Name:_____

1. A tuple is another sequence data type, so the sequence operations of indexing, slicing, concatenation, repetition, membership (in), and len() work on tuples too. Tuples are very similar to lists, i.e., comma-separated items enclosed in parenthese. The main difference is that **tuples are immutable** (cannot be modified).

Create two tuples as:
student1 = ('Bob', 123456, 'Jr', 3.12)
student2 = ('Sally', 654321, 'Fr', 0.0)

"Fields" of a tuple can be unpacked to its components using a single assignment statement as:
name, idnum, rank, gpa = student1

For the following sequence of Python statements, predict the result before executing each statement. Record the actual result if your prediction was wrong.

Statement	Predicted Result	Actual Result
student1[0]		
student1[2:]		
len(student1)		
student1[3] = 3.44		
student1 + student2		

2. A dictionary is an unordered set of key-value pairs (written as key:value). Keys must be unique and immutable (e.g., numerics, strings, tuples of immutable objects). Dictionaries are typically used to lookup the value corresponding to a specified key. Dictionaries can be written as comma-separated key:value pairs enclosed in curly braces. For example,

phoneNumbers = {'fienup':35918, 'gray':35917, 'east':32939,'drake':35811,'schafer':32187}

Access to individual key:value pairs looks syntactically like a sequence lookup using a key instead of an index. For example, `phoneNumbers['east']` returns 32939, and a new key:value pair can be added by `phoneNumbers['wallingford'] = 35919`. Additional, methods on dictionaries are:

Method	Usage	Explanation
keys	myDictionary.keys( )	Returns a list of keys in myDictionary
values	myDictionary.values( )	Returns a list of values in myDictionary
items	myDictionary.items( )	Returns a list of key:value tuples in myDictionary
get	myDictionary.get(myKey)	Returns the value associated with myKey; otherwise *None*
get	myDictionary.get(myKey, alt)	Returns the value associated with myKey; otherwise alt
in	myKey in myDictionary	Returns True if myKey is in myDictionary; otherwise False
has_key	myDictionary.has_key(myKey)	
del	del myDictionary[myKey]	Deletes the key:value pair whose key is myKey

Write code to:

- Add professor Poleksic's phone number 33388.

- Use the del method to delete professor Drake's entry.

Name:_____

For the following sequence of Python statements, predict the result before executing each statement. Record the actual result if your prediction was wrong.

Statement	Predicted Result	Actual Result
phoneNumbers.keys( )		
phoneNumbers.get(‘east’)		
phoneNumbers.get(‘me’) == None		
phoneNumbers.get(‘me’,’Not Found’)		
‘fienup’ in phoneNumbers		

3. Process the phoneNumber dictionary below to print the phone information one per line in alphabetical order by the name keys.

    phoneNumbers = {’fienup’:35918, ‘gray’:35917, ‘east’:32939,’drake’:35811,’schafer’:32187}