CS303E Week 10 Worksheet: More on Lists

Name: _____ EID: _____

Read the questions carefully, and answer each question in the space provided. Use scratch paper to do your work and then copy your answers neatly and legibly onto the test paper. Only answers recorded on the test paper will be graded.

1. (10 points: 1 point each) The following are true/false questions. **Write either T or F in the boxes at the bottom of page 1.** If there's any counterexample, it's false.

   (a) The expression `myList[2][3]` is valid for a nested list with three inner lists, each having at least four elements.

   (b) Accessing every element within a list of lists can be done with a nested loop.

   (c) Binary search is guaranteed to return the index of the first (leftmost) instance of an item in a list.

   (d) In a list of lists, all elements must be the same length.

   (e) The `len()` function can be used to find the total number of elements in a nested list. Example: `len([[1, 2],[3,5]]) == 4`.

   (f) When using the `.sort()` method on nested lists in Python, the sorting is applied only to the outermost list, leaving the inner lists unaltered.

   (g) In a linear search, one item is checked in the unexplored portion of the list each step, while in binary search, the search space is cut in half with each step.

   (h) Binary search does not work on an unsorted list.

   (i) Binary search always uses fewer comparisons than linear search.

   (j) `[random.randint(0, 99) for i in range(100)]` returns a list of 100 random integers between 0 and 99?

a	b	c	d	e	f	g	h	i	j

Page total: _____/10

Questions 2-8 are multiple choice. Each counts 2 points. **Write the letter of the BEST answer in the box on page 3. Please write your answer in UPPERCASE. Each problem has a single answer.**

2. In which scenario would using binary search likely be more advantageous than linear search?

   A. Searching through a small, unsorted list.
   B. Looking for the first occurrence of an item in a list.
   C. Searching through a large, sorted list.
   D. Finding the maximum value in an unsorted list.
   E. None of the above.

3. What does the following print?

```
L = [n for n in range(10) if n % 3]
print( L )
```

   A. `[0, 3, 6, 9]`
   B. `[3, 6, 9]`
   C. `4`
   D. `[1, 2, 4, 5, 7, 8]`
   E. none of the above

4. Given the following code, what is printed?

```
def func2(L):
    x = 0
    for i in range(len(L)):
        for j in range(len(L[i])):
            x += L[i][j]
    return x

2DList = [[1, 2], [3, 4]]
print( func2( 2DList ) )
```

   A. `[1, 2, 3, 4]`
   B. 24
   C. 10
   D. it's an error
   E. none of the above

5. How many comparisons does a linear search require to find the value 616 in the list `[9, 10, 21, 41, 98, 123, 364, 616, 1218]`?

   A. 6 B. 7 C. 8 D. 9

6. How many comparisons does a binary search likely require to find the value 616 in the list `[9, 10, 21, 41, 98, 123, 364, 616, 1218]`?

   A. 1 B. 2 C. 3 D. 4 E. 8

7. How could you remove `"G"` from the following nested list?:

```
Letters = [["A", "B", "C"], ["D", "E", "F", "G"], ["H", "I", "J"]]
```

   A. `Letters.remove("G")`
   B. `Letters.pop(1)`
   C. `Letters[1].remove("G")`
   D. `Letters[1].pop("G")`

8. How do you check if the string `"mew"` is present anywhere in the following nested list?:

```
x = [["eevee", "togepi"], ["mew", "mewtwo"], ["dragonite", "gengar"]]
```

   A. `"mew" in x`
   B. `True if "mew" in [item for item in x] else False`
   C. `"mew" in x[1]`
   D. `"mew" in x[1:1]`

2	3	4	5	6	7	8

Page total: _____/14

The following questions (2 points each) ask you to trace the behavior of some Python code and identify the output of that code. For each question, write the output for the code segment in the box provided. Don't worry about line breaks. If there's an error, say so, but you don't need to specify the type of error.

9.
```
queensLists = [["ariel"], ["belle", "tiana"], \
                ["mulan", "pocahantas", "rapunzel"]]
result = [homegirl for lst in queensLists for homegirl in lst]
print(result)
```

10.
```
inputNums = [13, 17, 25, 28, 10, 30, 21]
myLst = [ [0, 0, 0], [0, 0, 0], [0, 0, 0] ]

for num in inputNums:
    result1 = num % 3
    result2 = (num // 10) % 3
    myLst[result1][result2] += 1
print(myLst)
```

11.
```
powerpuff = [["blossom", "mojoJojo"], \
             ["bubbles", "utonium"], \
             ["buttercup", "bunny"]]
powerpuff.sort()
print(powerpuff)
```

12.
```
lst1 = [[1, -100], [5, 10]]
lst2 = [[1, -900], [999, 999]]
print(lst1 > lst2)
```

Page total: _____/8

13.
```
board = [["H", "X", "Z", "C"], ["A", "O", "T", "J"], \
         ["K", "P", "N", "D"], ["L", "M", "U", "K"]]

for i in range(len(board)):
    for j in range(len(board)):
        if i == j:
            print(board[i][j], end = "")
```

14.
```
sanrio = [["keroppi", "helloKitty"], \
          ["pompompurin", "myMelody", "cinnamoroll"], \
          ["chococat", "badtzmaru"]
cuties = [group[0] for group in sanrio]
print(cuties)
```

15.
```
looney = [["bugs", "daffy"], \
          [1930, 4], \
          ["tweety", "sylvester", "lola"]]
print(looney[-1][-2][0])
```

16.
```
myNums = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
total = 0
for i in range(len(myNums)):
    for j in range(4):
        if not j % 2:
            total += myNums[i][j]
print(total)
```

Page total: _____/8