

Started on	Friday, 25 April 2025, 11:40 AM
State	Finished
Completed on	Friday, 25 April 2025, 11:59 AM
Time taken	18 mins 58 secs
Grade	80.00 out of 100.00

Question 1

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_Sort(S):
2     size=len(S)
3     if(size>1):
4         mid=size//2
5         l=S[:mid]
6         r=S[mid:]
7         Merge_Sort(l)
8         Merge_Sort(r)
9         i=j=k=0
10        ls=len(l)
11        rs=len(r)
12        while(i<ls and j<rs):
13            if(l[i]<r[j]):
14                S[k]=l[i]
15                i+=1
16            else:
17                S[k]=r[j]
18                j+=1
19                k+=1
20        while(i<ls):
21            S[k]=l[i]
22            i+=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]	✓

	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 2

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 %)

```
1 def fibo(n):  
2     f=0  
3     s=1  
4     print("Fibonacci series:")  
5     for i in range(n):  
6         print(f)  
7         t=f+s  
8         f=s  
9         s=t  
10 n=int(input())  
11 fibo(n)
```

	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.

Question 3

Not answered

Mark 0.00 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 %)

1	
---	--

Question 4

Correct

Mark 20.00 out of 20.00

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	Found
	3	
	4	
	5	
	6	
	7	
	4	
search(List, n)	6	Found
	20	
	34	
	56	
	87	
	96	
	51	
	87	

Answer: (penalty regime: 0 %)

```

1 global key
2 def search(List,n):
3     for i in range(n):
4         if List[i]==key:
5             return i
6     else:
7         return -1
8 List=[]
9 n=int(input())
10 for i in range(n):
11     List.append(int(input()))
12 key=int(input())
13 res=search(List,n)
14 if(res!=-1):
15     print("Found")
16 else:
17     print("Not Found")

```

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

	Test	Input	Expected	Got	
✓	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.

Question 5

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 %)

```

1 def binarySearchAppr(arr,start,end,x):
2     if(start<=end):
3         mid=(start+end)//2
4         if(x==arr[mid]):
5             return mid
6         elif(x<arr[mid]):
7             return binarySearchAppr(arr,start,mid-1,x)
8         elif(x>arr[mid]):
9             return binarySearchAppr(arr,mid+1,end,x)
10    else:
11        return -1
12    arr=[]
13    n=int(input())
14    for i in range(n):
15        arr.append(input())
16    arr=sorted(arr)
17    x=input()
18    result=binarySearchAppr(arr,0,len(arr)-1,x)
19    if(result!=-1):
20        print(f"Element is present at index {result}")
21    else:
22        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.