

112 年特種考試地方政府公務人員考試試題

等 別：四等考試

類 科：資訊處理

科 目：程式設計概要

陳凱老師

一、撰寫遞迴函式是重要程式設計技巧之一。

(一)請說明下列遞迴函式 findnum 目的為何？(5 分)

```
int items, price[100]; //全域變數

int findnum (int i, int num) {
    if (i < items) {
        if (num < price[i])
            num = price[i];
        num = findnum (i+1, num);
    }
    return num;
}
```

```
int main () {
    int i, num;

    // 讀入數量及所有正整數價錢
    scanf("%d", &items);
    for (i=0; i<items; i++)
        scanf ("%d", &price[i]);
    num = findnum (0, -1);
    printf ("The ... is %d\n", num);
    return 0;
}
```

(二)請將 findnum() 函式改寫成非遞迴的函式。(10 分)

(三)請說明下列非遞迴程式目的為何？(5 分)

```
void convert (long m) {
    long count=-1, n=0;
    while (0 < m) {
        n = n*10 + m%2;
        m = m / 2;
        count++;
    }
    while (0 < count) {
        printf ("%ld", n%10);
        n = n / 10;
        count--;
    }
    printf ("%ld", n%10);
    return;
}
```

```
int main() {
    long m;

    scanf("%ld",&m);
    printf(" The ... value is ");
    convert (m);
    printf ("\n");
    return 0;
}
```

(四)請將 convert() 函式改寫成以遞迴運算的函式，且函式內不可新增變數。(10 分)

《考題難易》★★★

《破題關鍵》：基本的迴圈與遞迴函式，細心思考推導即可。m 可以自行設定一個值(ex: 13)，從輸出來反推。

【擬答】

(一) findnum()：找出所有輸入數值的最大值。

(二)

```
int findnum(int i, int num){
    int max = num;
    for(int x = i; x < item; x++){
        if(price[x] > max){
            max = price[x];
        }
    }
    return max;
}
```

(三)將輸入的十進位數值轉換成二進位後輸出。

(四)

```
int convert(long m){
    if (m <= 0)
        return 0;
    else
        return 10 * convert2(m / 2) + (m % 2);
}
```

二、請說明下列 PHP 程式設計的觀念。

(一) Class 和 Interface 的差異為何？請從可否宣告屬性、可否實例化、可否有實作方法 3 個面相加以說明。(5 分)

(二)若前端網頁以 HTML 程式上傳一個檔案到後端，請以 PHP 寫出後端要處理的部分，包括檢查檔案是否上傳成功、檢查檔案是否存在（不可覆蓋）、將上傳的檔案搬移到指定位置。(15 分)

前端：

```
<form method="post" enctype="multipart/form-
data"action="upload.php">
    <input type="file" name="to be uploaded">
    <input type="submit" value="Upload">
</form>
```

後端：

```
<?php
```

...

```
?>
```

《考題難易》★★★★★

《破題關鍵》：第一小題屬於物件導向的基礎題型。若對 PHP 的物件導向不熟悉，可以從 Java 語言切入作答。但是第二小題須了解 PHP 的檔案處理相關函式，屬於冷門題目，同學可能不好作答。

【擬答】

(一) Class :

1. 可宣告屬性
2. 可實例化
3. 可實作方法

Interface

1. 可宣告屬性，但是宣告為 const
2. 不可實例化
3. 無法實作方法

(二)

<?php

```
if ($_FILES['to_be_uploaded']['error'] === UPLOAD_ERR_OK){
    if (file_exists('./' . $_FILES['to_be_uploaded']['name'])){
        echo("檔案已存在!");
    }
    else {
        $file = $_FILES['to_be_uploaded']['tmpName'];
        $dest = 'fileDir/' . $_FILES['to_be_uploaded']['name'];
        move_uploaded_file($file, $dest);
    }
}
?>
```



資訊處理榮耀上榜

110地特四等	110地特四等	111普考	111地特三等	110地特三等	110地特三等	111地特四等	110普考
台北市狀元	金門縣狀元	全國榜眼	金門縣榜眼	桃園市第四	花東區第四	台北市第八	全國第十
于○	吳○展	羅○昌	李○杰	丁○妮	羅○哲	吳○進	陳○廷

高○孫○宇	高○于○	高○廖○	高○郭○	普○湯○	普○王○	普○陳○	普○徐○
高○邱○	高○王○	高○黃○	高○廖○	普○林○	普○邱○	普○鄭○	普○楊○
高○高○	高○施○	高○賴○	高○羅○	普○方○	普○許○	普○吳○	普○林○
高○林○	高○方○	高○黃○	高○劉○	普○高○	普○鄧○	普○曾○	普○許○
高○傅○	高○程○	高○張○	高○李○	普○鄭○	普○宋○	普○賴○	普○楊○
高○梁○	高○王○	高○郭○	高○曾○	普○林○	普○黃○	普○張○	普○林○
高○施○	高○楊○	高○胡○	高○于○	普○郭○	普○劉○	普○劉○	普○詹○
高○劉○	高○傅○	高○許○	高○陳○	普○黃○	普○張○	普○陳○	普○于○
高○鄧○	高○郭○	高○許○	普○王○	普○盧○	普○褚○	普○廖○	普○邱○
高○涂○	高○林○	高○陳○	普○梁○	普○朱○	普○李○	普○楊○	普○于○

普考榜眼
高普雙榜
半年考取
應屆考取



非常感謝補習班提供的題目資源，使得真正在上場考試時，總有這樣的心得：「好耶，這題我完全會寫。」當下真的非常開心，因為我先前沒有去報考其他考科練筆，完全依靠補習班資源，有這樣的結果實在太好了。

羅○昌 高普考-資訊處理

三、請說明下列程式設計概念的差異。(每小題 5 分，共 20 分)

(一)請說明傳址 (call-by-reference) 與傳值 (call-by-value) 參數傳遞的差異。

(二)請說明靜態及動態記憶體 (static memory allocation vs. dynamic memory allocation) 配置的主要差別。

(三)請說明語法錯誤 (syntax error)、語意錯誤 (semantic error)、執行錯誤 (run-time error) 的主要差別。

(四)上述的錯誤，編譯程式過程中可以發現的是那一種錯誤 (可複選)？

《考題難易》★★

《破題關鍵》：程式語言的基本觀念題型，大部分的內容在課堂上都有提到，同學應不難作答。

【擬答】

(一)傳址：參數傳遞是傳送變數的記憶體位址，可在函式中取出此位址源頭的值來使用。

傳值：參數傳遞是傳送變數的值，在函式內對此值的變更不會影響到源頭的值。

(二)靜態記憶體配置：由編譯器分配的記憶體空間。

動態記憶體配置：程式在執行時才配置的空間，彈性比靜態配置高。

(三)

1. 語法錯誤：程式本身的文法錯誤，無法通過編譯。

2. 語意錯誤：程式可以通過編譯，但是執行的結果與預期有落差。ex：

```
int n = 5;
```

```
if (n >= 5)
```

```
    cout << "yes";
```

```
if (n <= 5)
```

```
    cout << "no";
```

3. 執行錯誤：程式碼可編譯，但是執行時會發生錯誤，例如陣列索引超出範圍或是分母等於 0。

4. 編譯程式過程中可以發現：語法錯誤。



順利考取有訣竅



一年考取 高普雙榜
112高普考資訊處理
涂○璋

各科老師都會詳細解說考試重點，讓我好好學習那些不熟悉的知識。考前總複習班各科老師以最短的時間來幫大家複習重點及預測考試的命題趨向，對於一整年的課程有畫龍點睛的功效。



非本科系 高普雙榜
112高普考資訊處理
傅○華

老師都是以實例還有時事去講解，所以非常的清楚也好記憶，跟著老師的步驟，配合每一次章節結束的歷屆試題講解，就知道遇到題目時該如何解題，讓非本科生的我受益良多。

想了解更多訣竅？

歡迎至 志光.學儒.保成 全國門市洽詢

公職王歷屆試題 (112 地方政府特考)

四、超商預計發展合併集點卡程式，說明如下。若有 n 張集點卡要合併，但只能兩張兩張合併，因此共需合併 $n-1$ 次才能把點數全部集中到一張卡。2 張合併時會扣掉較少點數那張的 $1/10$ 點數（無條件進位至整數）做為手續費。例如，若有 A, B, C 3 張集點卡要合併，且點數分別為 25, 30, 51 點。若先合併 A, B，再合併 C，則會扣掉 $3+6$ 點，因此合併後剩 97 點，這也剛好是最差合併策略下的合併總點數；但在最佳的合併策略下（先合併 B, C，再合併 A），則可有 100 點。（每小題 15 分，共 30 分）

(一)請寫 `best_case()` 函式，計算 n 張集點卡合併過後最多可有多少點數。

(二)請寫 `worst_case()` 函式，計算 n 張集點卡合併過後最少可有多少點數。

```
#include <stdio.h>

int a[101], n; // 最多可有 100 張集點卡要合併，實際上要合併 n 張卡
int best_case () {
    ...
}
int worst_case () {
    ...
}

int main () {
    scanf ("%d", &n);
    for (i=0; i<n; i++)
        scanf ("%d", a[i]);
    printf ("Best case: %d points\n", best_case());
    printf ("Worst case: %d points\n", worst_case());
    return 0;
}
```

《考題難易》★★★★★

《破題關鍵》：要注意題目的輸入是存到陣列。本題採用 C++ 的 `vector` 來儲存跟排序，再轉換成陣列。這樣程式撰寫跟理解上相對容易，但是要熟悉 `vector` 相關函式的使用。

【擬答】

```
int best_case () {
    int sum = 0;
    sort(a, a + n);
    vector<int> v(a, a + n);
    sum += v.back(); // 倒數第一筆
    v.pop_back();

    int m = v.back();
    v.pop_back(); // 倒數第二筆
    int k = m / 10;
    if(m % 10 != 0)    k++;
    sum += m - k;
}
```

```
for(int i = 0; i < n - 2; i++){
    int m2 = v.back();
    v.pop_back();
    int k2 = m2 / 10;
    if(m2 % 10 != 0) k2++;
    sum += m2 - k2;
}
return sum;
}
```

```
int worst_case () {
    int sum = 0;
    sort(a, a + n);
    vector<int> v(a, a + n);
    sum += v[0]; // 第一筆
    v.erase(v.begin());

    int m = v[0]; // 第二筆
    v.erase(v.begin());
    int k = m / 10;
    if(m % 10 != 0) k++;
    sum += m - k;

    for(int i = 0; i < n - 2; i++){
        int m2 = v[i];
        v.erase(v.begin());
        int k2 = m2 / 10;
        if(m2 % 10 != 0) k2++;
        sum += m2 - k2;
    }
    return sum;
}
```