

---

**Started on** Saturday, 14 September 2024, 3:29 PM

---

**State** Finished

---

**Completed on** Saturday, 14 September 2024, 4:08 PM

---

**Time taken** 40 mins 31 secs

---

**Grade** **80.00** out of 100.00

---

## Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement quick sort on the given float values and print the sorted list and pivot value of each iteration.

For example:

Input	Result
5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]
4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]

Answer: (penalty regime: 0 %)

```

1 def quick_sort(alist, start, end):
2     if end - start > 1:
3         p = partition(alist, start, end)
4         quick_sort(alist, start, p)
5         quick_sort(alist, p + 1, end)
6
7
8 def partition(alist, start, end):
9     pivot = alist[start]
10    i = start + 1
11    j = end - 1
12    print("pivot: ",pivot)
13    while True:
14        while (i <= j and alist[i] <= pivot):
15            i = i + 1
16        while (i <= j and alist[j] >= pivot):
17            j = j - 1
18
19        if i <= j:
20            alist[i], alist[j] = alist[j], alist[i]
21        else:
22            alist[start], alist[j] = alist[j], alist[start]

```

	Input	Expected	Got	
✓	5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	✓

	Input	Expected	Got	
✓	4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	✓
✓	6 3.1 4.2 5.1 2.3 7.4 5.9	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

## Question 2

Incorrect

Mark 0.00 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
2 def Merge_Sort(S):
3     if len(S) < 2:
4         return S
5
6     mid = len(S) // 2
7     y = Merge_Sort(S[:mid])
8     z = Merge_Sort(S[mid:])
9     result = []
10
11     i = 0
12     j = 0
13
14     while i < len(y) and j < len(z):
15         if y[i] > z[j]:
16             result.append(z[j])
17             j+=1
18         else:
19             result.append(y[i])
20             i+=1
21     result += y[i:]
22     result += z[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]	***Run error*** Traceback (most recent call last): File "__tester__.python3", line 29, in <module> S.append(int(input())) ValueError: invalid literal for int() with base 10: '10.2'	✗

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

	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 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 def search(List,n):
2     for i in range(len(List)):
3         if List[i]==n:
4             return True
5     return False
6 List=[]
7 x=int(input())
8 for i in range(x):
9     List.append(float(input()))
10 n=float(input())
11 res=search(List,n)
12 if search(List,n):
13     print(n,"Found")
14 else:
15     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	✓

	Test	Input	Expected	Got	
✓	search(List, n)	4 3.2 1.5 6.4 7.8 6.1	6.1 Not Found	6.1 Not Found	✓
✓	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 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 3 4 5 6 7 4	Found
search(List, n)	6 20 34 56 87 96 51 87	Found

Answer: (penalty regime: 0 %)

```

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

```

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

	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.