Nesting Loops

Computer languages, including python, allow to place the code of one loop inside the code of another loop. The final arrangement is called “nested” loops. For the case of one loop inside of another, we can distinguish one inner and one outer loop. The "inner" loop is executed one time in its entirely for every time the outer loop executes. The inner loop is a “hard” worker while the outer loop is a “lazy” or “slower” worker.

Nesting is a powerful tool when you want the computer to "process" in "two or more dimensions".

The following example shows two nested for loops used to calculate and write out the product of two integers with small number of iterations:

```
for I in range(1,4):          # increment is 1
      for J in range(1,4):    # increment is 1
            prod = I * J
            print('%d * %d = %d'%(I,J,prod)
```

The resulting output:

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

Note that the inner for loop executes completely before the index variable of the outer for loop is incremented. In this code the resulting multiplications are not stored as before. The code uses only scalar variables.

If for-loops are nested, they should have independent loop control variables. If they have the same control variable, then the inner loop will change the value of the loop index that the outer loop just set and the program flow will become very difficult to follow.

Examples, avoid the following:

```
for I in range(1,6):
  for I in range(1,6):
    print('%d' %(I),end='')
  print()
```

Could you explain the behavior of the loop above?

Avoid also changing the control variable of the for loop, within the loop:

```
for kk in range(1,10):
  print('%d'%kk,end='')
  kk=kk+1   # this statement is wrong
  print('%d'%kk,end='')
```

Could you explain the behavior of the loop above?

Six (6) Nested Loop Exercises

Write one independent program to print each pattern below. Use nested for loops and print function to generate the patterns. The print must print ONE character or digit at a time. Optimize the number of counters and auxiliary variables. Submit the code and its output.

(1)

12345	
2345	
345	
45	
5	

(2)

1	
12	
123	
1234	
12345	

(3)

5	
44	
333	
2222	
11111	

(4)

55555	
4444	
333	
22	
1	

(5)

***** **** *** ** *	

(6) Optional

Compute total sum: 5+5+5+5+5+ 4+4+4+4+4+ 3+3+3+3+3+ 2+2+2+2+2+ 1+1+1+1+1+	