Started on	Saturday, 12 April 2025, 10:53 AM
State	Finished
Completed on	Saturday, 12 April 2025, 12:56 PM
Time taken	2 hours 2 mins
Overdue	2 mins 55 secs
Grade	**80.00** out of 100.00

Question **1**

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 o

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):
    for i in List:
        if i == n:
            print(n, "Found")
            return;
    print(n, "Not Found")

s = int(input())
List = []
for i in range(s):
    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.

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 %)

```
def quicksort(List):
    low = []
    high = []
    mid = []
    if len(List)>1:
        search = List[-1]
        for i in List:
            if i < search :
                low.append(i)
            elif i > search:
                high.append(i)
            else:
                mid.append(i)
        print("pivot: ", mid[-1])
        return quicksort(low)+mid+quicksort(high)
    else:
        return List


n = int(input())
List = []
for i in range(n):
```

	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]	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]	pivot: 5.4 pivot: 3.6 pivot: 9.5 [2.3, 3.6, 4.2, 5.4, 7.8, 9.5]	✖

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 linear search on the given tuple of float values.

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

For example:

Input	Result
5 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found
6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	Tuple: 6.2 found

Answer: (penalty regime: 0 %)

```
n = int(input())
li = []
for i in range(n):
    li.append(float(input()))
s = float(input())

tup = tuple(li)

def search(tup,s):
    for i in tup:
        if i == s:
            print(f"Tuple: {s} found")
            return
    print(f"Tuple: {s} not found")

search(tup,s)
```

	Input	Expected	Got	
✔	5 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found	Tuple: 6.4 found	✔
✔	6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	Tuple: 6.2 found	Tuple: 6.2 found	✔
✔	4 2.1 3.2 6.5 4.5 3.5	Tuple: 3.5 not found	Tuple: 3.5 not found	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Incorrect

Mark 20.00 out of 20.00

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

For example:

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

Answer: (penalty regime: 0 %)

```
def Merge_Sort(S):
    low = []
    high = []
    mid = []
    if len(List)>1:
        search = List[-1]
        for i in List:
            if i < search :
                low.append(i)
            elif i > search:
                high.append(i)
            else:
                mid.append(i)
        return quicksort(low)+mid+quicksort(high)
    else:
        return List



n = int(input())
S = []
for i in range(n):
```

	Test	Input	Expected	Got

✖	`Merge_Sort(S)`	`5`
		`10.2`
		`21.3`
		`3.5`
		`7.8`
		`9.8`
			`The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8]`
			`Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]`
				`The Original array i`
				`3.5, 7.8, 9.8]`
				
				`***Run error***`
				`Traceback (most rece`
				`File "__tester__.p`
				`<module>`
				`S = Merge_Sort(S`
				`File "__tester__.p`
				`Merge_Sort`
				`if len(List)>1:`
				`NameError: name 'Lis`

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 **5**

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 sum_of_seq(n):
    if n<=0:
        return 0
    return n+sum_of_seq(n-1)

n = sum_of_seq(int(input()))
print(n)
```

	Input	Expected	Got	
✔	20	210	210	✔
✔	36	666	666	✔

	Input	Expected	Got	
✔	45	1035	1035	✔
✔	58	1711	1711	✔
✔	65	2145	2145	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.