Due: Wed, Oct 23, 2013

Homework 11 is designed to give you a good preview of the format and content of the second midterm (Oct 28). It consists of 12 multiple choice questions and 3 programming problems.

When you take the midterm you will not have access to a Python programming environment. Therefore, it is essential to know how to read code and determine what it is going to do without running it. (This is very much like real life, where you should know what code will do even before you write it.)

For each multiple choice question, submit two answers: the one you get before you consult a Python interpreter and the one you get from the Python interpreter. When these two answers are different, include a sentence explaining why they are different.

Upload your answers as a .py file in Moodle.
Question 1

```python
j = 1
for i in range(5):
    temp = j
    j += i
    i = temp
    print(j, end = ' ')
```

a. 1 2 3 5 8
b. 1 2 3 5

e. 1 2 3

Question 2

```python
race = ['color', 'me', 'rad', '5k']
whaler = ['Call', 'me', 'Ishmael']
beginning = []
for i in range(len(race)):
    if len(beginning) < len(whaler):
        beginning.append(whaler[i])
    elif len(beginning) == len(whaler):
        break
    else:
        beginning.append(race[i])
print(beginning)
```

a. []
b. ["Call", "me", "Ishmael"]
c. ['Call', 'me', 'Ishmael']
d. ['Call', 'me', 'Ishmael', '5k']
e. None of the above

Question 3

```python
palindrome = 'hannah'
print(palindrome[3:] + palindrome[:3])
```

a. hannah
b. nahhan

Question 3

```python
palindrome = 'hannah'
print(palindrome[3:] + palindrome[:3])
```

a. hannah
b. nahhan

Question 3

```python
palindrome = 'hannah'
print(palindrome[3:] + palindrome[:3])
```

a. hannah
b. nahhan
Question 4
sibilants = ['s', 'z']
word = 'sizings'
sibilantCount = 0
for letter in word:
    if letter in sibilants:
        sibilantCount += 1
print(sibilantCount)

a. 0
b. 1
c. 2
d. 3
e. None of the above

Question 5
def vowelCheck(testStr):
    vowels = ['a', 'e', 'i', 'o', 'u']
    for letter in vowels:
        if letter not in testStr:
            return "no " + letter
    if len(testStr) == 0:
        return "no letters"
    else:
        return "no vowels"
s = 'even'
print(vowelCheck(s))

a. no a
b. no a no i no o no u
c. no letters
d. SyntaxError: hanging else
e. None of the above

Question 6
constants = [["e", 2.17, "pi", 3.14], 0, 1, "i"]
print(constants[1:3])

a. TypeError: incompatible types: list in list
b. ValueError: float value of str
c. 2.17, "pi"
d. 0, 1, "i"
e. None of the above

Question 7
dinner = [['protein',['turkey']],['fruit',['cranberry','pumpkin']],["carb", ['stuffing']]]
print(dinner[1][1])

a. SyntaxError: invalid syntax
b. TypeError: incompatible types 'fruit', ['cranberry','pumpkin']
c. 'cranberry','pumpkin'
d. ['cranberry','pumpkin']
e. None of the above
Question 8

```python
def mostlyLongWords(aStr, threshold):
    splitLine = aStr.split()
    shortCount = 0
    longCount = 0
    for word in splitLine:
        if len(word) < threshold:
            shortCount += 1
        else:
            longCount += 1
    if shortCount > longCount:
        return False
    if longCount > longCount:
        return True
    return =''

seussLine = 'There is no one alive who is you-er than you'
print(mostlyLongWords(seussLine, 5))
```

a. True
b. False
c. =
d. SyntaxError:incompatible return types
e. None of the above

Question 9

```python
T = True
F = False
boolExprs = [T and F, T or F, T and T, T or T, F and F, F or F]
trueCount = 0
for expr in boolExprs:
    if expr == True:
        trueCount += 1

print(trueCount)
```

a. 1
b. 2
c. 3
d. 4
e. None of the above
Question 10

def testReturn(aStr, subStr):
    if aStr.count(subStr) == 0:
        return(""")
    if aStr.count(subStr) == 1:
        return(subStr)
    if aStr.count(subStr) == 2:
        return(subStr + subStr)
    return(aStr)
s0 = 'ohboyohboyohboy'
s1 = 'boy'
print(testReturn(s0, s1))
a. ""
   b. boy
   c. boyboy
   d. ohboyohboyohboy
   e. Syntax error: multiple return statements

Question 11

bigCities = open('bigCities.txt', 'w')
bigCities.write('New York')
bigCities.close()
bigCities = open('bigCities.txt', 'r')
print(bigCities.read())
bigCities.close()

a. New YorkMoscow
   b. New York
   Moscow
   c. NameError: name 'bigCities' is already defined
   d. NameError: name 'bigCities' is not defined
   e. None of the above

Question 12

def repeatWords(sentence):
    repeats = []
    words = sentence.split()
    for word in words:
        if word not in repeats:
            repeats.append(word)
    return repeats
albert = "Insanity: doing the same thing over and over again and expecting different results"
print(repeatWords(albert))
a. ""
b. ['over']
c. ['over', 'and']
d. ['over', 'and', 'over', 'and']
e. None of the above
**Question 13**

We define the letters 'a', 'e', 'i', 'o' and 'u' as vowels. We do not consider any other letter as a vowel.

Write a function named initialVowels() that returns a list of words in a body of text that begin with a vowel. Include both capitalized and lower case instances of the vowels. A word should appear in the return list at most once, no matter how many times it occurs in the input string.

**Input:** a string that consists of words, separated by spaces  
**Return:** an list of words in the input string that begin with an upper or lower case vowel

For example, the following would be correct output:

```python
>>> mlk = 'Our lives begin to end the day we become silent about things that matter'
>>> print(initialVowels(mlk))
['Our', 'about']
```

**Question 14**

The three words 'a', 'an' and 'the' are the only articles in the English language. Write a function named countArticles(). Hint: Count both capitalized and lower case instances of articles.

**Input:** a string, named sentence  
**Return:** the number of words in sentence that are articles

For example, the following would be correct output:

```python
>>> theFlea = ['The flea is a mighty insect']
>>> print(articleCount(theFlea))
2
```

**Question 15**

Write a function named pluralCount() that takes two string parameters. The first parameter is the name of an input file that exists before pluralCount() is called. The second parameter is the name of an output file that pluralCount() creates and writes to. You may assume that the input file is in the current working directory and you should write the output file to that directory as well.

For each line in the input file, the function pluralCount() should write to the output file the number of words in the line that end in the letter 's'.

For example, if the following is the content of the file foxInSocks.txt:

```
Look, sir. Look, sir. Mr. Knox, sir.
Let's do tricks with bricks and blocks, sir.
Let's do tricks with chicks and clocks, sir.
```

The following function call:

```python
inF = 'foxInSocks.txt'
outF = 'foxRepLines.txt'
pluralCount(inF, outF)
```

should create the file ‘foxRepLines.txt’ with the content:

```
0
4
4
```