

## 電腦與問題解決學習單 (二)

班級 \_\_\_\_\_ 座號 \_\_\_\_\_ 姓名 \_\_\_\_\_

### 一、資料結構與演算法

- PASCAL 語言之父威爾斯教授提到「程式 = \_\_\_\_\_ + \_\_\_\_\_」的概念。
- 程式中的 \_\_\_\_\_ 可以暫存我們輸入的資料，還可以隨時修改內容。
- \_\_\_\_\_ 的功用就像是箱子，讓程式可以找到要存取的資料。

### 二、變數的資料型態

- 程式語言常用的資料型態有
  1. \_\_\_\_\_ (包含長整數)：佔 \_\_\_\_\_ 或 \_\_\_\_\_ 位元組。
  2. \_\_\_\_\_：單精度佔 \_\_\_\_\_ 位元組、倍精度佔 \_\_\_\_\_ 位元組。
  3. \_\_\_\_\_：所佔位元組視字串長度而定。
  4. \_\_\_\_\_：佔 \_\_\_\_\_ 位元組。
  5. \_\_\_\_\_：佔 \_\_\_\_\_ 位元組。
- 在程式中適當應用資料結構，會有以下好處：
  1. 可讓演算法的設計變得 \_\_\_\_\_，使程式更容易 \_\_\_\_\_。
  2. 可減少大量 \_\_\_\_\_ 的撰寫，提升程式的 \_\_\_\_\_。
  3. 可減少 \_\_\_\_\_ 的浪費，讓電腦不會耗用太多的系統資源。
- 常見的資料結構類型，有 \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_ 等。
- **陣列(array)**是最常見、使用率最高的資料結構，它是由多個元素組成的集合，要存取陣列中的每個元素要指定該元素的 \_\_\_\_\_ (index value) 來存取。
- **堆疊(stack)**是一個串列，它的兩端分別稱為頂端(\_\_\_\_\_)和底端(\_\_\_\_\_)，資料的推入(\_\_\_\_\_)和彈出(\_\_\_\_\_)都在頂端執行，採用 \_\_\_\_\_ (last in / first out, \_\_\_\_\_)的處理方式。
- **佇列(queue)**和堆疊的性質相近，只是佇列加入資料的方式，就像排隊一樣。佇列資料是固定由尾端(\_\_\_\_\_)加入，讀取資料時則固定由頭端(\_\_\_\_\_)開始，最先進來的資料優先處理，屬於 \_\_\_\_\_ (first in /first out, \_\_\_\_\_)的處理方式。

- **樹狀結構(tree)**是一種\_\_\_\_\_式的資料表示方式，是指一個元素可以像樹枝般分成多個元素的結構。

