

Started on	Wednesday, 21 May 2025, 3:12 PM
State	Finished
Completed on	Wednesday, 21 May 2025, 3:28 PM
Time taken	15 mins 17 secs
Grade	80.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

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

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 %)

```

1 def Merge_Sort(S):
2     if(len(S)>1):
3         mid = len(S)//2
4         left = S[:mid]
5         right = S[mid:]
6         Merge_Sort(left)
7         Merge_Sort(right)
8         i = j = k = 0
9         while(i < len(left) and j < len(right)):
10            if(left[i] < right[j]):
11                S[k] = left[i]
12                i = i +1
13            else:
14                S[k] = right[j]
15                j = j+1
16                k = k+1
17            while(i<len(left)):
18                S[k] = left[i]
19                i = i+1
20                k = k+1
21            while(j<len(right)):
22                S[k] = right[j]
```

	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 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]	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]	✓
✓	Merge_Sort(S)	4 2.3 6.1 4.5 96.5	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	✓

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 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 %)

```

1 def Factorial(n):
2     if n==0 or n==1:
3         return 1
4     else:
5         return n * Factorial(n-1)
6 n=int(input())
7
8 print("Factorial of number",n,"=",Factorial(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 **3**

Not answered

Mark 0.00 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 %)

1	
---	--

	Input	Expected	Got	
✖	5 ram john akbar seetha oviya john	Tuple: john found	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 11, in <module> List.append(float(input())) ValueError: could not convert string to float: 'ram'	✖

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 4

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	left: []
6.9	right: []
8.3	left: []
2.1	right: []
1.5	left: [1.5]
6.4	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	left: []
3.1	right: []
2.4	left: []
5.6	right: []
4.3	left: []
6.2	right: []
7.8	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 %)

```

1 def quickSort(arr):
2     if arr==[]:
3         return arr
4     pivot=arr[0:1]
5     left=quickSort([x for x in arr[1:] if x<pivot[0]])
6     right=quickSort([x for x in arr[1:] if x>=pivot[0]])
7     print("left: ",left)
8     print("right: ",right)
9     return left+pivot+right
10
11 l=[float(input()) for i in range(int(input()))]
12 s=quickSort(l)
13 print(s)

```

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

	Input	Expected	Got	
✓	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! ✓



Marks for this submission: 20.00/20.00.

Question 5

Correct

Mark 20.00 out of 20.00

Write a python program to implement binary search on the given list of float values using iterative method

For example:

Test	Input	Result
binarySearchAppr(arr, 0, len(arr)-1, x)	5 3.2 6.1 4.5 9.6 8.3 6.1	Element is present at index 2
binarySearchAppr(arr, 0, len(arr)-1, x)	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	Element is present at index 3

Answer: (penalty regime: 0 %)

```

1 def binarySearchAppr (arr, start, end, x):
2     if end >= start:
3         mid = (start +end)//2
4         if arr[mid] == x:
5             return mid
6         elif arr[mid] > x:
7             return binarySearchAppr(arr, start, mid-1, x)
8         else:
9             return binarySearchAppr(arr,mid+1,end,x)
10    else:
11        return -1
12 arr=[]
13 n=int(input())
14 for i in range(n):
15     arr.append(input())
16 arr = sorted(arr)
17 x =input()
18 result = binarySearchAppr(arr,0,len(arr)-1,x)
19 if result != -1:
20     print ("Element is present at index "+str(result))
21 else:
22     print ("Element is not present in array")

```

	Test	Input	Expected	Got	
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	5 3.2 6.1 4.5 9.6 8.3 6.1	Element is present at index 2	Element is present at index 2	✓
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	Element is present at index 3	Element is present at index 3	✓

	Test	Input	Expected	Got	
✓	binarySearchAppr(arr, 0, len(arr)-1, x)	8 2.1 6.3 5.2 4.2 9.3 6.7 5.6 9.8 7.2	Element is not present in array	Element is not present in array	✓

Passed all tests! ✓



Marks for this submission: 20.00/20.00.