

<b>Started on</b>	Thursday, 20 March 2025, 2:54 PM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 20 March 2025, 3:24 PM
<b>Time taken</b>	30 mins 9 secs
<b>Grade</b>	<b>100.00</b> 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 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 %)

```

1 def Merge_Sort(arr):
2     width = 1
3     n = len(arr)
4
5     while width < n:
6         for i in range(0, n, 2 * width):
7             left = arr[i:i + width]
8             right = arr[i + width:i + 2 * width]
9             arr[i:i + 2 * width] = merge(left, right)
10            width *= 2
11
12 def merge(left, right):
13     result = []
14     i = j = 0
15
16     while i < len(left) and j < len(right):
17         if left[i] < right[j]:
18             result.append(left[i])
19             i += 1
20         else:
21             result.append(right[j])
22             j += 1

```

	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 **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 %)

```

1 def qsort(L):
2     if L==[]:
3         return []
4     pivot = L[0:1]
5     left=qsort([x for x in L[1:] if x<L[0]])
6     right=qsort([x for x in L[1:] if x>=L[0]])
7     print("left: ",left)
8     print("right: ",right)
9     return left+pivot+right
10 list=[]
11 n=int(input())
12 for i in range(n):
13     list.append(float(input()))
14 print(qsort(list))

```

Input	Expected	Got
-------	----------	-----

	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 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
7 n = int(input())
8 print(f"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 **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 %)

```
1 def linear_search(tup, target):
2     for item in tup:
3         if item == target:
4             return True
5     return False
6
7 n = int(input())
8 lst = []
9 for _ in range(n):
10     lst.append(input())
11 tup = tuple(lst)
12 target = input()
13 if linear_search(tup, target):
14     print(f"Tuple: {target} found")
15 else:
16     print(f"Tuple: {target} not found")
```

	Input	Expected	Got	
✓	5 ram john akbar seetha oviya john	Tuple: john found	Tuple: john found	✓

	Input	Expected	Got	
✓	4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found	Tuple: rakesh not found	✓
✓	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 for a search function with parameter list name and the value to be searched on the given list of int values.

**For example:**

Test	Input	Result
search(List, n)	5	Found
	3	
	4	
	5	
	6	
	7	
	4	
search(List, n)	6	Found
	20	
	34	
	56	
	87	
	96	
	51	
	87	

**Answer:** (penalty regime: 0 %)

```

1 def search(List, n):
2     for item in List:
3         if item == n:
4             return True
5     return False
6 num_elements = int(input())
7 List = []
8
9 for _ in range(num_elements):
10     List.append(int(input()))
11
12 n = int(input())
13
14 if search(List, n):
15     print("Found")
16 else:
17     print("Not Found")

```

	Test	Input	Expected	Got	
✓	search(List, n)	5	Found	Found	✓
		3			
		4			
		5			
		6			
		7			
		4			

	Test	Input	Expected	Got	
✓	search(List, n)	6 20 34 56 87 96 51 87	Found	Found	✓
✓	search(List, n)	4 30 10 20 50 60	Not Found	Not Found	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 20.00/20.00.