Started on	Friday, 10 January 2025, 3:07 PM
State	Finished
Completed on	Friday, 10 January 2025, 3:31 PM
Time taken	24 mins 1 sec
Grade	**100.00** out of 100.00

Question **1**

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(list,n):
    for i in list:
        if i == n:
            print("Tuple:",n,"found")
            break
    else:
        print("Tuple:",n,"not found")
a=int(input())
List=()
for i in range(a):
    List+=(input(),)
#print(len(List))
n=input()
search(List,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	✔

	Input	Expected	Got	
✔	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 **2**

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 L
    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.

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 recur_sum(n):
   if n <= 1:
       return n
   else:
       return n + recur_sum(n-1)
num = int(input())
print(recur_sum(num))
```

	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 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(list,n):
    for i in list:
        if i == n:
            print("Tuple:",n,"found")
            break
    else:
        print("Tuple:",n,"not found")
a=int(input())
List=()
for i in range(a):
    List+=(input(),)
#print(len(List))
n=input()
search(List,n)
```

	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.

Question **5**

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 %)

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

```
def merge_sort_iterative(arr):
    stack = [[val] for val in arr]

    while len(stack) > 1:
        temp_stack = []
        for i in range(0, len(stack), 2):
            left = stack[i]
            right = stack[i + 1] if i + 1 < len(stack) else []
            merged = merge(left, right)
            temp_stack.append(merged)
            print(f"left:  {left}")
            print(f"Right:  {right}")
        stack = temp_stack

    return stack[0]

def merge(left, right):
    merged = []
```

	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! ✔

Correct

Marks for this submission: 20.00/20.00.