

Started on Thursday, 19 September 2024, 3:16 PM**State** Finished**Completed on** Thursday, 19 September 2024, 3:52 PM**Time taken** 35 mins 40 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 quick sort on the given float values and print the sorted list and pivot value of each iteration.

For example:

Input	Result
5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]
4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]

Answer: (penalty regime: 0 %)

```

7   pivot = alist[start]
8   i = start + 1
9   j = end - 1
10  print("pivot: ",pivot)
11  while True:
12      while (i <= j and alist[i] <= pivot):
13          i = i + 1
14      while (i <= j and alist[j] >= pivot):
15          j = j - 1
16
17      if i <= j:
18          alist[i], alist[j] = alist[j], alist[i]
19      else:
20          alist[start], alist[j] = alist[j], alist[start]
21      return j
22  alist = []
23  n=int(input())
24  for i in range(n):
25      alist.append(float(input()))
26  print('Input List\n', alist)
27  quick_sort(alist, 0, len(alist))
28  print('Sorted List\n', alist)

```

	Input	Expected	Got	
	5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	
	4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	
	6 3.1 4.2 5.1 2.3 7.4 5.9	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question 2

Correct

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.

Mark 20.00 out of 20.00

Flag question

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 %)

```

1 def search(List, x):
2     for i in range(len(List)):
3         if List[i] == x:
4             return True
5     return False
6 a=[]
7 n = int(input())
8 for i in range(n):
9     a.append(input())
10 x=input()
11 List=list(a)
12 if search(List, x):
13     print("Found")
14 else:
15     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.

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	Output
-------	--------

input	Result
5	Factorial of number 5 = 120
6	Factorial of number 6 = 720

Answer: (penalty regime: 0 %)

```

1 n=int(input())
2 fact=1
3 for i in range(1,n+1):
4     fact=fact*i
5 print("Factorial of number",n,"=",fact)
6

```

	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 jenifer nizam rakesh	Tuple: rakesh not found

Answer: (penalty regime: 0 %)

```

1 def linear_search(iterable,element):
2     is_found=False
3     for i in range(len(iterable)):
4         if iterable[i]==element:
5             is_found=True
6             return f"{element} found"
7     if not is_found:
8         return f"{element} not found"
9 list1=[]
10 n=int(input())
11 for i in range(n):
12     list1.append(input())
13 numbers_tuple=tuple(list1)
14 x=input()
15 print("Tuple:".linear_search(numbers_tuple,x))

```

	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 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     if len(S) < 2:
3         return S
4
5     mid = len(S)//2
6     y = Merge_Sort(S[:mid])
7     z = Merge_Sort(S[mid:])
8     result = []
9
10    i = 0
11    j = 0
12
13    while i<len(y) and j<len(z):
14        if y[i]>z[j]:
15            result.append(z[j])
16            j+=1
17        else:
18            result.append(y[i])
19            i+=1
20    result += y[i:]
21    result += z[j:]
22    return result

```

	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.

[Finish](#)