

《程式設計概要》

試題評析

今年的考試題型與去年類似，仍然出現連鎖題型。不過今年的題目較往年簡單，只要能熟悉一般的程式題型就能拿到高分。比較特殊的是第四題要求時間與空間複雜度，本題的答案與所使用的撰寫方式有關，可以使用迴圈與遞迴撰寫程式。若使用遞迴撰寫程式則時間複雜度的計算會比較複雜，建議使用迴圈撰寫。另外，今年延續往年可使用熟悉的語言作答，但題目舉例是以C語言為範例，所以建議使用C語言作答。今年的平均分數應會超過75分以上，用功的考生可得90分以上。

一、試寫一函式 (function) : `int SumOfDigits(int n)`。若 `n` 為正整數，則該函式傳回 `n` 以十進位表示法的所有位數中數字的總和。(30分)

例如：若 `int n=1236; int k=SumOfDigits(n)`；則 `k` 的值為 `1+2+3+6` 也就是等於 `12`。

若 `int n=58; int k= SumOfDigits(n)`；則 `k` 的值為 `5+8` 也就是等於 `13`。

答：

```
int SumOfDigits(int n){
    if(n<0) return(-1); /* 輸入參數有誤 */
    if(n==0) return(0);
    else return(SumOfDigits(n/10) + (n%10));
}
```

二、請利用第一題的 `SumOfDigits(n)` 函式，撰寫一程式解決以下問題：輸入兩個正整數 `A` 和 `B`，已知 `A` 不大於 `B`，找出所有大於等於 `A` 且小於等於 `B` 的正整數 `n` 中，可以被 `SumOfDigits(n)` 傳回值整除者。

例如：若 `n=100; int k=SumOfDigits(n)`；則 `k` 的值為 `1`，`k` 可以整除 `n`。

請撰寫一程式將所有符合上述條件的正整數一一在螢幕上印出，一行恰好印一個。(15分)

答：

```
void MultipleOfN(int A, int B){
    int i;

    if(A<=0 || B<=0 || A>B){ /* 輸入參數有誤 */
        printf("Input Error!!\n");
        return ;
    }
    else{
        for(i=A; i<=B; i++){
            if(i % SumOfDigits(i) == 0) printf("%d\n", i);
        }
    }
}
```

三、撰寫一程式解決以下問題：(25分)

輸入：三個正整數 `n`、`m` 及 `d`；已知 `n>2*d` 且 `m>2*d`。

輸出：在螢幕印出如下由星號(*)組成之 `n` 行 `m` 列每條邊厚度均為 `d` 之空心長方形圖案，所謂長方形某一條邊的厚度是指該邊的星號線條數。

例如：若 $n=10, m=8, d=2$ 則印出：

```
*****
*****
**      **
**      **
**      **
**      **
*****
*****
```

答：

```
void Star(int n, int m, int d){
    int i, j;

    if(n<=0 || m<=0 || d<=0 || n<=2*d || m<=2*d){ /* 輸入參數有誤 */
        printf(" Input Error\n");
        return ;
    }else{
        for(i=1; i<=m; i++){
            for(j=1; j<=n; j++){
                if(i<=d || j<=d || i>(m-d) || j>(n-d)) printf("*");
                else printf(" ");
            }
            printf("\n");
        }
    }
}
```

四、若把某數列的 i 個值紀錄為 $f(i)$ ，已知 $f(1)=1, f(2)=2, f(3)=3, f(4)=5$ ；且若 $i>4$ 則 $f(i)=3*f(i-1)+5*f(i-3)$ 。撰寫一函式 (function)：int fun(int n)。若輸入之 n 為正整數，則輸出 $f(n)$ 。並用Big-O符號分析執行該程式的時間及空間複雜度。(30分)

答：

```
int fun(int n){
    int i, f[n]; /* 空間複雜度: O(n), 使用n個元素的整數陣列 */

    if(n<=0) return(-1); /* 輸入參數有誤 */
    if(n<=3) return(n);
    if(n==4) return(5);

    f[2]=2; f[3]=3; f[4]=5;
    for(i=5; i<=n; i++){ /*時間複雜度: O(n), for迴圈與n成正比 */
        f[i]=3*f[i-1]+5*f[i-3];
    }
    return(f[n]);
}
```

```
}
```

時間複雜度: $O(n)$, for迴圈與 n 成正比

空間複雜度: $O(n)$, 使用 n 個元素的整數陣列

