Started on	Wednesday, 23 April 2025, 3:25 PM
State	Finished
Completed on	Wednesday, 23 April 2025, 4:02 PM
Time taken	37 mins 12 secs
Grade	80.00 out of 100.00

Question **1**Not answered

Mark 0.00 out of 20.00

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

For example:

Test	Input	Result
Merge_Sort(S)	5	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8]
	10.2	Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]
	21.3	
	3.5	
	7.8	
	9.8	
Merge_Sort(S)	6	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2]
	20.3	Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]
	41.2	
	5.3	
	6.2	
	8.1	
	65.2	

1			
			1.

```
Question 2
Correct
Mark 20.00 out of 20.00
```

Write a python program for a search function with parameter list name and the value to be searched on the given list of float values.

For example:

Test	Input	Result
search(List, n)	5	3.2 Found
	3.2	
	6.1	
	4.5	
	6.2	
	8.5	
	3.2	
search(List, n)	4	6.1 Not Found
	3.2	
	1.5	
	6.4	
	7.8	
	6.1	

```
1 v def search(List,n):
         for i in range(len(List)):
 2 🔻
 3 🔻
            if List[i]==n:
 4
                return 1
 5 🔻
         else:
 6
                 return 0
 7
 8
 9
10
11
12
    List=[]
13
   x=int(input())
14 v for i in range(x):
15 ele=float(input())
16
         List.append(ele)
17
    n=float(input())
18
   ans=search(List,n)
19 v if ans==1:
        print(n,"Found")
20
21 ▼ else:
        print(n,"Not Found")
22
```

	Test	Input	Expected	Got	
~	search(List, n)	5 3.2 6.1 4.5 6.2 8.5 3.2	3.2 Found	3.2 Found	*
~	search(List, n)	4 3.2 1.5 6.4 7.8 6.1	6.1 Not Found	6.1 Not Found	*
~	search(List, n)	7 2.1 3.2 6.5 4.1 5.2 7.1 8.2 9.3	9.3 Not Found	9.3 Not Found	*

Passed all tests! 🗸

Correct

```
Question 3
Correct
Mark 20.00 out of 20.00
```

Write a python program to implement quick sort on the given float array values.

For example:

Input	Result
5 6.9 8.3 2.1 1.5 6.4	<pre>left: [] right: [] left: [] right: [] left: [1.5] right: [6.4] left: [] right: [] left: [1.5, 2.1, 6.4] right: [8.3] [1.5, 2.1, 6.4, 6.9, 8.3]</pre>
6 3.1 2.4 5.6 4.3 6.2 7.8	left: [] right: [] left: [] right: [] left: [] right: [] left: [] right: [7.8] left: [4.3] right: [6.2, 7.8] left: [2.4] right: [4.3, 5.6, 6.2, 7.8] [2.4, 3.1, 4.3, 5.6, 6.2, 7.8]

```
1 def qsort(arr):
 2 🔻
           if arr==[]:
 3
                return []
 4
           pivot=arr[0:1]
          left=qsort([x for x in arr[1:] if x<arr[0]])
right=qsort([x for x in arr[1:] if x>=arr[0]])
print("left: ",left)
print("right: ",right)
 5
 6
 7
 8
           return left+pivot+right
10
     arr=[]
11
12
    n=int(input())
13 🔻
     for i in range(n):
          ele=float(input())
14
           arr.append(ele)
15
     print(qsort(arr))
16
17
18
```

	Input	Expected	Got	
~	5	left: []	left: []	~
	6.9	right: []	right: []	
	8.3	left: []	left: []	
	2.1	right: []	right: []	
	1.5	left: [1.5]	left: [1.5]	
	6.4	right: [6.4]	right: [6.4]	
		left: []	left: []	
		right: []	right: []	
		left: [1.5, 2.1, 6.4]	left: [1.5, 2.1, 6.4]	
		right: [8.3]	right: [8.3]	
		[1.5, 2.1, 6.4, 6.9, 8.3]	[1.5, 2.1, 6.4, 6.9, 8.3]	

	Input	Expected	Got	
~	6	left: []	left: []	~
	3.1	right: []	right: []	
	2.4	left: []	left: []	
	5.6	right: []	right: []	
	4.3	left: []	left: []	
	6.2	right: []	right: []	
	7.8	left: []	left: []	
		right: [7.8]	right: [7.8]	
		left: [4.3]	left: [4.3]	
		right: [6.2, 7.8]	right: [6.2, 7.8]	
		left: [2.4]	left: [2.4]	
		right: [4.3, 5.6, 6.2, 7.8]	right: [4.3, 5.6, 6.2, 7.8]	
		[2.4, 3.1, 4.3, 5.6, 6.2, 7.8]	[2.4, 3.1, 4.3, 5.6, 6.2, 7.8]	
~	8	left: []	left: []	~
	1.2	right: []	right: []	
	1.3	left: []	left: []	
	4.2	right: []	right: []	
	5.3	left: [6.8]	left: [6.8]	
	6.4	right: [9.2]	right: [9.2]	
	7.3	left: []	left: []	
	6.8	right: [6.8, 7.3, 9.2]	right: [6.8, 7.3, 9.2]	
	9.2	left: []	left: []	
		right: [6.4, 6.8, 7.3, 9.2]	right: [6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [5.3, 6.4, 6.8, 7.3, 9.2]	right: [5.3, 6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [1.3, 4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	right: [1.3, 4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	
		[1.2, 1.3, 4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	[1.2, 1.3, 4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	

Passed all tests! 🗸

Correct

Question 4
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 %)

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

Passed all tests! 🗸

Correct

```
Question 5
Correct
Mark 20.00 out of 20.00
```

Write a python program for a search function with parameter list name and the value to be searched on the given list of int values.

For example:

Test	Input	Result
search(List, n)	5	Found
	3	
	4	
	5	
	6	
	7	
	4	
search(List, n)	6	Found
	20	
	34	
	56	
	87	
	96	
	51	
	87	

```
1 v def search(List,n):
         for i in range(len(List)):
    if List[i]==n:
 2 🔻
 3 ▼
                 return 1
 5 🔻
         else:
 6
                 return 0
    List=[]
 7
 8
   x=int(input())
9 for i in range(x):
ele=float(input())
10
11
         List.append(ele)
12 n=float(input())
13
    ans=search(List,n)
14 v if ans==1:
15
         print("Found")
16 v else:
17
        print("Not Found")
```

	Test	Input	Expected	Got	
~	search(List, n)	5 3 4 5 6	Found	Found	*
		4			
*	search(List, n)	6 20 34 56 87 96 51 87	Found	Found	*

	Test	Input	Expected	Got	
~	search(List, n)	4	Not Found	Not Found	~
		30			
		10			
		20			
		50			
		60			

Passed all tests! 🗸

Correct