
Started on Thursday, 3 April 2025, 10:03 AM

State Finished

Completed on Thursday, 3 April 2025, 10:46 AM

Time taken 42 mins 42 secs

Grade **80.00** out of 100.00

Question 1

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 %)

```

1 List=[]
2 s=int(input())
3 for i in range(s):
4     List.append(input())
5 n=input()
6 def search(List,n):
7     for i in List:
8         if i==n:
9             print(n,"Found")
10            break;
11 else:
12     print(n,"Not Found")

```

	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 **2**

Correct

Mark 20.00 out of 20.00

Write a Python Program to print the fibonacci series upto n_terms using Recursion.

For example:

Input	Result
10	Fibonacci series: 0 1 1 2 3 5 8 13 21 34
5	Fibonacci series: 0 1 1 2 3
7	Fibonacci series: 0 1 1 2 3 5 8

Answer: (penalty regime: 0 %)

```
1 def fibonacci(n):
2     if n <= 0:
3         return 0
4     elif n == 1:
5         return 1;
6     else:
7         return fibonacci(n-1) + fibonacci(n-2)
8
9 def print_fibonacci_series(n_terms):
10     print("Fibonacci series:")
11     for i in range(n_terms):
12         print(fibonacci(i))
13
14 n_terms = int(input())
15 print_fibonacci_series(n_terms)
```

	Input	Expected	Got	
✓	10	Fibonacci series: 0 1 1 2 3 5 8 13 21 34	Fibonacci series: 0 1 1 2 3 5 8 13 21 34	✓
✓	5	Fibonacci series: 0 1 1 2 3	Fibonacci series: 0 1 1 2 3	✓
✓	7	Fibonacci series: 0 1 1 2 3 5 8	Fibonacci series: 0 1 1 2 3 5 8	✓
✓	9	Fibonacci series: 0 1 1 2 3 5 8 13 21	Fibonacci series: 0 1 1 2 3 5 8 13 21	✓
✓	11	Fibonacci series: 0 1 1 2 3 5 8 13 21 34 55	Fibonacci series: 0 1 1 2 3 5 8 13 21 34 55	✓

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 merge sort without using recursive function on the given list of values.

For example:

Input	Result
7	left: [33]
33	Right: [42]
42	left: [9]
9	Right: [37]
37	left: [8]
8	Right: [47]
47	left: [5]
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	left: [10]
10	Right: [3]
3	left: [5]
5	Right: [61]
61	left: [74]
74	Right: [92]
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 %)

1 ||

	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] ***Run error*** Traceback (most recent call last): File "__tester__.python3", line 40, in <module> print(mergesort(arr)) File "__tester__.python3", line 32, in mergesort arr[start:end]=merge(left,right) File "__tester__.python3", line 4, in merge for i in range(0, len()+len(right)): TypeError: len() takes exactly one argument (0 given)	✖

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 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 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 l=[float(input()) for i in range(int(input()))]
11 s=quickSort(l)
12 print(s)
13

```


	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 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	Tuple: 6.2 found
6.2	

Answer: (penalty regime: 0 %)

```

1 def LS(list,n,key):
2     for i in range(0,n):
3         if(list[i]==key):
4             return i
5     return -1
6 list=[]
7 n=int(input())
8 for i in range(0,n):
9     temp=float(input())
10    list.append(temp)
11
12 tuple_data = tuple(list)
13 key=float(input())
14 res=LS(tuple_data,n,key)
15
16 if(res==-1):
17     print("Tuple: %s not found"%key)
18 else:
19     print("Tuple: %s found"%key)

```

	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	Tuple: 6.2 found	Tuple: 6.2 found	
	6.2			

	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.