

Program Exercise #1 (1)

- Demo date: [Sep. 18](#) (Tue.) 23:59 p.m.
- TA: 詹祐維、黃定銳
- Reading Assignment: Chapter 2.1, 2.3-2.7
- Software: Dev-C++
- Submission:
 - Filename format:
學號_programNo.c
例如: M06455505_program1-1.c
 - In-class demo ([Sep. 19 Wed. 9:10-12:00 a.m.](#))
 - iLearn 2.0 submission
 - * 用自己的學號建立資料夾，並將 source code 放入資料夾，壓縮上傳 iLearn 2.0

Program Exercise #1 (2)

- Grading:
 - Correctness 50%
 - Program structure 20%
 - Comments 10%
 - Header block 5%
 - Variable dictionary 10%
 - Procedures and functions 5%
- Special notice:
 - 請勿抄襲別人程式(助教會當場進行測問、判定)，或是遲交作業，否則一律 0分計算
 - 請一律使用 C 語言來撰寫程式，且必須保證你的程式能夠再 Dev-C++ 軟體上成功編譯與執行，使用其他程式語言一律不予計分
 - 請依照題目給的輸入格式，否則不計分
 - 本次作業有三個子題目，個別配分分別佔 30%, 30%, 40%

Program Exercise #1 (3)

- Problem 1-1 (30%):

P.17, #8: Write both a **recursive** and an **iterative** C function to compute Fibonacci numbers.

Definition:

$$F(i) = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F(n-1)+F(n-2), & \text{if } n > 1 \end{cases}$$

Basic requirements:

輸入: 要求使用者輸入n值，其中n要至少為2，否則印出 **Error!**，並且要求使用者重新輸入

輸出: 在螢幕上顯示F(n)的值，並要求使用者輸入下一個n，直到使用者輸入非數字字元才結束程式

Example:

(1) Input:

%> n = 3

Output:

%> F(3) (iter.) = 2

%> F(3) (recur.) = 2

(2) Input:

%> n = 1

Output:

%> Error!

Program Exercise #1 (4)

- Problem 1-2 (30%):

P.17, #:10: Write a **recursive** function to compute Ackerman's function $A(m, n)$.

Definition:

$$A(i,j) = \begin{cases} j + 1, & i = 0 \\ A(i-1,1), & i > 0 \text{ and } j = 0 \\ A(i-1, A(i, j-1)), & i, j > 0 \end{cases}$$

Basic requirements:

輸入: 要求使用者輸入*i, j*值,

輸出: 在螢幕上顯示 $A(i,j)$ 的值, 並要求使用者輸入下一個*i, j*, 直到使用者輸入非數字字元才結束程式

Example:

Input:

%> i = 2

%> j = 2

Output:

%> $A(2,2) = 7$

Program Exercise #1 (5)

- Problems 1-3 (40%):

Write a **recursive** function to find the maximum value in a series of number.

Definition:

- (1) define a function: `int max(int x, int y)`, and return the maximum value from these two numbers.
- (2) recursively call the function: `int arraymax(int a[], int n)` to get the maximum value from the given input array A.

Basic requirements:

Input: given a series of number: A[8 6 1 17 16 23 9 9], separated by space.

Output: return the maximum number in the array A.

Example:

Input:

%> Input array: 8 6 1 17 16 23 9 9

Output:

%> 23