

Problem1 Ackerman function 程式說明

```
int ackerman(int m,int n){//這是用遞迴的寫法來實現Ackermann函數
    if (m == 0) { //m=0時,回傳n+1
        return n + 1;
    }
    else if (n == 0) { //n=0時,回傳ackerman(m - 1, 1)
        return ackerman(m - 1, 1);
    }
    else { //以上條件都不成立,回傳ackerman(m - 1, ackerman(m, n - 1))
        return ackerman(m - 1, ackerman(m, n - 1));
    }
}
```

這是使用遞迴的方式來實現 **Ackerman function**

m=0 時回傳 n+1

n=0 時,回傳 Ackerman(m-1,1)

假如以上條件都不成立

則回傳 **Ackerman(m-1, Ackerman(m,n-1))**

```
int ackermannnonrecursive(int m,int n){//這是用非遞迴的寫法來實現Ackermann函數
    while (m!=0) {
        if (m > 0 && n == 0) { //當m > 0 和 n == 0,就要處理a(m-1,1)
            n = 1;
            m -= 1;
        }
        else if (m > 0 && n > 0) { //當m > 0 和 n > 0,就要處理a(m-1,a(m,n-1))
            n -= 1;
            n = ackermannnonrecursive(m, n); // 遞迴處理n值
            m -= 1;
        }
    }
    return n + 1; //當m=0回傳n+1
}
```

這次用非遞迴的方式來實現 **Ackerman function**

當 $m > 0$ 且 $n = 0$ 時

就要處理遞迴裡的 **Ackerman(m-1,1)**

當 $m > 0$ 且 $n > 0$ 時

就要處理遞迴裡的 **Ackerman(m-1, Ackerman(m,n-1))**

```
int main() {  
    int m, n;  
    int a = 0;  
    int b = 0;  
    cout << "輸入兩個數字Enter two numbers:" ;  
    cin >> m >> n;  
    a = ackerman(m, n);  
    b = ackermannnonrecursive(m, n);  
    cout << "遞迴版本:" << a << endl;  
    cout << "非遞迴版本:" << b << endl;  
    return 0;  
}
```

主程式的部分就是請使用著輸入兩個數,也就是 **m** 和 **n** 值

經過程式的計算就會產生 **Ackerman function** 的值。