Name:_____

1. Strings in Python are immutable sequences containing only characters. A Python list is also a sequence collection, but a list can contain items of any type (e.g., character, strings, integers, floats, other lists, etc.). Lists are represented by comma-separated values enclosed in square brackets ('[', ']'). Operations on lists (**or any sequence collection**, e.g., strings) include:

Operation	**Operator**	**Explanation**	**Example** myList=[5,6,7,8] ListB=[8,9]	**Result of Example**
Indexing	[<index>]	Access the element specified by the index	myList[2]	7
Slicing	[:]	Extract a part of the list	myList[1:3]	[6, 7]
Concatenation	+	Combine lists together	myList + ListB	[5, 6, 7, 8, 8, 9]
Repetition	*	Concatenate a repeated number of times	ListB * 3	[8, 9, 8, 9, 8, 9]
Membership	in	Ask whether an item is in a list	3 in myList	False
Length	len(list)	How many items are in the list?	len(myList)	4

For the following lists, predict the results:

cheer = [2, 4, 6, 8, 'who', 'do', 'we', 'appreciate']
rhyme = [1, 2, 'buckle', 'your', 'shoe']

Expression	**Predicted Result**	**Actual Result**
`cheer[4]`		
`cheer[2:6]`		
`rhyme[:4]`		
`cheer[1:4] + rhyme[-2:]`		
`cheer[:2] * 3`		
`6 in rhyme`		
`len(cheer)`		
`[cheer[2:4]*4]`		

2. Lists in Python are mutable, i.e., they can be changed by assigning individual elements or slices new values.

For the following lists, predict the resulting lists:

Initial List Value	**Expression**	**Result**
`temp = ['a','b','c','d']`	`temp[1] = 99`	
`temp = ['a','b','c','d']`	`temp[1] = 'cat'`	
`temp = ['a','b','c','d']`	`temp[1] = ['cat', 'dog']`	
`temp = ['a','b','c','d']`	`temp[1:3]=[6, 7, 8, 9]`	
`temp = ['a','b','c','d']`	`temp[1:2] = 5`	

Name:_____

3. The following methods are provided by Lists in Python:

Method	Usage	Explanation
append	`myList.append(item)`	Adds item to the end of myList
extend	`myList.extend(otherList)`	Extends myList by adding all items in otherList to myList's end
insert	`myList.insert(i, item)`	Insert item in myList at index i
pop	`myList.pop( )`	Remove and return the last item in myList
pop(i)	`myList.pop(i)`	Remove and return the ith item in myList
del	`del myList[i]`	Deletes the item in the ith position of myList
remove	`myList.remove(item)`	Removes the first occurrence of item in myList
index	`myList.index(item)`	Returns the index of the first occurrence of item in myList
count	`myList.count(item)`	Returns the number of occurrences of item in myList
sort	`myList.sort( )`	Modifies myList to be sorted
reverse	`myList.reverse( )`	Modifies myList to be in reverse order

What would be the value of `myList` and `temp` after each of the following Python statements?

```
myList = range(5)
temp = range(10, 7, -1)

last = myList.pop( )

myList.extend(temp)

myList.insert(4, 3)

item = myList.pop(5)

del myList[2]
```

4. Suppose you have a small business and are maintaining customer information in a text file: customerData.txt. Each customer record is on a single line with 12 fields separated by commas(','). The order of the fields on a line is: First Name, Middle Initial, Last Name, Street Address, City, State, Zip Code, Country, Email Address, Telephone Number, Gender, and Birthday.

Write a program to read each line of text about a customer and generate a list of fields for that customer. Insert the list of fields for each customer into a customers list. Thus, the customers list is a list-of-lists where each item contains the information about a single customer. The general approach you should take is to:

- read each line from the file (contains all the information about a customer) as a string,
- split the line into a list of strings for the fields,
- insert this list of strings into the customers list.