Started on	Monday, 9 September 2024, 8:46 AM
State	Finished
Completed on	Monday, 9 September 2024, 9:11 AM
Time taken	24 mins 22 secs
Grade	**80.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 20.00

Write a python program to implement binary search on the given list of float values using iterative method

For example:

Test	Input	Result
`binarySearchAppr(arr, 0, len(arr)-1, x)`	5 3.2 6.1 4.5 9.6 8.3 6.1	`Element is present at index 2`
`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 3.1 2.3 5.1 4.6 3.2 9.5 4.6	`Element is present at index 3`

Answer: (penalty regime: 0 %)

```
def binary_search(arr,str,en,x):
    mid=(str+en)//2
    if st <= en:
        if arr[mid]==x:
            print("Element is present at index",mid)
        elif x<arr[mid]:
            binary_search(arr,st,mid-1,x)
        elif x>arr[mid]:
            binary_search(arr,mid+1,en,x)
    else:
        print("Element is not present in array")

n=int(input())
arr=[]
for i in range(n):
    arr+=(int(input())
x=int(input())
```

Syntax Error(s)

```
  File "__tester__.python3", line 17
    x=int(input())
    ^
SyntaxError: invalid syntax
```

Incorrect

Marks for this submission: 0.00/20.00.

Question **2**

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 %)

```
def linear_search(list1,n,key):
    for i in range(0,n):
        if(list1[i]==key):
            return i
    return -1
list1=[]
n=int(input())
for i in range(n):
    ele=input()
    list1.append(ele)
key=input()
def convert(list1):
    return tuple(list1)
res=linear_search(list1,n,key)
if(res==-1):
    print("Tuple: {} not found".format(key))
else:
    print("Tuple: {} found".format(key))
```

	Input	Expected	Got	
✔	5 ram john akbar seetha oviya john	Tuple: john found	Tuple: john found	✔

	Input	Expected	Got	
✔	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 **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 %)

```
def Merge_Sort(S):
    n = len(S)
    current_size = 1

    while current_size < n:
        left = 0
        while left < n - 1:
            mid = min(left + current_size - 1, n - 1)
            right = min(left + 2 * current_size - 1, n - 1)

            merge(S, left, mid, right)
            left += 2 * current_size

        current_size *= 2


def merge(S, left, mid, right):
    n1 = mid - left + 1
    n2 = right - mid

    L = [S[left + i] for i in range(n1)]
    R = [S[mid + 1 + i] for i in range(n2)]
```

	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]	✔

	Test	Input	Expected	Got	
✔	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 **4**

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 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]
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]

Answer: (penalty regime: 0 %)

```
def partition(arr, low, high):
    i = low - 1
    pivot = arr[high]

    for j in range(low, high):
        if arr[j] <= pivot:
            i += 1
            arr[i], arr[j] = arr[j], arr[i]
    arr[i + 1], arr[high] = arr[high], arr[i + 1]
    return i + 1

def quick_sort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        print("pivot: ", arr[pi])
        quick_sort(arr, low, pi - 1)
        quick_sort(arr, pi + 1, high)

# Take input from the user
def take_input():
    n = int(input())
    arr = []
```

	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]	✔

	Input	Expected	Got	
✔	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]	✔
✔	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 **5**

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 %)

```
def rec(n):
    if n<=1:
        return n
    else:
        return(rec(n-1)+rec(n-2))

nt=int(input())
if nt<=0:
    print("Invalid input ! Please input a positive value")
else:
    print("Fibonacci series:")
    for i in range(nt):
        print(rec(i))
```

	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.