CS115 - Test 3 Worksheet

Question 1 *(20 points)* **Assess:** [execution]
(a) *Show the output* that gets printed by the following code.

(b) *Draw a box-and-arrows diagram* to show the data at the time `print` is executed. For integers, you can draw them in place rather than as references.

```
L = [[1,2],[3,4]]
M = list(L)
L[1][1] = 5
M[0] = [6,7]
print(L,M)
```

Question 2 *(20 points)* **Assess:** [execution] This question is about the following code.

```
def numMatches( L1, L2 ):
    '''return the number of elements that match between two sorted lists'''
    i = 0
    j = 0
    matches = 0
    while i < len(L1) and j < len(L2):
        if L1[i] == L2[j]:
            matches += 1
            i += 1
            j += 1
        elif L1[i] < L2[j]:
            i += 1
        else:
            j += 1
    return matches

M1 = ["Chance", "DJ Shadow", "Khaled", "Meshell Ndegeocello", "St Vincent", "Travi$"]
M2 = ["Alicia Keys", "Chance", "Khaled", "Lila Downs", "Meshell Ndegeocello"]
```

Your job: make a loop trace, for the call `numMatches(M1,M2)`. That is, fill in the table below, tracing the values of the variables `i`, `j`, and `matches` for these lists. The first row already shows the initial values. Just add one row for each iteration of the loop, that shows their values after that iteration.

```
i    j    matches
-----------------
0    0    0
```

Question 3 *(20 points)* **Assess:** [coding] Function `wordScore` computes word scores for the Scrabble game. Implement `wordScoreLoop` so it does the same thing, but using a for- or while-loop instead of recursion.

```
letterScores = {'a': 1, 'b': 3, 'c': 3, 'd': 2, 'e': 1, 'f': 4, 'g': 2}

def wordScore(S):
    '''Assume S is a string.  Return the scrabble score of S, using letterScores.
    For letters not in letterScores, the score is 0.
       For example, wordScore("eagle") is 5 (i.e., 1 + 1 + 2 + 0 + 1). '''
    if S == '':
        return 0
    elif S[0] in letterScores:
        return letterScores[S[0]] + wordScore(S[1:])
    else:
        return wordScore(S[1:])

def wordScoreLoop(S):
    ''' Same as wordScore, but implemented with a loop. '''
```

Question 4 *(10 points)* **Assess:** [class] Write a python program that ask the users to input a positive integer n and then prints the hollow square with side length n as shown below for n = 5.

```
* * * * *
*       *
*       *
*       *
* * * * *
```

Question 5 *(20 points)* **Assess:** [class] This question is about the following code.

```
class Player:
    def __init__(self, name, genre):
        self.name = name
        self.genre = genre
        self.instruments = []

    def __str__(self):
        '''The artist and their instruments'''
        return "Artist " + self.name + " plays " + ", ".join(self.instruments)

    def copy(self):
        p = Player(self.name, self.genre)
        p.instruments = list(self.instruments)
        return p

    def addInst(self, instrument):
        self.instruments.append(instrument)


Meshell = Player("Ndegeocello", "rap")
Meshell.addInst("bass")
M2 = Meshell.copy()
M2.addInst("piano")
print(Meshell)
```

(a) The code prints one of the following two lines. *Circle the correct answer.*

```
Artist Ndegeocello plays bass

Artist Ndegeocello plays bass, piano
```

(b) *Justify your answer* by drawing a box-and-arrows diagram for the variables `Meshell`, `M2`, and the objects they reference.

Question 6 *(5 points)* **Assess:** [coding] Implement the following method that would be part of class `Player`.

```
    def __eq__(self,other):
        '''Whether self and other have same name and genre, and play the same instruments.''
```

This should return true even if the instruments were added in a different order. For example, the following code should print `True`:

```
p0 = Player("Attila Duck", "jazz")
p0.addInst("kazoo")
p0.addInst("guitar")
```

```python
p1 = Player("Attila Duck", "jazz")
p1.addInst("guitar")
p1.addInst("kazoo")
print(p0 == p1)
```