Started on	Thursday, 20 March 2025, 1:31 PM
State	Finished
Completed on	Saturday, 29 March 2025, 9:41 AM
Time taken	8 days 20 hours
Overdue	8 days 18 hours
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a Python Program Using a recursive function to calculate the sum of a sequence

For example:

Input	Result
20	210
36	666
45	1035

Answer: (penalty regime: 0 %)

```
def sumof(n,a,result):
    if (n<a):
        print (result)
    else:
        result+=a
        sumof(n,a+1,result)
n=int(input())
sumof(n,1,0)
```

	Input	Expected	Got	
✓	20	210	210	✓
✓	36	666	666	✓
✓	45	1035	1035	✓
✓	58	1711	1711	✓
✓	65	2145	2145	✓

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 without using recursive function on the given list of values.

For example:

Input	Result
7 33 42 9 37 8 47 5	left: [33] Right: [42] left: [9] Right: [37] left: [8] Right: [47] left: [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 10 3 5 61 74 92	left: [10] Right: [3] left: [5] Right: [61] left: [74] Right: [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 %)

```
print(5)
```

	Input	Expected	Got	
✖	7 33 42 9 37 8 47 5	left: [33] Right: [42] left: [9] Right: [37] left: [8] Right: [47] left: [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]	5	✖

Some hidden test cases failed, too.
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 implement binary search on the given list of string values using iterative method

For example:

Test	Input	Result
`binarySearchAppr(arr, 0, len(arr)-1, x)`	5 one two three four five two	Element is present at index 4
`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 one three five seven nine eleven thirteen	Element is not present in array

Answer: (penalty regime: 0 %)

```
def binarySearchAppr(arr, low,high, x):
    while(low<=high):
        mid=(low+high)//2
        if arr[mid]==x:
            return mid
        elif x<arr[mid]:
            high=mid-1
        else:
            low=mid+1
    return -1
n=int(input());
arr=[];
for i in range(n):
    arr.append(input())
x=input()
arr.sort();
result=binarySearchAppr(arr,0,len(arr)-1,x)
if result>=0:
    print("Element is present at index",result);
else:
    print("Element is not present in array");
```

	Test	Input	Expected	Got	
✓	`binarySearchAppr(arr, 0, len(arr)-1, x)`	5 one two three four five two	Element is present at index 4	Element is present at index 4	✓
✓	`binarySearchAppr(arr, 0, len(arr)-1, x)`	6 one three five seven nine eleven thirteen	Element is not present in array	Element is not present in array	✓

	Test	Input	Expected	Got	
✓	`binarySearchAppr(arr, 0, len(arr)-1, x)`	4 two four six eight six	Element is present at index 2	Element is present at index 2	✓

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 pivot_place(list1,first,last):
    pivot=list1[last]
    left=first
    right=last-1
    #print("pivot: ",pivot)
    while True:
        while left<=right and list1[left]<=pivot:
            left+=1
        while left<=right and list1[right]>=pivot:
            right-=1
        if left>right:
            break
        else:
            list1[left],list1[right]=list1[right],list1[left]
    list1[last],list1[left]=list1[left],list1[last]
    return left

def quicksort(list1,first,last):
    if first<last:
        p=pivot_place(list1,first,last)
        quicksort(list1,first,p-1)
        quicksort(list1,p+1,last)
```

	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 **5**

Correct

Mark 20.00 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 %)

```python
def binarySearchAppr(arr, low,high, x):
    while(low<=high):
        mid=(low+high)//2
        if arr[mid]==x:
            return mid
        elif x<arr[mid]:
            high=mid-1
        else:
            low=mid+1
    return -1
n=int(input());
arr=[];
for i in range(n):
    arr.append(input())
x=input()
arr.sort();
result=binarySearchAppr(arr,0,len(arr)-1,x)
if result>=0:
    print("Element is present at index",result);
else:
    print("Element is not present in array");
```

	Test	Input	Expected	Got	
✓	`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	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	Element is present at index 3	✓

	Test	Input	Expected	Got	
✓	`binarySearchAppr(arr, 0, len(arr)-1, x)`	8 2.1 6.3 5.2 4.2 9.3 6.7 5.6 9.8 7.2	Element is not present in array	Element is not present in array	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.