  
 if arr[mid] < ele:  
 low = mid + 1  
  
 # If x is smaller, ignore right half  
 elif arr[mid] > ele:  
 high = mid - 1  
  
 # means x is present at mid  
 else:  
 return mid+1  
  
 # If we reach here, then the element was not present  
 return -1  
  
  
# Test array  
arr = [2, 3, 4, 10, 40]  
Element = 4  
low = 0  
high = len(arr) - 1  
  
# Function call  
result = binary\_search(arr,low, high, Element)  
  
if result != -1:  
 print("Element is present at Posistion", str(result))  
else:  
 print("Element is not present in array")

**Binary Search (Using Recursion)**

def binary\_search(arr, low, high, ele):  
 # Check base case  
 if high >= low:  
  
 mid = (high + low) // 2  
  
 # If element is present at the middle itself  
 if arr[mid] == x:  
 return mid+1  
  
 # If element is smaller than mid, then it can only  
 # be present in left subarray  
 elif arr[mid] > x:  
 return binary\_search(arr, low, mid - 1, x)  
  
 # Else the element can only be present in right subarray  
 else:  
 return binary\_search(arr, mid + 1, high, x)  
  
 else:  
 # Element is not present in the array  
 return -1  
  
  
# Test array  
arr = [2, 3, 4, 10, 40]  
x = 3  
  
# Function call  
result = binary\_search(arr, 0, len(arr) - 1, x)  
  
if result != -1:  
 print("Element is present at Position", str(result))  
else:  
 print("Element is not present in array")