Started on	Monday, 14 July 2025, 1:25 PM
State	Finished
Completed on	Monday, 14 July 2025, 3:44 PM
Time taken	2 hours 19 mins
Overdue	19 mins 6 secs
Grade	**80.00** out of 100.00

Question **1**

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 %)

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

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

Passed all tests! ✔

Marks for this submission: 20.00/20.00.

Question **2**

Incorrect

Mark 0.00 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	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]
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]

Answer: (penalty regime: 0 %)

```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[0]
    left = [x for x in arr[1:] if x <= pivot]
    right = [x for x in arr[1:] if x > pivot]
    print("left: ", left)
    print("right: ", right)
    return quick_sort(left) + [pivot] + quick_sort(right)

n = int(input())
arr = []
for i in range(n):
    arr.append(float(input()))
print(quick_sort(arr))
```

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

Some hidden test cases failed, too.
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 to implement merge sort without using recursive function on the given list of values.

For example:

Input	Result
7 33 42 9 37 8 47 5	left: [33] Right: [42] left: [9] Right: [37] left: [8] Right: [47] left: [5] Right: [] left: [33, 42] Right: [9, 37] left: [8, 47] Right: [5] left: [9, 33, 37, 42] Right: [5, 8, 47] [5, 8, 9, 33, 37, 42, 47]
6 10 3 5 61 74 92	left: [10] Right: [3] left: [5] Right: [61] left: [74] Right: [92] left: [3, 10] Right: [5, 61] left: [74, 92] Right: [] left: [3, 5, 10, 61] Right: [74, 92] [3, 5, 10, 61, 74, 92]

Answer: (penalty regime: 0 %)

```
def merge(left, right):
    result = []
    while left and right:
        if left[0] <= right[0]:
            result.append(left.pop(0))
        else:
            result.append(right.pop(0))
    result += left
    result += right
    return result

n = int(input())
arr = []
for i in range(n):
    arr.append(int(input()))

lists = [[x] for x in arr]
while len(lists) > 1:
    if len(lists) % 2 != 0:
        lists.append([])
    merged_lists = []
    for i in range(0, len(lists), 2):
```

	Input	Expected	Got	
✔	7 33 42 9 37 8 47 5	left: [33] Right: [42] left: [9] Right: [37] left: [8] Right: [47] left: [5] Right: [] left: [33, 42] Right: [9, 37] left: [8, 47] Right: [5] left: [9, 33, 37, 42] Right: [5, 8, 47] [5, 8, 9, 33, 37, 42, 47]	left: [33] Right: [42] left: [9] Right: [37] left: [8] Right: [47] left: [5] Right: [] left: [33, 42] Right: [9, 37] left: [8, 47] Right: [5] left: [9, 33, 37, 42] Right: [5, 8, 47] [5, 8, 9, 33, 37, 42, 47]	✔
✔	6 10 3 5 61 74 92	left: [10] Right: [3] left: [5] Right: [61] left: [74] Right: [92] left: [3, 10] Right: [5, 61] left: [74, 92] Right: [] left: [3, 5, 10, 61] Right: [74, 92] [3, 5, 10, 61, 74, 92]	left: [10] Right: [3] left: [5] Right: [61] left: [74] Right: [92] left: [3, 10] Right: [5, 61] left: [74, 92] Right: [] left: [3, 5, 10, 61] Right: [74, 92] [3, 5, 10, 61, 74, 92]	✔
✔	5 4 12 6 98 3	left: [4] Right: [12] left: [6] Right: [98] left: [3] Right: [] left: [4, 12] Right: [6, 98] left: [3] Right: [] left: [4, 6, 12, 98] Right: [3] [3, 4, 6, 12, 98]	left: [4] Right: [12] left: [6] Right: [98] left: [3] Right: [] left: [4, 12] Right: [6, 98] left: [3] Right: [] left: [4, 6, 12, 98] Right: [3] [3, 4, 6, 12, 98]	✔

Passed all tests! ✔

Marks for this submission: 20.00/20.00.

Question **4**

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

Answer: (penalty regime: 0 %)

```
def search(List,n):
    if n in List:
        print("Found")
    else:
        print("Not Found")
List=[]
size=int(input())
for i in range(size):
    List.append(int(input()))
n=int(input())
```

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

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

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

Answer: (penalty regime: 0 %)

```
def search(List,n):
    if n in List:
        print(n,"Found")
    else:
        print(n,"Not Found")
List=[]
size=int(input())
for i in range(size):
    List.append(float(input()))
n=float(input())
```

	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

Marks for this submission: 20.00/20.00.