n = 3
while n > 0:
    if (n == 5):
        n = -99
        print(n)
    n = n + 1

What does this code print?

<table>
<thead>
<tr>
<th>A. 3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B. 3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C. 3</th>
<th>4</th>
<th>-99</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D. 3</th>
<th>4</th>
<th>5</th>
<th>-99</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Answer: C
With each iteration n will be
3
4
5 (which triggers if statement), then -99

so this code will print
3
4
-99
A valid password is one that is length 5 and starts with "xy". A valid password should terminate the loop. Which of these implements that specification? Note, the input function prints its argument as a prompt and returns whatever the user types as a string (after the user hits enter).

A. while True:
   s = input("Enter a password: ")
   if len(s) == 5 and s[:2] == 'xy':
       break
B. s = input("Enter a password: ")
   while len(s) == 5 and s[:2] == 'xy':
       s = input ("Enter a password: ")
C. Both A & B are correct
D. Neither A or B are correct

Answer: A
A is correct, the loop will continue until s satisfies the specifications. B is incorrect as the loop will continue while s meets the specification, not terminate.
a = 0
i = 0
while i < 10:
    a = a + 1

Will this loop terminate, be guaranteed to be an infinite loop or will it depend?

A. Terminate or not execute
B. Infinite loop
C. Depends
D. Syntax error

Answer: B
This is loop will be an infinite loop because the loop body does not change i, which remains < 10, and thus the loop conditional remains True forever.