

38. 使用 recursion (遞迴) 的三大原則:

- ① 給分身的任務一定要比(別人給)我的任務「小」
- ② 一定要有 recursion base(s) (可直接做掉, 不用叫分身做)
- ③ 整個運作過程永遠都會觸底反彈。

Recursion function 的長相,

if . . .

else if . . .

else if . . .

17. ~~can~~ constructor 是 class 宣告時直接定義裡面變數的功能。

40. 前述 a 的儲存位置是在 call stack, b 則是在 keep
call stack 是呼叫堆疊, 是儲存有關正在執行的子程式的堆疊, 每一層遞迴都必須在呼叫堆疊上增加一條位址, 因此如果程式出現無限遞迴 (或只是過多的遞迴層次), 呼叫堆疊就會產生堆疊溢位。

keep 僅跟 pointer 有關, 用來存 pointer 指向的位置。

資料結構 C++ 篇:

1. sqrt 是透過 `<math.h>` 或 `<c.math>` 得到的函式, 能把數字開根號 ex: `num = sqrt(num);`
2. atoi 是透過 `<stdlib.h>` 得到的函式, 能把字串變數字 ex: `num = atoi(str);`
3. 如果把函式中會回傳的值直接更動使用似乎不行

ex: `function (int &a)`

`function (a+1);`

用在 `istream` 或 `ifstream`

4. C++ 開啟檔案可用 `fopen`, `open` 或 `ifstream` 直接用 `ifstream` constructor 叫出來
用在 FILE

5. `c_str()` 可以把 `string` 改成
`char` 指標

ex: `string file;`
`fopen(file.c_str(), "r");`

6. `ifstream` 和 `ofstream` 這兩個型別
分別可以把檔案中的東西
輸入和把東西輸出到檔案中
`fstream` 則兩者皆可。

7. `cin.getline` 和 `getline` 不一樣

8. C++ 的 `string` 也可像 `char`
陣列一樣指定第幾格。

9. `struct` 在宣告時裡面的變數
似乎不能直接設值。

10. `ifstream` 在放入函式時一定

要加& (給位址)! 不然編譯不會過!

11. string 轉 char 可用 `char a[50];`
`strcpy(a, str.c_str());`

12. 用指標指向陣列時如果要 delete 要這樣寫:

```
int *a = new int[20];
```

```
delete[] a;
```

13. 如果設了一個 struct, 算直接 ex 比較快 XD!

```
ex: struct point {  
    int int a;  
    int b;  
};
```

```
point x; x.a = 5; x.b = 5;
```

```
point y = x; 這是可行的!
```


14. 用vector的話如果一直用
push-back來增加元素很影響
性能

15. ~~vec~~ vec.erase(1, 8)
刪除的範圍是[1]~[7]

16. erase有overload, 使用時要
注意型別

17. vector如果用在函式的話要加
&&&

18. 3221225477 通常是陣列或
指針的東西為空導致

19. 用push-back()幫vector加
新元素時, ()裡面要有東西。

20. vector在用之前記得裡面

要先有東西(尤其是二維的)。

20. 編譯錯時可能是因為
用C的函式讀了C++函式產生

的值(新舊衝突)。

21. 使用物件導向(class)時,要盡可能
讓每個函式只^(class)做一件事(高內聚),
且盡量讓他們獨立(低耦合)以
方便維護。

5. data member (DM) / member function

(MF) 的呼叫權限:

private: 只有本 class 的 MF 可使用、呼叫

protected: 本 class 及本 class 的子孫 class 的 MF、還有同個 package 的 class

public: 任何 class 的 MF 可使用、呼叫

沒設 (package-scoped) (default):

放在同一目錄下的 class 系統默認值

1. 作為何人、子、由圖所指示