# 解題想法

假設兩個未知數m、n。 此函數定義: A(m,n) = 當m = 0，n + 1; 當m > 0 且 n = 0，A(m – 1, 1); 當m > 0 且 n > 0，A(m – 1, A(m, n - 1));其遞迴的寫法為  
int ack(int m,int n)

{

if(m == 0)

{

return n+1;

}

else if (n ==0)

{

return ack(m-1,1);

}

else return ack(m-1,ack(m,n-1));

}

# 演算法設計與實作

int main()

{

int a,b;

cin >> a>>b;

cout << ack(a,b) << endl;

return 0;

}

## 效能分析

空間複雜度：

 當 m=0 時，空間複雜度是 O(1)。

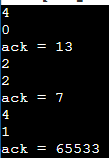
 當 m>0 且 n=0 時，空間複雜度是 O(m。

 當 m>0 且 n>0時，空間複雜度是超多項式的，

時間複雜度:

* 當 m = 0 時，函數的時間複雜度是O(1)，因為只有一個基本運算。
* 當 n = 0 且 m > 0 時，函數的時間複雜度是O(n)，因為它遞歸呼叫自身n次。
* 當 m > 0 且 n > 0 時，函數的時間複雜度非常高

## 測試與過程



當 m=4且 n=0 時：

A(4,0)=A(3,1)

接下來計算 A(3,1)：

A(3,1)=A(2,A(3,0))

而 A(3,0) 為：

A(3,0)=A(2,1)

繼續計算 A(2,1)：

A(2,1)=A(1,A(2,0))

而 A(2,0 為：

A(2,0)=A(1,1)

繼續計算 A(1,1)：

A(1,1)=A(0,A(1,0))

而 A(1,0) 為：

A(1,0)=A(0,1)

最後 A(0,1) 為：

A(0,1)=1+1=2

所以：

A(1,0)=A(0,1)=2

A(1,1)=A(0,A(1,0))=A(0,2)=2+1=3

A(2,0)=A(1,1)=3

A(2,1)=A(1,A(2,0))=A(1,3)

接下來計算 A(1,3)：

A(1,3)=A(0,A(1,2))

A(1,2)=A(0,A(1,1))=A(0,3)=4

所以：

A(1,3)=A(0,4)=5

因此：

A(2,1)=A(1,3)=5A(2, 1) = A(1, 3) = 5

A(3,1)=A(2,A(3,0))=A(2,5)

接下來計算 A(2,5)A(2, 5)A(2,5)：

A(2,5)=A(1,A(2,4))

然後一層一層推演下去，我們最終可以得到：

所以：

A(2,5)=A(1,11)=A(0,A(1,10))=12

因此：

A(3,1)=13

最終：

A(4,0)=A(3,1)=13

這樣，我們得到 A(4,0)=13