def binary\_search(arr,key):

start=0

end=len(arr)-1

mid=(start+end)//2

while(start<=end):

if (arr[mid]==key):

return mid

elif(key>arr[mid]):

start= mid+1

else:

end=mid-1

mid=(start+end)//2

return -1

arr=[4,5,6,7,8,9,10,11,12]

key=7

result=binary\_search(arr,key)

print(result)

class Solution:

def binarysearch(self, arr, k):

start, end = 0, len(arr) - 1

result = -1

while start <= end:

mid = (start + end) // 2

if arr[mid] == k:

result = mid

end = mid - 1

elif arr[mid] < k :

start = mid + 1

else:

end = mid - 1

return result

def quick\_sort(arr):

if len(arr)<=1:

return arr

left=[]

right=[]

equal=[]

pvt=arr[-1]

for i in arr:

if(i<pvt):

left.append(i)

elif(i>pvt):

right.append(i)

else:

equal.append(i)

print("pvt",pvt)

print("left sub array",left)

print("right sub array",right)

print("equal array",equal)

return quick\_sort(left)+equal+quick\_sort(right)

arr=[23,63,44,57,12,45,36]

sorted\_arr=quick\_sort(arr)

print(sorted\_arr)

def insertion\_sort(arr):

for i in range(1,len(arr)):

key=arr[i]

j=i-1

while j>=0 and key<arr[j]:

arr[j+1]=arr[j]

j-=1

arr[j+1]=key

return arr

arr=[8,3,14,9]

sorted\_arr=insertion\_sort(arr)

print(arr)

def bubble\_sort(arr):

n=len(arr)

for i in range(n):

for j in range(0,n-i-1):

if arr[j]>arr[j+1]:

arr[j],arr[j+1]=arr[j+1],arr[j]

return arr

arr=[8,3,45,9]

sorted\_sort=bubble\_sort(arr)

print(arr)