Name: _______________________________________________________________
NetID/Login: __________________________
Community Standard Acknowledgment (signature): __________________________

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1.</td>
<td>4 pts</td>
<td></td>
</tr>
<tr>
<td>Problem 2</td>
<td>6 pts</td>
<td></td>
</tr>
<tr>
<td>TOTAL:</td>
<td>10 pts</td>
<td></td>
</tr>
</tbody>
</table>

In writing code you do not need to worry about specifying the proper *import statements*. Don’t worry about getting function or method names exactly right. Assume that all libraries and packages we’ve discussed are imported in any code you write.
def getCows(guess, secret):
    """
    guess: Type str
    secret: Type str
    """
    cows = "" # String of cows to be returned
    alreadySeen = ""
    
    for i in range(len(guess)):
        if guess[i] != secret[i] and guess[i] in secret and guess[i] not in alreadySeen:
            cows = cows + "cow 
            alreadySeen += guess[i]
    return cows.strip()

What best describes the what this function returns:

A) Returns a string with “cow” for each character that is in secret but not the correct place in guess

B) Returns a string with “cow” if the two strings are different and the more different they are the more “cow” are appended to the returning string

C) Returns a string with “cow” for each character that secret and guess has in common, but are not in the correct place, but only if it is the first time this happens.

D) Returns a string with “cow” for each character from guess that is in secret but in the wrong place and that is already been there before.
PART B: (1 POINT)

Consider the following functions:

```python
import string
numLetters = 26

def lower_hide3(oneLetter):
    return string.lowercase[(string.lowercase.find(oneLetter)+3)%numLetters]

def upper_hide3(oneLetter):
    return string.uppercase[(string.uppercase.find(oneLetter)+3)%numLetters]

def encryptVersion_3(str):
    """
    str must be a string.
    """
    result = ""
    for c in str:
        if c.isupper():
            result = result + upper_hide3(c)
        elif c.islower():
            result = result + lower_hide3(c)
        else:
            result += c
    return result

What is printed with this call of the function encryptVersion_3:

```print encryptVersion_3("Duke Year: 2014")```
PART C: (1 POINT)

Consider the following dictionary:

```python
dict = {"Owl": 5, "Bear": 2, "Red Wolf": 14, "Red Robin": 14, "Beetle": 3}
```

The dictionary is sorted by the lines of code below:

```python
sortedPairs = sorted(dict.items(), key=operator.itemgetter(0))
sortedPairs = sorted(sortedPairs, key=operator.itemgetter(1), reverse=True)
```

Which of the following is the value of sortedPairs:

A) [('Red Robin', 14), ('Red Wolf', 14), ('Owl', 5), ('Beetle', 3), ('Bear', 2)]
B) [('Red Wolf', 14), ('Red Robin', 14), ('Owl', 5), ('Beetle', 3), ('Bear', 2)]
C) [('Bear', 2), ('Beetle', 3), ('Owl', 5), ('Red Robin', 14), ('Red Wolf', 14)]
D) [('Bear', 2), ('Beetle', 3), ('Owl', 5), ('Red Wolf', 14), ('Red Robin', 14)]
PART D : (1 POINT)

Which of the following is the list that results from the list comprehension:

\[
\text{[ num for num in range(12) if num/2 < 4 ]}
\]

A) [0, 1, 2, 3, ]

B) [0, 1, 2, 3, 4, 5, 6, 7, 8]

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

D) [0, 1, 2, 3, 4, 5, 6, 7]
PROBLEM 2 : ( 6 POINT )

Data for parts of Barcelona’s soccer team 2012 is given below in a dictionary.

```python
barcelona = {
    "Messi": [(4022, 11, 8)],
    "David Villa": [(1834, 2, 1)],
    "Xavi": [(3210, 5, 1)],
    "Jordi Alba": [(3260, 5, 2)]
}
```

The statistic for each player is a tuple element in a list. The values of the tuples are:

```
[(Minues played, assist, goal scored)]
```

Complete this function so that it returns a list of tuples where the first tuple element is the name of a player and the second is how many goal he scored. For the example in Barcelona given above the function should return:

```
[('Messi', 8), ('Jordi Alba', 2), ('Xavi', 1), ('David Villa', 1)]
```

```python
def goalsScored( dataDict ):
    return [ (name, statLst[0][2]) for (name, statLst) in dataDict.items() ]
```

NOTE: The difficult part in this type of assignment is the nesting; a tuple within a list in a dictionary. To help you crack problems like this it is a good idea to write a helping line to yourself with the content of your variables. For example:
```
(name, statLst) were statLst = [(4022,11,8)]
```
OR
```
dataDict.items() = [ (“Messi”, [(4022,11,8)]),
                     “DavidVilla”, [(1834,12,1)]) ... ]
```