

6. Bonus: Code Explanation

原題目:

```
int p(int i, int N) {  
    return (i < N && printf("%d\n", i) && !p(i + 1, N)) || printf("%d\n", i);  
}
```

因為C語言的邏輯 AND 跟 OR 採用的是「短路式求值」([請看維基百科解釋](#))，所以連續的 AND 判斷會做到其中有人求出 `False` 為止。

而 OR 則是相反，做到求出 `True` 就停止！

`printf` 因為回傳的是字元數量，format string 裡面至少會有 `\n`，所以一定會回傳非零！

所以這支程式會不斷累加且同時輸出到 `N-1`，此時 `i < N` 會失敗，跳到後側的 `printf` 輸出 `N` 回傳非零，然後回到 call stack 的上一層，因為 OR 還沒做到 `True`，所以會再次執行這個 call stack frame 後側的 `printf`，輸出這個 call stack frame 的 `i` 後回傳非零(也就退到上個 call stack frame)。

上述的行為會導致程式從輸出 `i` 到 `N-1` 然後輸出 `N` 到 `i`。(這裡的區間描述皆為inclusive)