Practice Problems

1. What will this print?

```
def printer( L ):
   # L should be a list
        if len(L) == 1:
                print( L[0] )
        else:
                print( L[-1] )
                printer(L[0:-1])

def main():
        printer( [1, 2, 3, 4, 5] )
main()
```

2. What will this print?

```
def printer(L, M):
        for i in L:
                print( M[i] )
def main():
        printer( [2, 0, 1, 2], ["I", "Like", "Python", "Coding"] )
main()
```

3. In the followng program Python is rude enough to flag an error on the print(L) line in main( ), claiming that L is not defined. What is wrong?

```
def BuildList():
        L = []
        done = False
         while not done:
                n = eval(input( "Enter a number or 0 to exit: " ))
                 if n == 0:
                        done = True
                else:
                        L.append(n)
        return L
def main():
        BuildList()
        print( L )
main()
```

4. Consider the following annoying program:

```
def main():
        x = -1
        while x != 0:
                x = eval(input("gimmee a number: " ))
                if x == 0:
                        done = True
                x = 13
        print( "I like %d better than your number." % x)

main()
```

This program never halts, even when I enter 0 and the program sets variable Done to True. Why doesn't it halt? [Note: don't rewrite the program; tell me why this version doesn't halt.

5. Here is a recursive function that sometimes works and sometimes doesn't. If I call countDownBy2(8) it prints

```
8
6
4
2
0
```

and stops. But for other numbers it doesn't stop. For this question say both what is wrong with this function and how to rewrite it so that it halts for any argument.

```
def countDownBy2(n):
        print(n)
         if n != 0:
                countDownBy2(n-2)
```

6. Consider the following program, which stores information about a person's friends in a dictionary. The key is a person's name; the value associated with that key is a list of the person's friends. If I enter "john mary" it is fine. If I then enter "mary fred" it is fine. If I then enter "fred mary" it is still fine. But if I enter next "john suzie" it crashes. Do you have a better explanation than that the program just likes monogamy? Maybe it doesn't like suzie?

```
def main():
        D = { }
        done = False
        while not done:
                pair = input( "Enter a pair of friends, or a blank line to exit: ")
                if pair == "":
                        done = True
                else:
                        pair = pair.strip( "\n" )
                        (a, b) = pair.split()
                         if a in D.keys():
                                D[a].append(b)
                        else:
                                D[a] = b
        for x in D.keys():
                print(x, D[x])

main()
```

7. The following program is badly written. Class B doesn't even have a constructor. But will it run or crash? If it runs, what will it print?

```
class A:
        def __init__(self, x):
                self.x = x
                self.y = x+1

        def __str__(self):
                return "(%d %d)"%(self.x, self.y)

class B(A):
        def addZ(self):
                self.z = self.x + self.y

        def __str__(self):
                return "%d"%self.z

def main():
        b = B(23)
        b.addZ()
        print(b)

main()
```

8. Here is a program that keeps track of people and their favorite colors. Sadly, it doesn't work. Even though I make x represent person "Anna" and set x's favorite color to "blue", it still prints "Anna's favorite color is purple." What is wrong? Does it just like purple??

```
class Foo:
        def __init__(self, name):
                self.name = name
                self.favoriteColor = "purple"

        def changeFavoriteColor(self, fave):
                favoriteColor = fave

        def __str__(self):
                return "%s's favorite color is %s."%(self.name, self.favoriteColor)

        def main():
                x = Foo("Anna")
                x.changeFavoriteColor("blue")
                print(x)

        main()
```

9. Write a program that prints "I am thinking of a number between 1 and 100", and then saves a random number. The program should go into a loop that over and over asks the user for a number and says if that number is too high or too low. The program finally ends when the user guesses the program's number. I hate this game.

10. Write a program that will print the first 100 numbers whose digits sum to 10. Here's a hint: we can find the digits of x by repeatedly taking x%10 as one of the digits, then replacing x by x//10. For example, if we start with x=123, x%10 is 3 and x///10 is 12. Our first digit is 3 and we replace x by 12. Now x%10 is 2 and x//10 is 1, so our next digit is 2 and we replace x by 1. Since x is now less than 10, our last digit is x itself, which is 1. The digits of the original x are 3, 2, 1 and they sum to 6.

11. I have a file named "data.txt" that is organized like this:

```
#Lines that start with a hashtag are comments and
#should be ignored
23      45      16
33      24      33
14      44      26
29      43      39
```

I don't know how many lines are in the file. Every line with numbers has exactly three numbers separated by white space.
Write a function

readFile( )

that opens up the file and reads the first data column into list X, the second column into list Y and the third column into list Z, then returns X, Y, Z. For example, with the data shown above this will return
[23, 33, 14, 29], [45, 24, 44, 43], [16, 33, 26, 39]

12. Suppose list L is sorted in increasing order, such as L = [2, 3, 6, 7, 13, 19, 21]. Write a recursive function Find(x, L, low, high) that returns True if element x is in the portion of L between indices low and high, and False if it isn't. For example, with the list above Find(19, L, 0, 6) returns True and Find(15, L, 0, 6) returns False. You might want to start by comparing x to the element at index mid=(low+high)//2.

13. Write a program that will ask you for a file name. The program should then modify that file by adding the line "bob was here" at the top.

14. Write a class Animal whose constructor wants the name of the animal, such as “Donald Duck” or “Bessie the Cow”. Give the class an __str__( ) method that returns the animal’s name. Then write three subclasses of Animal: Cat, Dog, and Duck. The Cat __str__( ) method should return “meow”; the Dog method “woof” and the Duck method “quack”. The following program should then print “woof meow meow quack”:

```
def main( ):
        x = Dog( “Lassie”)
        y = Cat( “Sylvester” )
        z = Cat( “Cat in the Hat” )
        w = Duck( “Donald” )
        print(x, y, z, w)
```

15. Write a program that reads one of our dictionary files, such as “words2.txt” from the Anagram lab. Your program should print it out in the following order: first all of the 1-letter words (like “a” and “I”), then all of the 2-letter words, then the 3-letter ones, and so forth.