

hw0506：

最初的輸入值可以分成兩個情況來解析：

1. $i \geq N$

依此條件， $i < N$ 的判斷式會回傳 0(False)，又由於 AND 判斷式只要其中一個不為 1(True)，便會直接回傳 0，因此 $i < N$ 以後只剩括號外的 printf 會被執行。又因 printf 的回傳值是成功輸出的字元數，而不論 i 值為何，`\n` 會使回傳值必 ≥ 1 (判斷上等價於 1)，因此待 printf 執行輸出 i 值後，函式回傳 `0||1 = 1`。

2. $i < N$

依此條件， $i < N$ 的判斷式會回傳 1(True)，接著執行 printf 輸出 i 值，依上述討論可得回傳值恆等價於 1(True)。當執行到 `!p(i+1,n)` 時，程式會先去執行呼叫函式內的程式，因此將會重複「 $i < N$ 回傳 1，printf 輸出 i 並回傳等價 1」的程序，直到當某次呼叫函式時 $i = N$ ，便會執行 1. 所描述的情況，最終回傳 1(True)。此時在此上一層的 p 函式，便會接收到 `!1 = 0(False)`，使整個 AND 判斷式回傳 0(False)，再與回傳值恆等價於 1(True) 的 printf 進行 OR 判斷並執行 printf 輸出，最終回傳 1(True)，依此規律持續回傳至最初的函式，使最後的回傳值一樣為 1(True)。

依照上列兩情況的討論，可得輸出結果：

1. 若 $i \geq N$ ，可輸出 i 值，並回傳 1(True)。

2. 若 $i < N$ ，可得 $i \quad i+1 \quad i+2 \quad \dots \quad N-1 \quad N \quad N-1 \quad \dots \quad i+2 \quad i+1 \quad i$
最終回傳 1(True)。