
Started on Tuesday, 15 April 2025, 1:46 PM

State Finished

Completed on Tuesday, 15 April 2025, 3:00 PM

Time taken 1 hour 13 mins

Grade **80.00** out of 100.00

Question 1

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 x=input()
17 arr=sorted(arr)
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	✓

	Test	Input	Expected	Got	
✓	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	✓
✓	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 **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)  
12  
13
```

	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 quick sort on the given float array values.

For example:

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 %)

1 ||

Question 4

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 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found
6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	Tuple: 6.2 found

Answer: (penalty regime: 0 %)

```

1 def search(List,key,n):
2     for i in range(n):
3         if(List[i]==key):
4             return i
5     else:
6         return -1
7 List=[]
8 n=int(input())
9 for i in range(n):
10     List.append(float(input()))
11 key=float(input())
12 result=search(List,key,n)
13 if(result!=-1):
14     print(f"Tuple: {key} found")
15 else:
16     print(f"Tuple: {key} not found")

```

	Input	Expected	Got	
✓	5 3.2 1.5 6.4 7.8 9.5 6.4	Tuple: 6.4 found	Tuple: 6.4 found	✓

	Input	Expected	Got	
✓	6 3.2 1.2 3.4 5.3 6.2 6.8 6.2	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	✓

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 merge sort using iterative approach on the given list of float values.

For example:

Test	Input	Result
Merge_Sort(S)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]
Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]

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)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]	✓

	Test	Input	Expected	Got	
✓	Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]	✓
✓	Merge_Sort(S)	4 2.3 6.1 4.5 96.5	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.