Unit 4 - Lesson 5 - Built-in List Functions

About this unit

Built-in List Functions

Built-in List Functions

Unit • 100% completed

Built-in List Functions - Python

Assessment

1.1. Understanding List functions

12:57

ere is a list of built-in functions that can be applied on a list:

all(): Returns True if all the items in the list have a true value.

```
print(all([' ', ',', '1', '2']))
True
```

. **any()**: Returns True if atleast one item in the list has a true value.

```
print(any([' ', ',', '1', '2']))
True
```

3. **enumerate()**: Returns an enumerate object consisting of the index and the values of all items of the ist as a tuple pair.

```
print(list(enumerate(['a','b','c','d','e'])))
[(0, 'a'), (1, 'b'), (2, 'c'), (3, 'd'), (4, 'e')]
```

4. **len()**: It calculates the length of the list i.e., the number of elements in the list.

```
print(len(['a', 'b', 'c', 'd', 'e']))
5
```

5. **max()**: Returns the item with the highest value from the list

```
print(max([1, 2, 3, 4, 5]))
5
```

Sample Test Cases

Explorer

Listfuncs...

```
1   data = input("data: ")
2   list1 = data.split(",")
3   #find the length
4   size = len(list1)
5   #convert the elements into integer type
6   for i in range(size):
7       list1[i] = int(list1[i])
8   print("length:",size)#print the length of list1
9   print("list enumerate:", list(enumerate(list1)))#print t
    enumerate of list1
10  print("max:",max(list1))#print the maximum value of list
11  print("min:",min(list1))#print the minimum value of list
12  print("list:",sorted(list1))#print the sorted list
13
```

Average time: 0.007 s (7.25 ms)

Maximum time: 0.010 s (10.00 ms)

2 out of 2 shown test case(s) passed

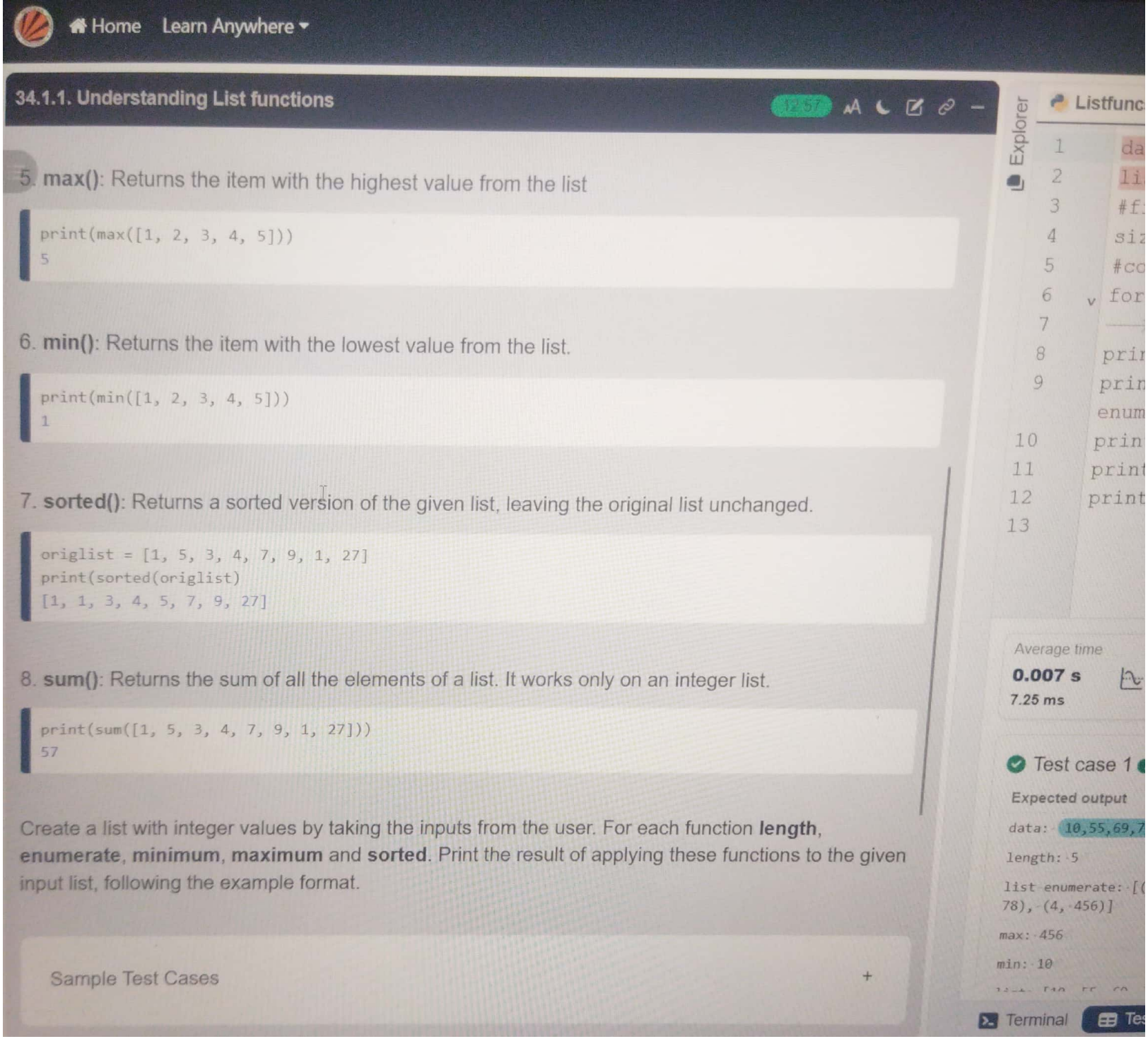
2 out of 2 hidden test case(s) passed

Test case 1 8 ms

Debug

Expected output	Actual output
data: 10,55,69,78,456	data: 10,55,69,78,456
length: 5	length: 5
list enumerate: [(0, 10), (1, 55), (2, 69), (3, 78), (4, 456)]	list enumerate: [(0, 10), (1, 55), (2, 69), (3, 78), (4, 456)]
max: 456	max: 456
min: 10	min: 10

Terminal | Test cases


Home
Learn Anywhere
34.1.1. Understanding List functions
5. max(): Returns the item with the highest value from the list
print(max([1, 2, 3, 4, 5]))
5
6. min(): Returns the item with the lowest value from the list.
print(min([1, 2, 3, 4, 5]))
1
7. sorted(): Returns a sorted version of the given list, leaving the original list unchanged.
origlist = [1, 5, 3, 4, 7, 9, 1, 27]
print(sorted(origlist)
[1, 1, 3, 4, 5, 7, 9, 27]
8. sum(): Returns the sum of all the elements of a list. It works only on an integer list.
print(sum([1, 5, 3, 4, 7, 9, 1, 27]))
57
Create a list with integer values by taking the inputs from the user. For each function length, enumerate, minimum, maximum and sorted. Print the result of applying these functions to the given input list, following the example format.
Sample Test Cases
Explorer
Listfunc
Average time
0.007 s
7.25 ms
Test case 1
Expected output
data: 10,55,69,7
length: 5
list enumerate: [(
78), (4, 456)]
max: 456
min: 10
Terminal

Write the code

Your task is to:

- Take comma (',') separated list of integers as input from the user.
- In the first line of output print the sum of all integers values in the list.
- In the second line of output print the cube value of the sum computed above.
- In the last line of output print the sum of 4^{th} power of the integer values in the list .

Note:

- **Refer to the Displayed test cases for a better understanding.**
- **The inputs numbers can be randomly distributed, and can be +ve or -ve.**

Constraints:

- 1 <= number of elements <= 100
- **Make use of built-in functions on lists to make your solution efficient.**

Sample Test case:

list: 1,3,2,5,4 -----> Input elements of the list separated by comma(,).

15 -----> Sum of all values.

3375 ----> cube value of the sum computed.

979------> Sum of 4^{th} power of values is $1^4 + 3^4 + 2^4 + 5^4 + 4^4 = 1 + 81 + 16 + 625 + 256 = 979$

Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when the user's input and output match the expected input and output.

Sample Test Cases

Test Case 1:

listoperations.py

```
1  #write your code here
```

Write the code

Your task is to:

- Take comma (',') separated list of integers as input from the user, every input represents total count of project assignments completed by your team mates in ongoing year.
- During verification of the list if you found that in terms of progress 0 assignments are completed by everyone in your team, in this case print **"Invalid"**. Otherwise print the count of assignments completed by your team members those are odd ordered in the list.

Note:

- **Refer to the Displayed test cases for a better understanding.**
- **The input assignment count is a positive integer >= 0.**

Constraints:

- 1 <= number of elements <= 100
- **Make use of built-in functions on lists to make your solution efficient.**

Sample Test case 1:
list: 11,13,12,14,15 -----> Input elements of the list separated by comma(,).
13 14 -----> As all assignment counts are ok as per constraint we, print odd ordered i.e. at index 1, 3, and 5.

Sample Test case 2:
11,13,12,0,14
13 0 -----> As all assignment counts are ok as per constraint, we print odd ordered i.e. at index 1, 3, and 5

Sample test case 3:
0,0,0,0
Invalid -----> All your team members are good for nothing, not tolerated.

Instructions:

listoperations.py

```
1 #write your code here
2 list1 = list(map(int,input("list: ").split(",")))
3
```

Execution Results

0 out of 3 shown cases successful

0 out of 5 hidden cases successful

Show on

✖ **Test Case - 1** (Execution Time: 10 ms)

Expected Output	User Output
list: 11,13,12,14,15	list: 11,13,12,14,15
13 14	Empty

⚠ : indicates the mismatch in the expected output

✖ **Test Case - 2** (Execution Time: 6 ms)

Expected Output	User Output
list: 11,13,12,0,14	list: 11,13,12,0,14
13 0	Empty

⚠ : indicates the mismatch in the expected output.

Finish | Clear | Submit | Prev | Next

Write the code

Your task is to:

- Take comma (',') separated list of integers as input from the user.
- In the first line on output print the square value of max element out of the list.
- In the second line of output print the square value of min element out of the list.
- In the last line of output print the absolute difference of above computed values as (min value)^3 - (max value)^3 .

Note:

- **Refer to the Displayed test cases for a better understanding.**
- **The inputs numbers can be randomly distributed, and can be +ve or -ve.**

Constraints:

- 1 <= number of elements <= 100
- **Make use of built-in functions on lists to make your solution efficient.**

Sample Test case:

list: 1,3,2,5,4 -----> Input elements of the list separated by comma(,).
25 -----> Max value square.
1----> Min value square.
124------> Absolute difference as mentioned in the problem statement.

Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when the user's input and output match the expected input and output.

Sample Test Cases

listoperations.py

```
1  #write your code here
```

Execution Results

0 out of 3 shown cases passed

0 out of 4 hidden cases passed

✖ Test Case - 1 (Execution Time: 3 ms)

Expected Output	User Outp
list: 1,3,2,5,4	1,3,2,5,
25	Empty
1	Empty
124	Empty

⚠ indicates the mismatch in the expected output.

✖ Test Case - 2 (Execution Time: 4 ms)

Expected Output	User Output
list: -1,-3,-2,-4,-5	-1,-3,-2,-
1	Empty

Finish Clear Submit Prev Next

Write the code

Your task is to:

- Take comma (',') separated list of integers as input from the user, every input represents the encrypted code's.
- During verification of the list if you found **atleast one** encryption as **0**, then print **"Invalid"**. Otherwise print the even ordered encrypted code values

Note:

- **Refer to the Displayed test cases for a better understanding.**
- **The input encrypted code value's are positive integers >= 0.**

Constraints:

- 1 <= number of elements <= 100
- **Make use of built-in functions on lists to make your solution efficient.**

Sample Test case 1:
list: 11,13,12,15,14 -----> Input elements of the list separated by comma(,).
11 12 14 -----> As all encrypted code's are ok as per constraint we print even ordered code's i.e. at index 0, 2, and 4.

Sample Test case 2:
11,13,12,0,14
Invalid -----> Because we found out atleast one invalid encrypted code i.e. 0.

Instructions:

- Your input and output must follow the input and output layout mentioned in the visible sample test case.
- Hidden test cases will only pass when the user's input and output match the expected input and output.

listoperations.py

```
1  #write your code here
```

Execution Results

0 out of 3 shown cases successful

0 out of 6 hidden cases successful

✖ Test Case - 1 (Execution Time: 2 ms)

Expected Output	User Output
list: 1,3,2,0,4 Invalid	1 1,3,2,0,4 Empty

⚠ indicates the mismatch in the expected output.

✖ Test Case - 2 (Execution Time: 4 ms)

Expected Output	User Output
list: 1,3,2,5,4 1 2 4	1 1,3,2,5,4 Empty

⚠ indicates the mismatch in the expected output.

Finish Clear Submit Prev Next