
Started on Monday, 26 May 2025, 10:55 AM

State Finished

Completed on Monday, 26 May 2025, 9:45 PM

Time taken 10 hours 49 mins

Overdue 8 hours 49 mins

Grade 80.00 out of 100.00

Question 1

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 search(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=input()
10    list.append(temp)
11 key=input()
12 res=search(list,n,key)
13 if(res!=-1):
14     print("Tuple:",key,"found")
15 else:
16     print("Tuple:",key,"not found")

```

	Input	Expected	Got	
✓	5 ram john akbar seetha oviya john	Tuple: john found	Tuple: john found	✓
✓	4 rohini fathima jenifer nizam rakesh	Tuple: rakesh not found	Tuple: rakesh not found	✓

	Input	Expected	Got	
✓	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 2

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	pivot: 9.7
6.3	pivot: 5.8
1.2	pivot: 4.6
4.6	[1.2, 4.6, 5.8, 6.3, 9.7]
5.8	
9.7	
6	pivot: 5.4
2.3	pivot: 3.6
7.8	pivot: 7.8
9.5	[2.3, 3.6, 4.2, 5.4, 7.8, 9.5]
4.2	
3.6	
5.4	

Answer: (penalty regime: 0 %)

```

1 def quick_sort(arr, low, high):
2     if low < high:
3         pi = partition(arr, low, high)
4         print(f"pivot: {arr[pi]}")
5         quick_sort(arr, low, pi - 1)
6         quick_sort(arr, pi + 1, high)
7 def partition(arr, low, high):
8     pivot = arr[high]
9     i = low - 1
10    for j in range(low, high):
11        if arr[j] <= pivot:
12            i += 1
13            arr[i], arr[j] = arr[j], arr[i]
14    arr[i + 1], arr[high] = arr[high], arr[i + 1]
15    return i + 1
16 n = int(input())
17 arr = []
18 for _ in range(n):
19     arr.append(float(input()))
20
21 quick_sort(arr, 0, len(arr) - 1)
22

```

	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 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 %)

```

1 def merge(l,r):
2     i=j=0
3     arr=[]
4     while i<len(l) and j<len(r):
5         if l[i]<r[j]:
6             arr.append(l[i])
7             i+=1
8         else:
9             arr.append(r[j])
10            j+=1
11    arr+=l[i:]+r[j:]
12    return arr
13 def Merge_Sort(arr):
14     size=1
15     n=len(arr)
16     while size<n:
17         size*=2
18         for pos in range(0,n,size):
19             start=pos
20             mid=pos+int(size//2)
21             end=pos+size
22             l=arr[start:mid]
```

	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 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 %)

```

1 def search(tpl, key):
2     for i in range(len(tpl)):
3         if tpl[i] == key:
4             return i
5     return -1
6 n = int(input())
7 lst = []
8 for _ in range(n):
9     val = float(input())
10    lst.append(val)
11    tpl = tuple(lst)
12    key = float(input())
13    result = search(tpl, key)
14    if result == -1:
15        print("Tuple: %.1f not found" % key)
16    else:
17        print("Tuple: %.1f found" % key)
18

```

	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.

Question **5**

Not answered

Mark 0.00 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 ||