Started on	Saturday, 26 April 2025, 3:35 PM
State	Finished
Completed on	Saturday, 26 April 2025, 4:12 PM
Time taken	37 mins 6 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 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]
         left=quickSort([x for x in arr[1:] if x<pivot[0]])</pre>
 5
 6
         right=quickSort([x for x in arr[1:] if x>=pivot[0]])
         print("left: ",left)
print("right: ",right)
 7
 8
 9
         return left+pivot+right
10
11 | l=[float(input()) for i in range(int(input()))]
12 s=quickSort(1)
13 print(s)
```

```
Input Expected
                                                Got
5
      left: []
                                                left: []
                                                                                          ~
      right: []
                                                right: []
6.9
8.3
      left: []
                                                left: []
2.1
      right: []
                                                right: []
      left: [1.5]
                                                left: [1.5]
1.5
      right: [6.4]
                                                right: [6.4]
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	left: []	left: []	~
	3.1	right: []	right: []	
	2.4	left: []	left: []	
	5.6	right: []	right: []	
	4.3	left: []	left: []	
	6.2	right: []	right: []	
	7.8	left: []	left: []	
		right: [7.8]	right: [7.8]	
		left: [4.3]	left: [4.3]	
		right: [6.2, 7.8]	right: [6.2, 7.8]	
		left: [2.4]	left: [2.4]	
		right: [4.3, 5.6, 6.2, 7.8]	right: [4.3, 5.6, 6.2, 7.8]	
		[2.4, 3.1, 4.3, 5.6, 6.2, 7.8]	[2.4, 3.1, 4.3, 5.6, 6.2, 7.8]	
~	8	left: []	left: []	~
	1.2	right: []	right: []	
	1.3	left: []	left: []	
	4.2	right: []	right: []	
	5.3	left: [6.8]	left: [6.8]	
	6.4	right: [9.2]	right: [9.2]	
	7.3	left: []	left: []	
	6.8	right: [6.8, 7.3, 9.2]	right: [6.8, 7.3, 9.2]	
	9.2	left: []	left: []	
		right: [6.4, 6.8, 7.3, 9.2]	right: [6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [5.3, 6.4, 6.8, 7.3, 9.2]	right: [5.3, 6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	right: [4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	
		left: []	left: []	
		right: [1.3, 4.2, 5.3, 6.4, 6.8, 7.3, 9.2]	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]	[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 **2**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:

TOT CAG	•
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		
- II		

Question **3**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 v def search(Tuple,x):
         for i in range(len(Tuple)):
 2 🔻
 3 ₹
             if Tuple[i]==x:
 4
                 return 1
        else:
 6
             return 0
 7
 8
 9
   n=int(input())
arr=[]
10
11
12 v for i in range(n):
13
        ele=(input())
        arr.append(ele)
14
15
   Tuple=tuple(arr)
x=input()
ns=search(Tuple,x)
if ans==1:
        print("Tuple:",x,"found")
19
20 v else:
        print("Tuple:",x,"not found")
21
22
```

	Input	Expected	Got	
~	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	~
~	6 rose jasmine tulips marigold hibiscus lotus lilly	Tuple: lilly not found	Tuple: lilly not found	*

Passed all tests! ✓

Question **4**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 %)

```
def Factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n * Factorial(n-1)
    n=int(input())
    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 **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	Tuple:	6.4	found
3.2			
1.5			
6.4			
7.8			
9.5			
6.4			
6	Tuple:	6.2	found
3.2			
1.2			
3.4			
5.3			
6.2			
6.8			
6.2			

Answer: (penalty regime: 0 %)

```
1 v def search(Tuple,x):
        for i in range(len(Tuple)):
2 🔻
             if Tuple[i]==x:
 3 ▼
 4
                 return 1
 5 🔻
        else:
 6
             return 0
 7
 8
 9
10
   n=int(input())
11
   arr=[]
12 v for i in range(n):
        ele=float(input())
13
        arr.append(ele)
14
   Tuple=tuple(arr)
15
   x=float(input())
ans=search(Tuple,x)
16
17
18 v if ans==1:
19
        print("Tuple:",x,"found")
20 v else:
        print("Tuple:",x,"not found")
21
22
```

	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	~
~	4 2.1 3.2 6.5 4.5 3.5	Tuple: 3.5 not found	Tuple: 3.5 not found	~

Correct

Marks for this submission: 20.00/20.00.