

1. 若  $n$  為 10，則此程式的可變空間需求為 (6 分)

```
float rsum(float list[], int n)
{
    if (n) return rsum(list, n-1) + list[n-1];
    return 0;
}
```

2. 計算下列函數的總步驟數 (6 分)

```
void add( int a[], int b[], int c [], int m, int n)
{
    int i, j;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            c[i][j] = a[i][j] + b[j][j];
}
```

3. 在 `int list[8]` 中，`list[i]=i+3`，對於  $i=0\dots 7$ ；`&(list[6])` 為 `0x123456` (11 分)

(A) `*(list+5)` 為

(B) `(list+2)` 為

4. 後置轉中置 (12 分)

(A) `ab*5+`

(B) `ab/c-de*+ac*-`

5. 下列的敘述中，哪些為真(true)? (30 分)

(A) 由上而下(top-down)的方法是以程式最後目標為開端，並利用這個最後結果將程式分割成數個可以管裡的片段。

(B) 在分析階段會觀察程式需要哪些資料物件(data objects)與運算(operations)來處理這些資料物件。

(C) 資料物件的表示法與其相關演算法效率無關，所以其先後順序不重要。

(D) 驗證階段包含證明程式正確性、使用大量輸入資料來測試程式、移除錯誤。

(E) 堆疊(stack)為一後進先出(last-in-first-out)的有序串列。

(F) 堆疊(stack)常被用於作業系統所建構的工作排程。

6. 下列的敘述中，哪些為真(true)? (35 分)

(A) 演算法要滿足以下條件：輸入、輸出、明確性(definiteness)、無限性(infiniteness)、有效率性(effectiveness)。

(B) 抽象化資料型態(abstract data type)將物件及其運算規格，與物件實際表示方法與運算的實作方法分開。

(C) 與機器相關的時間和空間估計，亦即為複雜度理論。

(D) 效能分析(performance analysis)與量測(performance measurement)都可用來找出沒有效率程式碼片段。

(E) 空間複雜度(space complexity)是指完成程式所需要的記憶體大小。

(F)  $O(n \log n)$ 的時間複雜度高於  $O(n^2)$ 。

(G)  $O(2^n)$ 的時間複雜度高於  $O(n^{10})$ 。