

等 別：四等考試
類 科：資訊處理
科 目：程式設計概要
考試時間：1 小時 30 分

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

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)

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
```

...

```
?>
```

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)？

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;
```

```
}
```