Started on	Thursday, 6 March 2025, 2:24 PM
State	Finished
Completed on	Thursday, 6 March 2025, 2:54 PM
Time taken	29 mins 55 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 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.

	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:  [4.0, 2.0, 3.0, 1.0, 6.0, 5.0]`
				`Array after sorting is:  [1.0, 2.0, 3.0, 4.0, 5.0, 6.0]`
					✖

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 **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 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 %)

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

```
def search(List,x):
    for i in List:
        if(i==x):
            return True
    return False

List=[]
n=int(input())
for i in range(n):
    List.append(eval(input()))
x=eval(input())
if(search(List,x)):
    print(f"{x} Found".format(x))
else:
    print(f"{x} Not Found".format(x))
```

	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	✓

	Test	Input	Expected	Got	
✓	`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 **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 sum(n):
    if(n==1):
        return n
    else:
        return n+(sum(n-1))
n=int(input())
print(sum(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 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 %)

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

```
def search(Tuple,n):
    for i in Tuple:
        if(i==n):
            return True
    return False


Tuple=[]
x=int(input())
for i in range(x):
    num=eval(input())
    Tuple.append(num)
n=eval(input())
if(search(Tuple,n)):
    print("Tuple: {} found".format(n))
else:
    print("Tuple: {} not found".format(n))
```

	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 **5**

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
	`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 %)

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

```
def qsort(L):
    if L==[]:
        return[]
    pivot=L[0:1]
    left=qsort([x for x in L[1:]if x<L[0]])
    right=qsort([x for x in L[1:]if x>=L[0]])
    print("left: ",left)
    print("right: ",right)
    return left+pivot+right
list1=[]
n=int(input())
for i in range(n):
    list1.append(float(input()))
print(qsort(list1))
```

	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: [] 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]	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]	✓
✓	8 1.2 1.3 4.2 5.3 6.4 7.3 6.8 9.2	left: [] right: [] left: [] right: [] left: [6.8] right: [9.2] left: [] right: [6.8, 7.3, 9.2] left: [] right: [6.4, 6.8, 7.3, 9.2] left: [] right: [5.3, 6.4, 6.8, 7.3, 9.2] left: [] right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2] left: [] 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]	left: [] right: [] left: [] right: [] left: [6.8] right: [9.2] left: [] right: [6.8, 7.3, 9.2] left: [] right: [6.4, 6.8, 7.3, 9.2] left: [] right: [5.3, 6.4, 6.8, 7.3, 9.2] left: [] right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2] left: [] 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]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.