1. What will the following program print?

```python
def h(n):
    n = n+1
    for i in range(1, n):
        print("*", end="")
    print()

def g(n):
    for i in range(0, n):
        h(i)

def f(n):
    g(2*n+1)

def main():
    f(2)

main()
```

**Answer:**

*  
**  
***  
****
2. Here is a program with a recursive function foo. What will this program print?

```python
def foo(L):
    if len(L) == 0:
        return 1
    else:
        return 2*foo(L[1:])

def main():
    n = foo([7, 4, 2, 4, 9, 5, 6])
    print(n)

main()

Answer: 128 = 2^{\text{len(foo)}}
```
3. Here is a program that uses 3 related classes.
   a) What is printed by the line in main() print(T) ? \textbf{Answer: growl}
   b) What is printed by the line in main() print(L) ? \textbf{Answer: roar}

   class Cat:
   def \_\_init\_\_(self, name):
       self.name = name
       self.voice = "meow"

   def \_\_str\_\_(self):
       return self.voice

   class Tiger(Cat):
       def \_\_init\_\_(self, name):
           Cat.\_\_init\_\_(self, name)
           self.voice = "growl"

   class Lion(Cat):
       def \_\_init\_\_(self, name):
           Cat.\_\_init\_\_(self, name)

       def \_\_str\_\_(self):
           return "roar"

   def main():
       T=Tiger("Tony")
       print(T)
       L = Lion("Cowardly")
       print(L)

   main()
4. Here is a program that is supposed to store data in a list. It allows me to enter a bunch of numbers, but when it prints the list on the last line in main(), it only prints [].
   a) Explain in English what is wrong.
   b) Change the program (just cross out and write over it; you don’t need to rewrite all of the code) so that it will print the data you enter.

```python
def AddStuffToList(L):
    L = []
    done = False
    while not done:
        x = input("Enter a number, or a blank line to exit: ")
        if x == "":
            done = True
        else:
            L.append(eval(x))

def main():
    myList = []
    AddStuffToList(myList)
    print( myList )

main()
```

**Answers:**

   a) The line `L = []` in `AddStuffToList` means that `L` in that function refers to a different list than `myList`. So `myList` never changes.
   b) Just cross out the line `L = []` in `AddStuffToList`
5. In the following program function addUp(L) is supposed to return the sum of the entries of L that come before the first negative entry. So if L is [2, 3, -10, 5, 1] as it is in this program, addUp(L) should return 2+3, or 5. Sadly, in this program it actually returns 2+3+5+1, or 11. Explain in English why it works this way. You do not need to rewrite the program.

```python
def addUp(L):
    # This adds together the entries of L that precede
    # the first negative entry
    sum = 0
    for i in L:
        if i < 0:
            done = True
        else:
            sum = sum + i
    return sum

def main( ):
    print( addUp([2, 3, -10, 5, 1]))

main()
```

**Answer:** Setting variable done to True doesn’t stop the for-loop from running through the entire list.
6. The following program makes Persons named “Harry” and “Ron” and sets “Ron” as the friend of “Harry”. But when I go to print the “Harry” Person it says he has no friends. Explain why.

class Person:
    def __init__(self, name):
        self.name = name
        self.friend = None

    def makeFriends(self, other):
        self.friend = other
        other.friend = self

    def __str__(self):
        if self.friend == None:
            return "%s has no friends" % self.name
        else:
            return "%s is a friend of %s" %(self.name, self.friend.name)

def main():
    Person("Harry").makeFriends(Person("Ron"))
    print( Person( "Harry" ))

main()

**Answer:** Each call to Person(“Harry”) constructs a new person. So print( Person("Harry")) refers to a different person named “Harry” than the line above.
7. Write a function `grid(n)` that prints out an `nxn` grid of numbers. If `n` is 3 it should print

```
1 2 3
2 3 1
3 1 2
```

If `n` is 4 it should print

```
1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3
```

and so forth. This should work for any positive value of `n`, starting with `n = 1`.

Answer:
```python
def grid(n):
    for i in range(1, n+1):
        j = i
        counter = 0
        while counter < n:
            print( j, end=""),
            counter += 1
        j += 1
    if j > n:
        j = 1
    print()
```
8. Write function `isSorted( L )` which takes as an argument a list of numbers and returns True if the list is in non-decreasing order (each element is at least as large as the previous element), and False otherwise. For example, `isSorted( [2, 3, 3, 4] )` returns True while `isSorted( [2, 3, 2, 4] )` returns False.

   **Answer:**
   ```python
def isSorted( L ):
    if L == []:
        return True
    else:
        x = L[0]
        for y in L:
            if y < x:
                return False
            else:
                x = y
        return True
```
9. Write a program that serves as an inventory program for a grocery store. The program has two phases. In the first phase it allows you to repeatedly enter the names of items (strings). If the name is anything but the empty string it then asks you for the bar code for the item (another string) and how many you have. If you enter the same item name twice, the data from the second time overwrites the first. This phase ends when you give an empty string for the item. In the second phase, if you enter a barcode it prints the name and quantity of the item. If you enter a faulty barcode it says “I don’t recognize that code.” This phase ends on an empty barcode. Here is a sample run of this program:

First phase:

item: apples
barcode: 12345
quantity: 345

item: bananas
barcode: 22334
quantity: 48

item: apples
barcode: 12345
quantity: 325

item: lemons
barcode: 54321
quantity: 36

item: <blank line>

Second phase:

barcode: 54321
We have 36 lemons.

barcode: 12345
We have 325 apples.

barcode: <blank line>

Answer: see next page
def main():
    Inventory = {}
    done = False
    print( "First phase: " )
    while not done:
        item = input( "   item: " )
        if item == "":
            done = True
        else:
            barcode = input( "   barcode: " )
            quantity = eval(input( "   quantity: " ))
            Inventory[barcode] = (item, quantity)

    done = False
    print( "Second phase: " )
    while not done:
        barcode = input( "   barcode: " )
        if barcode == "":
            done = True
        else:
            if barcode in Inventory.keys():
                name, quantity = Inventory[barcode]
                print( "We have %d of %s." % (quantity, name))
            else:
                print( "I don't recognize that barcode." )

main()
10. Write a recursive function maximum(L) that returns the largest element in a list of numbers. For example, maximum( [2, 6, 4, 5, 9, 1]) is 9. You can assume that the length of L is at least one, so there should be a largest element. Note that I am looking for a recursive function for this, not a loop.

Answer:

def maximum(L):
    if len(L) == 1:
        return L[0]
    else:
        m = maximum(L[1:])
        if L[0] > m:
            return L[0]
        else:
            return m
Please write and sign the Honor Pledge when you have finished the exam.