

Day3: 函數與遞迴 07/10

◇ 自訂函數

```
double dist(double x1, double y1, double x2, double y2)
{
    return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
}
```

◇ 自訂結構 typedef struct

計算兩點歐幾里得距離的函數

```
typedef struct { double x,y;} Point;
double dist(Point a, Point b)
{
    return hypot(a.x-b.x, a.y-b.y);
}
```

```
double hypot(
)
{
    return _____;
}
```

◇ 計算組合數

- 編寫函數，參數是兩個非負整數 n 和 m
- 傳回組合數結果
- 其中， $m \leq n \leq 25m$

例如： $n=25$, $m=12$ 時，答案為 5200300

◇ 編寫求 $n!$ 的函數

```
long long factorial(int n){
    long long m=_____;
    for (int i=1; i<=n; i++)
        m = _____;
    return m;
}
```

◇ $C(n, m) = n! / ((n - m)! \times m!)$

```
long long factorial(int n){
    long long m=1;
    for (int i=1; i<=n; i++)
        m *= i;
    return m;
}
```

```
long long C(int n, int m)
{
    return factorial(____) / (factorial(____)*factorial(____));
}
```

◇ 別忘了測試

- n=25, m=12 時，答案為 5200300。
- n=21, m=1 答案為？
- 即使最終答案在我們選擇的資料型別內，計算的中間結果仍可能溢位。

◇ 對複雜的運算式進行簡化

減少計算量(約分)、避免中間結果溢出

```
long long C(int n, int m){
    if (m < n-m) m = n-m;
    long long ans = 1;
    for (int i=____; i<=____; i++) ans *= i;
    for (int i =____; i<=____; i++) ans /= i;
    return ans;
}
```

◇ 質數判定

- 編寫一函數，參數是一個正整數 n 。如果 n 是質數，返回 1，否則返回 0。
- 質數定義：任意大於 1 的正整數 n ，若只能被 1 和自身整除，則 n 為質數

is_prime (bug)

```
int is_prime(int n)
{
    for (int i=2; i*i <=n ; i++)
        if (n%i == 0) return 0;
    return 1;
}
```

思考 why：

$n = 1$

n 很大

is_prime (正確)

```
int is_prime(int n)
{
    if (n<=1) return 0;
    int m = floor(sqrt(n) + 0.5);
    for (int i=2; i<=m ; i++)
        if (n%i == 0) return 0;
    return 1;
}
```

思考 why：

m 為何不為 $\text{sqrt}(n)$ 即可？

◇ 兩數交換 swap() - 三變數交換

思考以下程式執行結果：

```
#include <stdio.h>
void swap(int a, int b){
    int t=a; a=b; b=t;
}
int main(){
    int a = 3, b=4;
    swap(a,b);
    printf("%d %d\n", a,b);
    return 0;
}
```

呼叫函數時的參數為實際參數 (Argument)。
定義函數時的參數為形式參數 (Parameter)。
函數內部的變數為區域變數 (local variable)。
main() 函數以外的變數為全域變數 (global variable)

◇ swap() 以指標當作參數

思考以下程式執行結果：

```
#include <stdio.h>
void swap(int* a, int* b){
    int t=*a; *a=*b; *b=t;
}
int main(){
    int a = 3, b=4;
    swap(&a, &b);
    printf("%d %d\n", a,b);
    return 0;
}
```

```
int* a;
```

整數指標的宣告

*a 為 a 指向的變數值

&a 為 a 的位址

◇ 陣列作為參數與返回值

計算陣列元素和 (傳陣列起始位址與陣列大小)

```
int sum(int* a, int n)
{
    int ans = 0;
    for (int i = 0; i<n; i++)
        ans +=a[i];
    return ans;
}
```

```
int main()
{
    int a[] = {1,2,3,4};
    printf(" %d\n ", sum(a,4));
    return 0;
}
```

若改計算

sum(a+1,3)

答案為何?

◇ 遞迴函數 Recursive function

$f(n) = n! = n * f(n-1)$

○ $f(0)=1$

○ $f(n) = f(n-1)*n$

```
#include <stdio.h>
int f(int n){
    if (n==0) return 1;
    else return f(n-1)*n;
}
```

問：

$f(3) = ?$

$f(3)$ 會如何呼叫堆疊

2. 給定一陣列 $a[10]=\{ 1, 3, 9, 2, 5, 8, 4, 9, 6, 7 \}$ ，i.e., $a[0]=1, a[1]=3, \dots, a[8]=6, a[9]=7$ ，以 $f(a, 10)$ 呼叫執行右側函式後，回傳值為何？

- (A) 1
- (B) 2
- (C) 7
- (D) 9

```
int f (int a[], int n) {
    int index = 0;
    for (int i=1; i<=n-1; i=i+1) {
        if (a[i] >= a[index]) {
            index = i;
        }
    }
    return index;
}
```

3. 給定一整數陣列 $a[0]$ 、 $a[1]$ 、 \dots 、 $a[99]$ 且 $a[k]=3k+1$ ，以 $value=100$ 呼叫以下兩函式，假設函式 $f1$ 及 $f2$ 之 **while** 迴圈主體分別執行 $n1$ 與 $n2$ 次 (i.e, 計算 **if** 敘述執行次數，不包含 **else if** 敘述)，請問 $n1$ 與 $n2$ 之值為何？ 註： $(low + high)/2$ 只取整數部分。

```
int f1(int a[], int value) {
    int r_value = -1;
    int i = 0;
    while (i < 100) {
        if (a[i] == value) {
            r_value = i;
            break;
        }
        i = i + 1;
    }
    return r_value;
}
```

```
int f2(int a[], int value) {
    int r_value = -1;
    int low = 0, high = 99;
    int mid;
    while (low <= high) {
        mid = (low + high)/2;
        if (a[mid] == value) {
            r_value = mid;
            break;
        }
        else if (a[mid] < value) {
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return r_value;
}
```

5. 函數 f 定義如下，如果呼叫 $f(1000)$ ，指令 $sum=sum+i$ 被執行的次數最接近下列何者？

- (A) 1000
- (B) 3000
- (C) 5000
- (D) 10000

```
int f (int n) {
    int sum=0;
    if (n<2) {
        return 0;
    }
    for (int i=1; i<=n; i=i+1) {
        sum = sum + i;
    }
    sum = sum + f(2*n/3);
    return sum;
}
```

7. 請問以 $a(13,15)$ 呼叫右側 $a()$ 函式，函式執行完後其回傳值為何？

- (A) 90
- (B) 103
- (C) 93
- (D) 60

```
int a(int n, int m) {
    if (n < 10) {
        if (m < 10) {
            return n + m ;
        }
        else {
            return a(n, m-2) + m ;
        }
    }
    else {
        return a(n-1, m) + n ;
    }
}
```

10. 給定右側 $g()$ 函式， $g(13)$ 回傳值為何？

- (A) 16
- (B) 18
- (C) 19
- (D) 22

```
int g(int a) {
    if (a > 1) {
        return g(a - 2) + 3;
    }
    return a;
}
```

12. 給定右側函式 `f1()` 及 `f2()`。 `f1(1)` 運算過程中，以下敘述何者為錯？

- (A) 印出的數字最大的是 4
- (B) `f1` 一共被呼叫二次
- (C) `f2` 一共被呼叫三次
- (D) 數字 2 被印出兩次

```
void f1 (int m) {
    if (m > 3) {
        printf ("%d\n", m);
        return;
    }
    else {
        printf ("%d\n", m);
        f2(m+2);
        printf ("%d\n", m);
    }
}

void f2 (int n) {
    if (n > 3) {
        printf ("%d\n", n);
        return;
    }
    else {
        printf ("%d\n", n);
        f1(n-1);
        printf ("%d\n", n);
    }
}
```

15. 若以 `f(22)` 呼叫右側 `f()` 函式，總共會印出多少數字？

- (A) 16
- (B) 22
- (C) 11
- (D) 15

```
void f(int n) {
    printf ("%d\n", n);
    while (n != 1) {
        if ((n%2)==1) {
            n = 3*n + 1;
        }
        else {
            n = n / 2;
        }
        printf ("%d\n", n);
    }
}
```


14. 右側程式輸出為何？

- (A) bar: 6
bar: 1
bar: 8
- (B) bar: 6
foo: 1
bar: 3
- (C) bar: 1
foo: 1
bar: 8
- (D) bar: 6
foo: 1
foo: 3

```
void foo (int i) {  
    if (i <= 5) {  
        printf ("foo: %d\n", i);  
    }  
    else {  
        bar(i - 10);  
    }  
}  
  
void bar (int i) {  
    if (i <= 10) {  
        printf ("bar: %d\n", i);  
    }  
    else {  
        foo(i - 5);  
    }  
}  
  
void main() {  
    foo(15106);  
    bar(3091);  
    foo(6693);  
}
```

20. 右側為一個計算 n 階層的函式，請問該如何修改才會得到正確的結果？

```
1. int fun (int n) {  
2.     int fac = 1;  
3.     if (n >= 0) {  
4.         fac = n * fun(n - 1);  
5.     }  
6.     return fac;  
7. }
```

- (A) 第 2 行，改為 `int fac = n;`
- (B) 第 3 行，改為 `if (n > 0) {`
- (C) 第 4 行，改為 `fac = n * fun(n+1);`
- (D) 第 4 行，改為 `fac = fac * fun(n-1);`

22. 右側 `f()` 函式執行後所回傳的值為何？

- (A) 1023
- (B) 1024
- (C) 2047
- (D) 2048

```
int f() {  
    int p = 2;  
    while (p < 2000) {  
        p = 2 * p;  
    }  
    return p;  
}
```

23. 右側 $f()$ 函式 (a), (b), (c) 處需分別填入哪些數字，方能使得 $f(4)$ 輸出 2468 的結果？

- (A) 1, 2, 1
- (B) 0, 1, 2
- (C) 0, 2, 1
- (D) 1, 1, 1

```
int f(int n) {
    int p = 0;
    int i = n;
    while (i >= (a) ) {
        p = 10 - (b) * i;
        printf ("%d", p);
        i = i - (c) ;
    }
}
```

24. 右側 $g(4)$ 函式呼叫執行後，回傳值為何？

- (A) 6
- (B) 11
- (C) 13
- (D) 14

```
int f (int n) {
    if (n > 3) {
        return 1;
    }
    else if (n == 2) {
        return (3 + f(n+1));
    }
    else {
        return (1 + f(n+1));
    }
}

int g(int n) {
    int j = 0;
    for (int i=1; i<=n-1; i=i+1) {
        j = j + f(i);
    }
    return j;
}
```

25. 右側 $Mystery()$ 函式 **else** 部分運算式應為何，才能使得 $Mystery(9)$ 的回傳值為 34。

- (A) $x + Mystery(x-1)$
- (B) $x * Mystery(x-1)$
- (C) $Mystery(x-2) + Mystery(x+2)$
- (D) $Mystery(x-2) + Mystery(x-1)$

```
int Mystery (int x) {
    if (x <= 1) {
        return x;
    }
    else {
        return _____ ;
    }
}
```

習題：UVa 100- $3n+1$

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=36