Midterm

04/11/2019

Name_____

1. (10 Pts) What is the output of the following Python programs? If the execution fails at run-time, write the output up until the failure happens, then write FAIL.

PROGRAM A

```
a = set()
b = "test"
a.add(b)
a.add("test")
print(len(a))
```

PROGRAM B

```
s = "abba is a band, is abba"
a = []
cnt = 0
for item in s.split():
     if 'bb' in item:
          a.append(item)
print("Len1: {} Len2: {}".format(len(s), len(a)))
```

PROGRAM C

```
h = {"yes":[1,2]}
h["no"] = [3,4]
for item in h.values():
     print(item[0])
print(h[0])
```

PROGRAM D

```
for i in range(0,9,5):
     print(i)
```

2. (10 Pts) Indicate whether the following strings are matched **entirely** (with re.search()) by the regular expression "aa*bb*|b+|(bb|aa)*". (Write "A" if accepted, and "N" if not accepted in the blank provided.)

a) "aabb" _____

b) "bbb" _____

c) "aaa" _____

d) "aaaa" _____

3. **(5 points):** Given the data in the table below, fill in the contingency table for Cohen's Kappa:

	Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**	**Item 6**	**Item 7**	**Item 8**	**Item 9**	**Item 10**	**Item 11**
Rater 1	Y	Y	Y	Y	Y	N	N	N	Y	Y	N
Rater 2	Y	Y	Y	Y	Y	N	N	N	N	N	Y

		Rater 1	
		Y	**N**
Rater 2	**Y**		
	N		

4. **(5 points):** Using the contingency table from the previous question, calculate the overserved agreement for Cohen's Kappa (Po is the probability that the two annotators agree) – I will accept fractions.

Hint: Po **=** P(Rater 1 = Rater 2)

5. (5 Pts) Which of the following is NOT a valid JSON object (I have prettified the files to make it easy to read. So, ignore indentation and spacing):

FILE A

```
{ "name": "Anthony",
  "age": 20,
  "phone": "555-555-5555",
  "email": "anthony@xyz.com",
  "happy": yes }
```

FILE B

```
{ "name": "Anthony",
  "age": “20”,
  "phone": "555-555-5555",
  "email": “anthony@xyz.com” }
```

FILE C

```
{ "name": "Anthony",
  "age": 20,
  "phone": "555-555-5555",
  "email": "anthony@xyz.com" }
```

Write your answer in the following blank (File A, B, or C): _____

Why did you choose that answer?

6. (5 Pts) Which of the following is NOT a valid JSON list object:

FILE A

```
[ {"pet":"dog", "fun":"true"}, {“pet":"dog", "fun":"true"} ]
```

FILE B

```
 {"pet":"dog", "fun":"false"}, {“pet":"dog", "fun":"true"}
```

FILE C

```
[ {"dogAge":3} ]
```

Write your answer in the following blank (File A, B, or C): _____

Why did you choose that answer?

7. (10 Pts) You have data in two formats – JSON and CSV. The datasets contain the same type information. You want to combine both datasets into a single list. Fill in the blanks to achieve the desired output.

FILE: jsonFile.json

```
[ {"name":"Anthony","Age":102}, {"name":"John","Age":12} ]
```

FILE: csvFile.csv

```
Name|Age
Jane|43
Elizabeth|84
```

CODE: myCode.py

```
import csv
import json

jsonFile = open("jsonFile.json")
data = json.load(jsonFile)
jsonFile.close()

csvFile = open("csvFile.csv")

iCSV = csv.reader(csvFile, delimiter= _____________________)
next(iCSV)
for row in iCSV:
      newDict = {"name":row[0]}

      newDict[_____________________] = _____________________________
      data.append(newDict)
csvFile.close()

for item in data:

      print(_____________________________________________)
```

OUTPUT

```
Anthony
John
Jane
Elizabeth
```

8. (10 Pts) Write the ENTIRE output of the following programs:

FILE A: example.py

```
cnt = 0
mySum = 0
print("cnt: {} mySum: {}".format(cnt, mySum))
while cnt < 5 or mySum < 5:
      if cnt % 2 == 1:
            mySum += 1
            print("cnt: {} mySum: {}".format(cnt, mySum))
      cnt += 1
```

FILE B: example.py

```
text = "How much wood would a Woodchuck chuck if a woodchuck could chuck wood"
cnt1 = 0
myDict = {}
for word in text.split():
      if word in myDict:
            myDict[word.lower()] += 1
      else:
            myDict[word.lower()] = 1
      if 'wood' in word:
            cnt1 += 1
print("cnt1: {}".format(cnt1))
print("myDict: {}".format(myDict['wood']))
```

EXTRA CREDIT (10 Pts) Complete the code for the following function so it matches its documentation:

FILE: example.py

```
def doubleList(numberList):
     ''' For each of the numbers in the list numberList,
         print a line containing twice the original number.
         For example, doubleList([3, 1, 5]) would print 6 2 10,
         where each number is printed on a new line. Finally,
         return the sum of all the doubled numbers.

         :param list numberList: A list of ints
         :return: int return the sum of all the doubled numbers
     '''
```