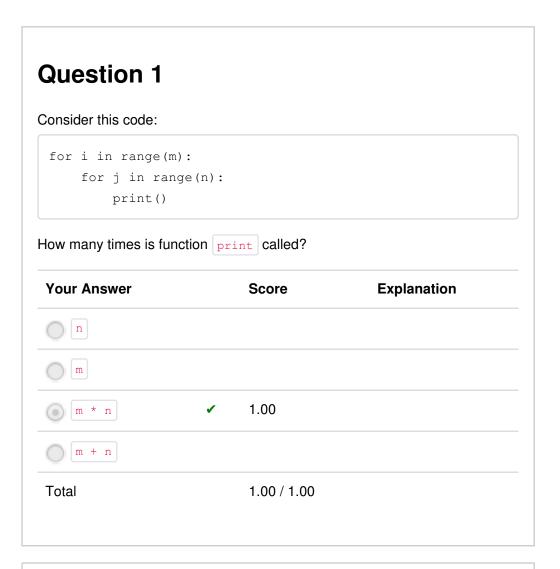
Feedback — Week 3 Exercise

You submitted this homework on **Sun 14 Apr 2013 8:31 AM PDT -0700**. You got a score of **17.00** out of **18.00**. You can attempt again, if you'd like.

Some of these questions refer to the sorting functions from the videos. We have posted the summaries and code on the Video Lectures page.



Question 2

Consider this code:

```
for i in range(m):
    print()

for j in range(n):
    print()

How many times is function print called?

Your Answer Score Explanation

m * n

m * n

Total 1.00 / 1.00
```

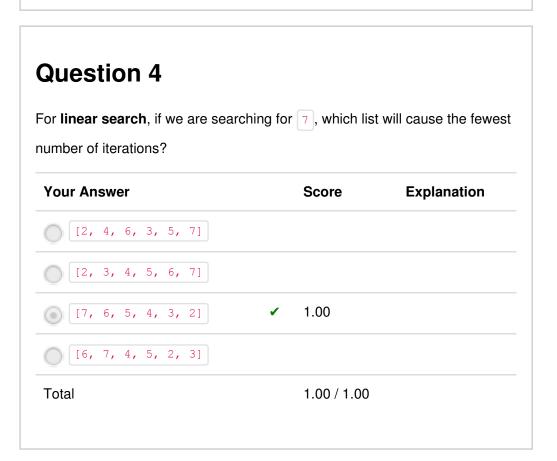
Assume variable \square refers to a list of items.

You have a problem you are trying to solve and you figured out two different approaches that would work.

```
# Approach 1:
for i in range(len(L)):
    for j in range(len(L)):
        # do a few assignment statements to accomplish th
e task.
```

```
# Approach 2:
for i in range(1000):
    for j in range(len(L)):
        # do a few assignment statements to accomplish th
e task.
```

Your Answer	5	Score	Explanation
When L has exactly 1000 items.			
When L contains strings.			
When L has more than 1000 items.	✓ 1	1.00	
When L is sorted.			
Total	1	1.00 / 1.00	



The list [4, 2, 5, 6, 7, 3, 1] is shown below after each pass of a sorting algorithm:

[1, 2, 5, 6, 7, 3, 4]
[1, 2, 5, 6, 7, 3, 4]
[1, 2, 3, 6, 7, 5, 4]
[1, 2, 3, 4, 7, 5, 6]
[1, 2, 3, 4, 5, 7, 6]
[1, 2, 3, 4, 5, 6, 7]
[1, 2, 3, 4, 5, 6, 7]

Which sorting algorithm is being executed?

Your Answer		Score	Explanation
bubble sort			
insertion sort			
selection sort	1	1.00	
Total		1.00 / 1.00	

Question 6

The list [4, 2, 5, 6, 7, 3, 1] is shown below after each pass of a sorting algorithm:

Which sorting algorithm is being executed?

Your Answer		Score	Explanation
bubble sort	✓	1.00	



The list [4, 2, 5, 6, 7, 3, 1] is shown below after each pass of a sorting algorithm:

```
[4, 2, 5, 6, 7, 3, 1]
[2, 4, 5, 6, 7, 3, 1]
[2, 4, 5, 6, 7, 3, 1]
[2, 4, 5, 6, 7, 3, 1]
[2, 4, 5, 6, 7, 3, 1]
[2, 3, 4, 5, 6, 7, 1]
[1, 2, 3, 4, 5, 6, 7]
```

Which sorting algorithm is being executed?

bubble sortinsertion sort✓ 1.00selection sort	Score Explanation
selection sort	✓ 1.00
Total 1.00 / 1.00	1.00 / 1.00

Question 8

List [1, 5, 8, 7, 6, 1, 7] is being sorted using **selection sort**. Here is what the list will look like after each of the first three passes:

• After the 1st pass: [1, 5, 8, 7, 6, 1, 7]

• After the 2nd pass: [1, 1, 8, 7, 6, 5, 7]

• After the 3rd pass: [1, 1, 5, 7, 6, 8, 7]

What will the list look like after the 4th pass?

Your Answer		Score	Explanation
[1, 1, 5, 6, 7, 8, 7]	✓	1.00	
[1, 1, 5, 7, 6, 8, 7]			
[1, 1, 5, 6, 7, 7, 8]			
Total		1.00 / 1.00	

Question 9

List [6, 8, 2, 1, 1, 9, 4] is being sorted using **insertion sort**. Here is what the list will look like after each of the first three passes:

• After the 1st pass: [6, 8, 2, 1, 1, 9, 4]

• After the 2nd pass: [6, 8, 2, 1, 1, 9, 4]

• After the 3rd pass: [2, 6, 8, 1, 1, 9, 4]

What will the list look like after the 4th pass?

Your Answer	Score	Explanation
[1, 2, 6, 8, 1, 9, 4]	✓ 1.00	
[1, 1, 2, 6, 8, 9, 4]		
[1, 6, 8, 2, 1, 9, 4]		
Total	1.00 / 1.00	

In **bubble sort**, on the first pass through the list, which item gets moved to the far right?

Your Answer		Score	Explanation
The largest item.	•	1.00	
The item that was originally at the second-last index.			
The item that was originally at index 0.			
The smallest item.			
An odd number.			
Total		1.00 / 1.00	
Total			

Question 11

Here is the code for function insert with docstring and comments removed:

```
def insert(L, i):
    value = L[i]

j = i
while j != 0 and L[j - 1] > value:
    L[j] = L[j - 1]
    j = j - 1

L[j] = value
```

In the following list, there is an x at index x. In this question, you will choose a value for that variable.

```
L = [2, 5, 6, 7, 8, x, 4]
```

The first 5 items are sorted.

If we call <u>insert(L, 5)</u>, that unknown value will be inserted into the sorted section, growing the sorted section by 1 item. Select a value for x that would be moved all the way to index o in the list.

Your Answer		Score	Explanation
1	✓	1.00	
3			
9			
4			
Total		1.00 / 1.00	

Question 12

Here is the code for function insert with docstring and comments removed:

```
def insert(L, i):
    value = L[i]

j = i
while j != 0 and L[j - 1] > value:
    L[j] = L[j - 1]
    j = j - 1

L[j] = value
```

In the following list, there is an x at index x at i

```
L = [2, 5, 6, 7, 8, x, 4]
```

The first 5 items are sorted.

If we call <code>insert(L, 5)</code>, that unknown value will be inserted into the sorted section, growing the sorted section by 1 item. Select the value for <code>x</code> that would not move.

Your Answer		Score	Explanation
4			
0			
9	✓	1.00	
3			
Total		1.00 / 1.00	

Question 13

Here is the code for function insert with docstring and comments removed:

```
def insert(L, i):
    value = L[i]

j = i
while j != 0 and L[j - 1] > value:
    L[j] = L[j - 1]
    j = j - 1

L[j] = value
```

In general, function call <code>insert(L, i)</code> might move the item at index <code>i</code> all the way to index <code>0</code> in the list (if that item is smaller than everything

in the sorted section); it might not move it at all (if that item is larger than everything in the sorted section); or it might be moved partway (if that item is neither smaller nor larger than everything in the sorted section).

The while loop can be terminated for one of two reasons: j == 0 or $L[j - 1] \leftarrow value$. In which situation does the loop terminate because j == 0?

	Score	Explanation
•	1.00	
	1.00 / 1.00	
	•	1.00 /

Question 14

Here is the code for function insert:

```
def insert(L, i):
    value = L[i]

j = i
while j != 0 and L[j - 1] > value:
    L[j] = L[j - 1]
    j = j - 1

L[j] = value
```

For function call [insert(L, i)], in the worst case, the item at index [i]

is moved all the way to index o. Variable j starts off at i and is decreased by 1 on each iteration of the while loop until it reaches o. In this worst-case situation, how many times is the body of the while loop executed?

Your Answer		Score	Explanation
<u>i - 1</u>			
i + 1			
i	✓	1.00	
2 * i			
Total		1.00 / 1.00	

Question 15

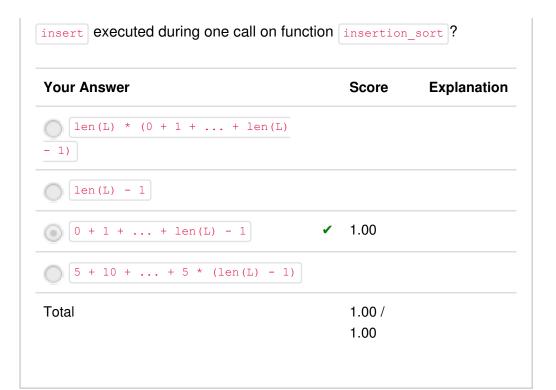
Here is the code for function <code>insertion_sort</code>:

```
def insertion_sort(L):
   for i in range(len(L)):
      insert(L, i)
```

This question is about the *worst-case* running time for this code. (The worst case for insertion sort happens when a list is sorted in reverse, from largest to smallest.)

- On the first iteration of this loop, i refers to 0, so insert (L, 0) is called, and the while loop in function insert iterates 0 times.
- On the second iteration, insert(L, 1) is called, and the while loop in function insert iterates 1 time.
- On the last iteration, insert (L, len(L) 1) is called, and the while loop in function insert iterates len(L) 1 times.

In total, how many times is the body of the while loop in function



In the worst case, on a call on function <code>insertion_sort(L)</code>, the total number of times the loop body in function <code>insert</code> is executed is this:

```
0 + 1 + 2 + 3 + \dots + (len(L) - 3) + (len(L) - 2) + (len(L) - 1)
```

The 0 doesn't affect the sum, so we can simplify to this:

```
1 + 2 + 3 + ... + (len(L) - 3) + (len(L) - 2) + (len(L) - 1)
```

We can add the first and last items together, and the second and second-last items together, and so on:

```
1 + (len(L) - 1)  # The 1 and the -1 cancel, leaving
len(L)
+ 2 + (len(L) - 2)  # The 2 and the -2 cancel, leaving
len(L)
+ 3 + (len(L) - 3)  # The 3 and the -3 cancel, leaving
len(L)
```

Every line in the equation adds up to len(L). Roughly how many lines in the equation are there, and what is the total number of times the loop body is executed? (Hint: work this out using a smaller example, such as a length [9] list, and then generalize.) **Your Answer** Score **Explanation** 1.00 Number of lines: len(L) / 2 Total number of times the loop body is executed: len(L) * len(L) / 2 Number of lines: len(L) / 10 Total number of times the loop body is executed: len(L) * len(L) / 10 Number of lines: len(L) Total number of times the loop body is

Total 1.00 / 1.00

Question 17

executed: len(L) * len(L)

For a call on function <code>insertion_sort(L)</code>, in the *worst case*, select the running time:

