Started on	Wednesday, 21 May 2025, 3:44 PM
State	Finished
Completed on	Wednesday, 21 May 2025, 4:09 PM
Time taken	24 mins 56 secs
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a Python Program to print factorial of a number recursively.

For example:

Input	Result
5	Factorial of number 5 = 120
6	Factorial of number 6 = 720

Answer: (penalty regime: 0 %)

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

```
def fact(n):
    if n==0:
        return 1
    return n*fact(n-1)
n=int(input())
print("Factorial of number",n,"=",fact(n))
```

	Input	Expected	Got	
✓	5	Factorial of number 5 = 120	Factorial of number 5 = 120	✓
✓	6	Factorial of number 6 = 720	Factorial of number 6 = 720	✓
✓	7	Factorial of number 7 = 5040	Factorial of number 7 = 5040	✓
✓	8	Factorial of number 8 = 40320	Factorial of number 8 = 40320	✓

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 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, n):
        for i in range(len(List)):
                if List[i] == n:
                        return "Found"
        return "Not Found"

List = []
x=int(input())
for i in range(x):
    List.append(float(input()))
n =float(input())
print(n, search(List,n))
```

	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	✓

	Test	Input	Expected	Got	

✓	`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 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.

```
def Merge_Sort(S):
    if len(S)>1:
        mid = len(S)//2
        left_half=S[:mid]
        right_half=S[mid:]
        Merge_Sort(left_half)
        Merge_Sort(right_half)
        i=j=k=0
        while i<len(left_half) and j<len(right_half):
            if left_half[i]<right_half[j]:
                S[k]=left_half[i]
                i+=1
            else:
                S[k]=right_half[j]
                j+=1
            k+=1
        while i<len(left_half):
            S[k]=left_half[i]
```

	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, 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]`
				`The Original array is:  [2, 6, 4, 3, 1]`
				`Array after sorting is:  [1, 2, 3, 4, 6]`
					✓

	Test	Input	Expected	Got	

✓	`Merge_Sort(S)`	4
		3
		5
		6
		1
			`The Original array is:  [3, 5, 6, 1]`
			`Array after sorting is:  [1, 3, 5, 6]`
				`The Original array is:  [3, 5, 6, 1]`
				`Array after sorting is:  [1, 3, 5, 6]`
					✓

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 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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def partition(arr,low,high):
    pivot=arr[high]
    i=low-1
    for j in range(low,high):
        if arr[j]<=pivot:
            i+=1
            arr[i],arr[j]=arr[j],arr[i]
    arr[i+1],arr[high]=arr[high],arr[i+1]
    return i+1
def quick_sort(arr,low,high):
    if low<high:
        pivot_index=partition(arr,low,high)
        print("pivot: ",arr[pivot_index])
        quick_sort(arr,low,pivot_index-1)
        quick_sort(arr,pivot_index+1,high)
n=int(input())
arr=[]
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: 7.8 [2.3, 3.6, 4.2, 5.4, 7.8, 9.5]	✓

	Input	Expected	Got	
✓	4 3.2 6.4 8.7 1.5	pivot: 1.5 pivot: 3.2 pivot: 6.4 [1.5, 3.2, 6.4, 8.7]	pivot: 1.5 pivot: 3.2 pivot: 6.4 [1.5, 3.2, 6.4, 8.7]	✓

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 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 linearSearch(tup, x):
    for i in range(len(tup)):
        if(tup[i]==x):
            return "found"
    return "not found"

arr=[]
n = int(input())
for i in range(n):
    arr.append(float(input()))
x = float(input())
arr.sort()
ans = linearSearch(tuple(arr),x)
print(f"Tuple: {x} {ans}")
```

	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.