CS 1110 Prelim 2 April 21, 2015

(Print Last Name)

(Print First Name)

(Net ID)

Circle Your Lab:

ACCEL:	Tue 12:20	Tue 1:25	Tue 2:30	Tue 3:35		
ACCEL :	Wed 10:10	Wed 11:15	Wed 12:20	Wed 1:25	Wed 2:30	Wed 3:35
PHILLIPS :	Tue 12:20	Tue 1:25	Wed 12:20			

Write your name and NetID on every page.

Check that your exam has (11) pages counting this one.

In cases where you write code, you will lose points for ambiguous indentation.

Academic Integrity

It is a violation of the Academic Integrity Code to look at any exam other than your own, to look at any other reference material, or to otherwise give or receive unauthorized help. We also ask that you not discuss this exam with students who are scheduled to take a later makeup. Academic Integrity is expected of all students of Cornell University at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare I shall not give, use or receive unauthorized aid in this examination.

(Signature)

Problem 1	15 points	
Problem 2	15 points	
Problem 3	15 points	
Problem 4	20 points	
Problem 5	15 points	
Problem 6	20 points	

1 Functions and Lists

(a) The *even-odd* sort of a list that has even length permutes entries so that all the even-index entries come first followed by all the odd-indexed entries. To illustrate, suppose we have the following length-8 list:

’a’	’b’	’c’	’d’	’e’	’f’	’g’	’h’

Here are the length-4 lists of the even-indexed entries and the odd-indexed entries:

’a’	’c’	’e’	’g’

’b’	’d’	’f’	’h’

And here is the even-odd sort of the above length-8 list:

’a’	’c’	’e’	’g’	’b’	’d’	’f’	’h’

This operation can be carried out very simply using list slicing and list concatenation. Indeed if `x` has length `n` and `n` is even, then the list `x[0:n:2] + x[1:n:2]` is the even-odd sort of `x`. Implement that following procedure so that it performs as specified

```
def EvenOddSort(x)
    """ Performs an even-odd sort of x

    """ Precondition: x is a list with even length
```

Implement this function using just for-loops and subscripting. No list slicing or list concatenation allowed. Note that `EvenOddSort` does not return any values.

```
Solution:



This problem builds on their perfect shuffle experience.

     n = len(x)
     m = n/2
     y = []
     z = []
     for k in range(m):
         y.append(x[2*k])
         z.append(x[2*k+1])
     for k in range(m):
         x[k] = y[k]
         x[k+m] = z[k]
```

(b) Assuming that the procedure `EvenOddSort` is available, implement the following function so that it performs as specified:

```
def MultipleSort(x,N):
    """ Returns a list obtained by performing N even-odd
    sorts of the list x. The list x is not altered’

    Precondition: x is a list with  even length and N is a positive int.
    """
```

Solution

Again, this counts on them understanding the perfect shuffle problem.

```
y = []
for a in x:
    y.append(a)
for k in range(N):
    EvenOddSort(y)
return y
```

2 Farthest Point

Assume the existence of the following class:

```
class Point:
   """ Attributes:
          x   the x-coordinate   [float]
          y   the y-coordinate   [float]
   """
   def __init__(self,x,y):
       self.x = x
       self.y = y

   def Dist(self,other):
       """ Returns a float that is the distance from self to other.

       Precondition: other is a Point
       """
       return sqrt((self.x-other.x)**2+(self.y-other.y)**2)
```

Complete the following function so that it performs as specified

```
def FarthestPt(L,idx,P)
   """ Returns an integer j with the property that the distance from
   L[j] to P is maximum among all the unvisited points

   If idx[i] = 1, then we say that L[i] has been visited. If idx[i] = 0, then
   we say that L[i] is unvisited.

   Preconditions: L is a list of references to Point objects, P is a reference
   to a point object, and idx is a list of ints that are either zero or 1. The
   lists idx and L have the same length and idx has at least one zero entry.
   """
```

Solution

Counts on the student being familiar with the Traveling fanatic problem (A5) combined with experience using methods. (Lec 20)

```
d = 0
for j in range(len(L):
    dj = P.Dist(L[j])
    if idx[j]==0 and dj>d
         k = j
         d = dj
return k
```

Name: ______________________________ NetID: ______________

3 Nested Loops

(a) What is the output if the following is executed?

```
s = 'abcd'
for i in range(4):
    for j in range(i+1,4):
         print s[i]+s[j]
```

```
solution


       ab
       ac
       ad
       bc
       bd
       cd
```

(b) Describe in English what the following script does and indicate clearly what the first, last, and second to last lines of output are. (Hint. You may find it useful to think about your solution to part (a).)

```
s = 'abcdefghijklmnopqrstuvwxyz'
for i in range(26):
    for j in range(i+1,26):
        for k in range(j+1,26):
             print s[i]+s[j]+s[k]
```

```
      Solution

An alphabetical list of all 3-letter words that have  no repeat characters



             wyz
             xyz
```

(c) What are the lines of output if the following is executed? (The order is not important.)

```
D1 = {’a’:’one’, ’b’:’two’, ’c’:three, ’d’:’four’}
D2 = {’c’:’five’, ’d’:’six’, ’e’:seven,’f’:’eight’}
D = {}
for d in D1:
   D[d] = D1[d]
for d in D2:
   D[d] = D2[d]
for d in D
   print d,D[d]
```

```
Solution

      a     one
      b     two
      c     five
      d     six
      e     seven
      f     eight
```

4 What's the Output

(a) If the following is executed, then what is the output?

```
x = [10,20,30]
for k in range(1000):
     x.append(x[0])
     x = x[1:4]
print x
```

Solution

```
 20 30 10
 30 10 20
 10 20 30

   etc
```

So back to where we started from after 3 iterations. Therefore, back to where we started from after 999 iterations. We do one more and get

```
20 30 10
```

(b) If the following is executed, then what is the output?

```
x = [10,20,30,40]
y = x
for k in range(4):
    x[k] = y[3-k]
print x
```

x and y are aliased so here are the changes in x

```
Start     10  20  30  40
k=0       40  20  30  40
k=1       40  30  30  40
k=2       40  30  30  40
k=3       40  30  30  40
```

(c) If the following is executed, then what is the output? For full credit you must also draw two state diagrams. The first should depict the situation just after the `Q.x = 0` statement and the second should depict the situation just after the `P = Point(7,8)` statement.

```
P = Point(3,4)
Q = P
Q.x = 0
print Q.x, Q.y, P.x, P.y
P = Point(7,8)
print Q.x, Q.y, P.x, P.y
```

```
(Note: the Point class referenced here is defined in Question 2 of the exam.)

Solution:

     At the first print statement, P and Q are aliased. So

                          0  4   0   4

     At the second print statement they refer to different Point objects so

                          0  4  7  8
```

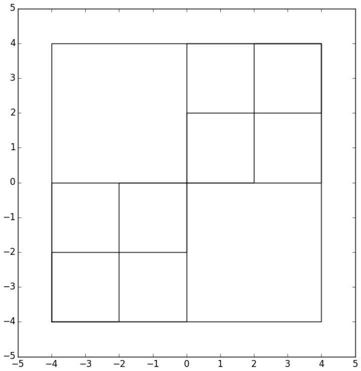
(d) Sketch the figure that is produced when following module is run.

```
from simpleGraphicsE import *

def DrawSquare(x,y,s,Level):
    # Draws the outline of a square with center (x,y) and side length s.
    DrawRect(x,y,s,s)
    if Level>0:
        DrawSquare(x+s/4,y+s/4,s/2,Level-1)
        DrawSquare(x-s/4,y-s/4,s/2,Level-1)

if __name__ == '__main__':
    MakeWindow(5)
    DrawSquare(0,0,8,2)
    ShowWindow()
```

Name: ______________________________ NetID: ________________



Name: ______________________________ NetID: ________________

5 Lists of Objects and Dictionaries

Recall the class Sonnet that you worked with in A6:

```
class Sonnet:
    """
    attributes:
         index       a string that encodes the sonnet number as a Roman Numeral
         text        a length-14 list of strings that encodes the sonnet. (Each string a sonnet line).
         firstLine   a string that is the first line of the sonnet without punctuation
         nWords      an int that is the number of words in the sonnet
         lastWords   a length-14 list of strings, one for the last word in each line
    """
```

(a) Suppose L is a list of references to Sonnet objects and that j satisfies $0 \leq j < 14$. Write code that prints out the jth line of every sonnet that is referenced by L.

Solution:

```
     for S in L:
          print S.text[j]
```

(b) Suppose L is a list of references to Sonnet objects. Give a complete implementation of a function LastD(L) that returns a dictionary whose keys are sonnet indices like 'CXIII' and whose values are the corresponding lists of last words. You must include a full specification.

Solution:

```
     def LastD(L):
        D = {}
        for S in L:
            D[S.index] = S.lastwords
        return D
```

(c) Complete the following function so that it performs as specified

```
    def  F(s,D):
       """ Returns True if s is a key for D and every element in D[s] is also
       a key in D. Otherwise returns False.

       Precondition: s is a nonempty string and D is a dictionary whose keys are strings
       and whose values are lists of strings.

       """

   Solution
             if s not in D:
                 return False
             else:
                 for t in D[s]:
                     if t not in  D
```

```
         return False
return True
```

6 Short Answer

(a) Explain the difference between a class and an object.

(b) Assume that `F(x)` is a function that works only if it is passed an argument that is an integer. A student writes the following script with an eye towards checking out the values that `F` returns:

```
while True:
    x = raw_input(’Enter x: ’)
    try:
        x = int(x)
    except:
        break
y = F(x)
print x,y
```

Explain why the script is NOT helpful.

(c) What is revealed when the `timeit` module is used to compare the efficiency of linear search and binary search?

(d) Assume that `L` is a list of references to Sonnet objects. (See Problem 5 for a description if the Sonnet class.) Does the statement `N = sum(L.nWords)` assign to `N` the total number of words in all the Sonnets referenced by `L`? Explain.

Name: ____________________ NetID: ____________

Function Information

Function	What It Does
`len(s)`	returns an `int` that is the length of string `s`
`s.count(t)`	returns an `int` that is the number of occurrences of string `t` in string `s`
`s.find(t)`	returns an `int` that is the index of the first occurrence of string `t` in the string `s`. Returns -1 if no occurrence.
`s.replace(t1,t2)`	returns a string that is obtained from `s` by replacing all occurrences of `t1` with `t2`.
`floor(x)`	returns a float whose value is the largest integer less than or equal to the value of `x`.
`ceil(x)`	returns a float whose value is the smallest integer greater than or equal to the value of `x`
`int(x)`	If `x` has type `float`, converts its value into an `int`. If `x` is a string like '-123', converts it into an `int` like -123
`float(x)`	If `x` has type `int`, converts its value into a `float`. If `x` is a string like '1.23', converts it into a `float` like 1.23.
`str(x)`	Converts the value of `x` into a string.
`DrawRect(x,y,L,W)`	Draws a rectangle with center (x, y), horizontal dimension L, and vertical dimension W.
`DrawDisk(x,y,r)`	Draws a circle with center (x, y) and radius r.
`DrawStar(x,y,r)`	Draws a star with center (x, y) and radius r.
`DrawLineSeg(x,y,L,d)`	Draws a length L line segment that starts at (x, y) and makes counterclockwise angle of d degrees with the positive x-axis.
`x.append(y)`	adds a new element to the end of the list `x` and assigns to it the value referenced by `y`.
`deepcopy(x)`	creates a complete copy of the object that is referenced by `x`.
`sum(x)`	returns the sum of the values in list `x` assuming that all its entries are numbers.