

CSE 160 Section 4

Lists!

Logistics

Due 10/21: Practice Problems Check in #3

Due 10/28: HW 3 (Image Blur)



HW 2 Thoughts?

Lecture Key Points Review

Lists

- **Make a list**
 - `listname = [item0, item1, ..., itemn]`
- **Access an element**
 - `listname[i]` returns the item at index `i`
- **List Slicing**
 - `listname[a : b]` creates a list that contains the elements in between indexes `a` (inclusive) and `b` (exclusive).
- **Length**
 - `len(listname)` will return the number of elements in the list.



Functions that modify lists

- **Append**
 - `listname.append(item)` adds item to the end of the list
- **Insert**
 - `listname.insert(i, item)` puts the item at index `i`, and moves everything else to the right
- **Extend**
 - `list1.extend(list2)` will combine the contents of `list2` to the end of `list1`
- **Reverse**
 - `listname.reverse()` reverses the order of the list (in-place)
- **Sort**
 - `listname.sort()` sorts the list in ascending order (smallest to greatest) (in-place)



Kahoot!



PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Question 1

Q: Given the following list, what is the index for the value 89?

```
lst = [22, 45, 65, 89, 5]
```

Index: 0 1 2 3 4

A: 3

Remember that we count indices starting at 0



Question 2

Q: What is the output of the following code?

A: 45

Prints the value at index 1

```
lst = [22, 45, 65, 89, 5]  
print(lst[1])
```



Question 3

Q: What is the output of the following code?

A: [45, 65]

Prints the sublist starting at index 1 and stopping at (not including) index 3

```
lst = [22, 45, 65, 89, 5]  
print(lst[1:3])
```



Question 4

Q: What is the output of the following code?

A: [22, 45, 65, 89, 999, 5]

Inserts the value 999 before index 4.

999 is now at index 4, everything else pushed to the right.

```
lst = [22, 45, 65, 89, 5]
lst.insert(4, 999)
print(lst)
```



Question 5

Q: What is the output of the following code?

A: [1, 2, 3, 4, 5, 6]

Adds all the values of lst2 to the end of lst1

```
lst1 = [1, 2, 3]
lst2 = [4, 5, 6]
lst1.extend(lst2)
print(lst1)
```



Question 6

Q: What is the output of the following code?

A: [3, 2, 1]

Reverses the contents of the list (in place).

```
lst = [1, 2, 3]
lst.reverse()
print(lst)
```



Question 7

Q: What is the output of the following code?

A: [1, 2, 3]

Sorts the contents of the list in ascending order (in place).

```
lst = [2, 1, 3]
lst.sort()
print(lst)
```



Section Handout Problems

- We will go over problems 1-3, but solutions for all problems will be posted!
- Great practice - go to OH if you have questions!

1. What values would be printed when you run the following lines of code?

```
list_1 = [1, 2, 3, 4, 5]
list_2 = list_1
list_3 = list_1[:] # equivalent to list_1[0:5]
list_2[0] = 98
list_1[4] = 99
print("List 1:", list_1)
print("List 2:", list_2)
print("List 3:", list_3)
```



2. Given the following, modify `list_1` so it contains numbers 1 through 26 in increasing order:

```
list_1 = [1, 2, 3, 4]
list_2 = [10, 12]
list_3 = [21, 22, 23, 24]
list_4 = [13, 14, 15]
list_5 = [16, 17, 18, 19, 20, 25, 26]
list_6 = [9, 8, 7, 6, 5]
list_7 = [11]
```

a) using only the following operations:

- `list accesses []`
- `extend()`
- `insert()`
- `reverse()`
- `for loop`



2. Given the following, modify `list_1` so it contains numbers 1 through 26 in increasing order:

```
list_1 = [1, 2, 3, 4]
list_2 = [10, 12]
list_3 = [21, 22, 23, 24]
list_4 = [13, 14, 15]
list_5 = [16, 17, 18, 19, 20, 25, 26]
list_6 = [9, 8, 7, 6, 5]
list_7 = [11]
```

b) using the same operations as in part (a), but this time you are allowed to use the `sort()` function.

[Python Tutor](#)



3. Create a function `dot_product(list1, list2)` which takes in two lists of integers and returns their dot product.

```
def dot_product(list1, list2):
```



Thanks :)
