```
resource url?
QUESTION ----------------
Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping
the adjacent elements if they are in wrong order. You read an list of numbers.
You need to arrange the elements in ascending order and print the result. The
sorting should be done using bubble sort.
------------------ANSWER
n = int(input())
arr = list(map(int, input().split()))

def bubbleSort(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 = bubbleSort(arr)
print(' '.join(map(str, arr)))
------------------------------
QUESTION ----------------
To find the frequency of numbers in a list and display in sorted
order.Constraints: 1<=n, arr[i]<=100 Input: 1 68 79 4 90 68 1 4 5 output: 1 2 4
2 5 1 68 2 79 1 90 1
------------------ANSWER
arr = list(map(int, input().split()))

freq = {}
for num in arr:
    if num in freq:
        freq[num] += 1
    else:
        freq[num] = 1

for key, value in sorted(freq.items()):
    print(key, value)
------------------------------
QUESTION ----------------
Given an list, find peak element in it. A peak element is an element that is
greater than its neighbors.
------------------ANSWER
t=int(input())
num=input()
num=num.split()
L=[]
S=[]
for i in num:
    L.append(int(i))
for i in range(t):
    if(i==0):
        if(L[i]>=L[i+1]):
            S.append(L[i])
    elif(0<i<(t-1)):
        if(L[i-1]<=L[i]>=L[i+1]):
            S.append(L[i])
    elif(i==t-1):
        if(L[i]>=L[i-1]):
            S.append(L[i])
for i in S:
    print(i,end=" ")
------------------------------
QUESTION ----------------
Write a Python program to sort a list of elements using the merge sort
```

algorithm.

```
------------------ANSWER
n = int(input())
arr = list(map(int, input().split()))

def mergeSort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left_half = arr[:mid]
    right_half = arr[mid:]
    return merge(mergeSort(left_half), mergeSort(right_half))

def merge(left, right):
    merged = []
    left_index = 0
    right_index = 0
    while left_index < len(left) and right_index < len(right):
        if left[left_index] <= right[right_index]:
            merged.append(left[left_index])
            left_index += 1
        else:
            merged.append(right[right_index])
            right_index += 1
    merged.extend(left[left_index:])
    merged.extend(right[right_index:])
    return merged

arr = mergeSort(arr)
print(' '.join(map(str, arr)))
------------------------------
```

QUESTION ----------------

An list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.Input FormatThe first line contains a single integer n , the length of listThe second line contains n space-separated integers, list[i].The third line contains integer k.Output FormatPrint Yes or No.Sample Input70 1 2 4 6 5 3 1 Sample OutputYes

```
------------------ANSWER
t=int(input())
num=input()
num=num.split()
p=int(input())
L=[]
for i in num:
    L.append(int(i))
flag=0
for i in range(t):
    for j in range(i+1,t):
        if(i+j==p):
            flag=1
            break
if(flag==0):
    print("Yes")
else:
    print("No")
------------------------------
```