# problem 1

## Problem 1:

Ackermann's function $A(m,n)$ is defined as follows:

$$A(m,n) = \begin{cases} n+1 & \text{, if } m = 0 \\ A(m-1, 1) & \text{, if } n = 0 \\ A(m-1, A(m,n-1)) & \text{, otherwise} \end{cases}$$

This function is studied because it grows very fast for small values of $m$ and $n$. Write a recursive function for computing this function. Then write a nonrecursive algorithm for computing Ackermann's function.

### 完整程式碼:

```
/*  1017 hw1 資工二乙 41143264 楊育哲
    A(m, n) 遞迴 (阿克曼函數)
*/
#include <iostream>
using namespace std;

int recA(int m, int n){//以if-else實作遞迴版本
    if(m==0) return n+1;
    else if(n==0) return recA(m-1, 1);
    else return recA(m-1, recA(m, n-1));
}

int nonrecA(int m, int n){//以stack實作蝶帶版本
    int stackOfM[100]={0}, current=0;
    stackOfM[0] = m;
    while(current>=0){
        m = stackOfM[current--];
        if(m==0) n++;
        else if(n==0){
            n=1;
            stackOfM[++current]=m-1;
        }else{
            stackOfM[++current]=m-1;
            stackOfM[++current]=m;
            n--;
        }
    }
    return n;
}

int main(){
    cout<<recA(1, 1)<<" "<<nonrecA(1, 1);
```

```
    return 0;
}
```

## recursive function:

```
int recA(int m, int n){//以if-else實作遞迴版本
    if(m==0) return n+1;
    else if(n==0) return recA(m-1, 1);
    else return recA(m-1, recA(m, n-1));
}
```

1. 解題說明:

   依題目要求，將敘述以if-else形式呈現。

2. 校能分析:

   - S(P)=4*(n+1), 4 words(m, n, 回傳值, 回傳位址), n+1次遞迴
   - T(P)=2*(n+1), 2 steps(if-else, return), n+1次遞迴
   - f(n)=O(n)

3. 測試與驗證

   測試: cout<<recA(1, 1);// ←’3’

   驗證: recA(1, 1)=recA(1-1, recA(1, 0))=recA(0, recA(0, 1))=recA(0, 2)=3

## nonrecursive function:

```
int nonrecA(int m, int n){//以stack實作蝶帶版本
    int stackOfM[100]={0}, current=0;
    stackOfM[0] = m;
    while(current>=0){
        m = stackOfM[current--];
        if(m==0) n++;
        else if(n==0){
            n=1;
            stackOfM[++current]=m-1;
        }else{
            stackOfM[++current]=m-1;
            stackOfM[++current]=m;
            n--;
        }
    }
    return n;
}
```

1. 解題說明:

   以stack替代遞迴，其中if-else中做的判斷及行為都等 同於recA中的if-else式。

2. 效能分析:

   - S(P)=4+0, 4 words(m, n, stackOfM, current)
   - 4+5*(n+1)≤T(P)≤4+6*(n+1), 4 for {宣告*2, while, return},
   - f(n)=O(n)

3. 測試與驗證:

   測試: cout<<nonrecA(1, 1);// ←’3’

   驗證: 分析nonrecA(1, 1):

   (1) m=1, n=1, stackOfM={1, 0, …}, current=0

   (2) m=1, n=0, stackOfM={1-1, 1, …}, current=1

   (3) m=0, n=1, stackOfM={0, 1-1, …}, current=1

   (4) m=0, n=2, stackOfM={0, 0, …}, current=0

   (5) m=0, n=3, stackOfM={0, 0, …}, current=-1

   最後回傳n，即3。