當函式p被呼叫的時候，傳入i跟N兩個初始值

情況一:i<N

進到return 程式由左邊執行到右邊

先得到i<N 成立 來到printf 會把i印出來且printf回傳值等於輸出的字元數大於0 成立 接著到 !p(i+1,N)，會呼叫下一個p函式。

進到下一個p函式後又再次檢查i<N成立，接著printf，但這次printf的結果是原本的i+1，printf後又呼叫了下一個p函式，以此類推，直到呼叫到i==N時 ，不符合i<N，對於and運算符號來說不管後面的結果如何那結果都會是0，因此就不做後面的printf跟呼叫下一個p了。

也就是說這一段程式碼會printf i,i+1,i+2直到最後一個數值i=N-1被printf出來。

接著到了or運算符後面，從剛剛結束的i==N開始印出，印出i=N後，因為printf的輸出字符數>0->回傳值>0 整個p的回傳值為1，!p(i+1,N) 也就是0。

後面碰到or運算符，因為前面是0，所以需要判斷後面，

也就是說會printf i=N-1，重複上述步驟一路往回推把之前沒做完的p函式做完，也就是說這一段code會從i==N開始printf到最後一個i==i. 最後的return值會因為or運算符後的printf輸出字元數>1->回傳值>1，所以最後的回傳值在or運算下等於1。

並在最後的畫面呈現出從i一路印到N-1再從N一路印到i的畫面

情況2:i>=N:

由於i不會小於N,所以前面and運算符的printf不會運作,只會做一次or運算符後面得printf,也就是只會印出i本身。