Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program to implement merge sort using iterative approach on the given list of values.

For example:

Test	Input	Result
Merge_Sort(S)	6 4 2 3 1 6 5	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]
Merge_Sort(S)	5 2 6 4 3 1	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def Merge_Sort(S):

    if len(S)>1:
        mid=len(S)//2
        left=S[:mid]
        right=S[mid:]
        Merge_Sort(left)
        Merge_Sort(right)
        i=j=k=0
        while(i<len(left) and j<len(right)):
            if(left[i]<right[j]):
                S[k]=left[i]
                i+=1
            else:
                S[k]=right[j]
                j+=1
            k+=1
        while(i<len(left)):
```

	Test	Input	Expected	Got	
✔	Merge_Sort(S)	6 4 2 3 1 6 5	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]	The Original array is: [4, 2, 3, 1, 6, 5] Array after sorting is: [1, 2, 3, 4, 5, 6]	✔
✔	Merge_Sort(S)	5 2 6 4 3 1	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]	The Original array is: [2, 6, 4, 3, 1] Array after sorting is: [1, 2, 3, 4, 6]	✔

	Test	Input	Expected	Got	
✔	Merge_Sort(S)	4 3 5 6 1	The Original array is: [3, 5, 6, 1] Array after sorting is: [1, 3, 5, 6]	The Original array is: [3, 5, 6, 1] Array after sorting is: [1, 3, 5, 6]	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Incorrect

Mark 0.00 out of 20.00

Write a python program to implement the quick sort using recursion on the given list of float values.

For example:

Input	Result
5 6.3 1.2 4.6 5.8 9.7	pivot: 9.7 pivot: 5.8 pivot: 4.6 [1.2, 4.6, 5.8, 6.3, 9.7]
6 2.3 7.8 9.5 4.2 3.6 5.4	pivot: 5.4 pivot: 3.6 pivot: 7.8 [2.3, 3.6, 4.2, 5.4, 7.8, 9.5]

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def partition(arr,start,end):
    pivot=arr[end]
    i=start-1
    for j in range(start,end):
        if(arr[j]<pivot):
            i+=1
            arr[i],arr[j]=arr[j],arr[i]
    i+=1
    arr[i],arr[end]=arr[end],arr[i]


def quick_sort(arr,start,end):
    if(end<=start):
        return
    pivot=partition(arr,start,end)
    quick_sort(arr,start,pivot-1)
    quick_sort(arr,pivot+1,end)
```

	Input	Expected	Got	
✖	5 6.3 1.2 4.6 5.8 9.7	pivot: 9.7 pivot: 5.8 pivot: 4.6 [1.2, 4.6, 5.8, 6.3, 9.7]	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 25, in <module> quick_sort(arr,start,end) File "__tester__.python3", line 16, in quick_sort quick_sort(arr,start,pivot-1) TypeError: unsupported operand type(s) for -: 'NoneType' and 'int'	✖

Testing was aborted due to error.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a Python Program Using a recursive function to calculate the sum of a sequence
For example:

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
def sumOfSequence(num):
    if(num==1):
        return 1
    return num+sumOfSequence(num-1)
n=int(input())
print(sumOfSequence(n))
```

	Input	Expected	Got	
✔	20	210	210	✔
✔	36	666	666	✔
✔	45	1035	1035	✔
✔	58	1711	1711	✔
✔	65	2145	2145	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to implement binary search on the given list of float values using iterative method

For example:

Test	Input	Result
binarySearchAppr(arr, 0, len(arr)-1, x)	5 3.2 6.1 4.5 9.6 8.3 6.1	Element is present at index 2
binarySearchAppr(arr, 0, len(arr)-1, x)	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	Element is present at index 3

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def binarySearchAppr(arr, low, high, x):
    while(low<=high):
        mid=(low+high)//2
        if(arr[mid]==x):
            return mid
        elif(arr[mid]<x):
            low=mid+1
        else:
            high=mid-1
    return -1

n=int(input())
arr=[]
for i in range(n):
    arr.append(float(input()))
x=float(input())
arr.sort()
result=binarySearchAppr(arr, 0, len(arr)-1, x)
```

	Test	Input	Expected	Got	
✔	binarySearchAppr(arr, 0, len(arr)-1, x)	5 3.2 6.1 4.5 9.6 8.3 6.1	Element is present at index 2	Element is present at index 2	✔

	Test	Input	Expected	Got	
✔	`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	`Element is present at index 3`	`Element is present at index 3`	✔
✔	`binarySearchAppr(arr, 0, len(arr)-1, x)`	8 2.1 6.3 5.2 4.2 9.3 6.7 5.6 9.8 7.2	`Element is not present in array`	`Element is not present in array`	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program to implement linear search on the given tuple of string values.

note: As the tuple is immutable convert the list to tuple to perform search

For example:

Input	Result
5 ram john akbar seetha oviya john	Tuple: john found
4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
def search(arr,x):
    for i in range(len(arr)):
        if(arr[i]==x):
            return True
    return False
n=int(input())
arr=[]
for i in range(n):
    arr.append(input())
x=input()
if(search(arr,x)):
    print("Tuple:",x,"found")
else:
    print("Tuple:",x,"not found")
```

	Input	Expected	Got	
✔	5 ram john akbar seetha oviya john	Tuple: john found	Tuple: john found	✔
✔	4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found	Tuple: rakesh not found	✔

	Input	Expected	Got	
✔	6 rose jasmine tulips marigold hibiscus lotus lilly	Tuple: lilly not found	Tuple: lilly not found	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.