TT DS PYTHON MODULE-19

Started on	Thursday, 20 March 2025, 2:54 PM
State	Finished
Completed on	Thursday, 20 March 2025, 3:24 PM
Time taken	30 mins 9 secs
Grade	**100.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Flag question

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 %)

```
def Merge_Sort(arr):
    width = 1
    n = len(arr)

    while width < n:
        for i in range(0, n, 2 * width):
            left = arr[i:i + width]
            right = arr[i + width:i + 2 * width]
            arr[i:i + 2 * width] = merge(left, right)
        width *= 2

def merge(left, right):
    result = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            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]
	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

Write a python program to implement quick sort on the given float array values.

For example:

of 20.00

Flag question

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 %)

```
def qsort(L):
    if L==[]:
        return []
    pivot = L[0:1]
    left=qsort([x for x in L[1:] if x<L[0]])
    right=qsort([x for x in L[1:] if x>=L[0]])
    print("left: ",left)
    print("right: ",right)
    return left+pivot+right
list=[]
n=int(input())
for i in range(n):
    list.append(float(input()))
print(qsort(list))
```

	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: [] 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: [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

Flag question

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(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

Flag question

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	Tuple: rakesh not found

jenifer nizam rakesh

Answer: (penalty regime: 0 %)

```
def linear_search(tup, target):
    for item in tup:
        if item == target:
            return True
    return False

n = int(input())
lst = []
for _ in range(n):
    lst.append(input())
tup = tuple(lst)
target = input()
if linear_search(tup, target):
    print(f"Tuple: {target} found")
else:
    print(f"Tuple: {target} 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	
	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

Flag question

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 %)

```
def search(List, n):
```

```
    for item in List:
        if item == n:
            return True
    return False
num_elements = int(input())
List = []

for _ in range(num_elements):
    List.append(int(input()))

n = int(input())

if search(List, n):
    print("Found")
else:
    print("Not Found")
```

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

Finish r