《程式設計概要》

一、撰寫遞迴函式是重要程式設計技巧之一。

㈠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;
}
```

㈡請將findnum()函式改寫成非遞迴的函式。（10分）

㈢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;
}
```

㈣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）

試題評析	本試題是測驗迴圈與遞迴的題目，這是上課時一直強調的每年必考題。題目會給迴圈或遞迴的程式，要求改為另一種寫法，或者直接給問題要求用這兩種方式寫出程式。進位轉換已經出了很多次，包含最大公因數、費氏級數等問題都是常見的考題，本題容易輕鬆取得高分。
考點命中	《高點・程式設計概要講義》第一回，許振明平編撰，頁80、97、170。

答：

(一)找出陣列price內的最大正整數。

(二)

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

    }
```

```
        return num;
}
```
(三)十進位數值轉二進位數值，如(12)10=(1100)2
(四)
```
void convert (long m) {
        if(m==0){
                printf("0");
                return;
        }else{
                convert(m/2);
                printf("%d", m%2);
        }
}
```

二、請說明下列PHP程式設計的觀念。

㈠Class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）

㈡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

   . . .

?>
```

試題評析	類別(class)是一種自訂的抽象型資料型態，介面(interface)是函數成員只宣告不定義內容的特殊類別，由子類別(subclass)實作父介面後實作函數內容。至於PHP Web程式，每年必考題，今年測驗PHP上傳檔案功能，注意要寫出題目要求的三個重點：(1)測試上傳的檔案檔名是否存在？(2)上傳檔案是否成功？(3)將檔案搬到指定的目錄下。只要寫出這三段程式，就可獲取高分。
考點命中	《高點・程式設計概要講義》第二回，許振明平編撰，頁165-166、207。

答：

(一)

class:
可宣告屬性
可實例化
可實作方法

interface:
可宣告屬性(靜態常數)
不可實例化
不可實作方法

(二)

```
<?php
    $currentDirectory = getcwd();
    $uploadDirectory = "/uploads/";
    $fileName = $_FILES['to_be_uploaded']['name'];
    $uploadPath = $currentDirectory . $uploadDirectory . basename($fileName);

    $fileTmpName    = $_FILES['to_be_uploaded']['tmp_name'];

    $uploadOk = true;

    if (isset($_POST['submit'])) {

        if (file_exists($uploadPath)) {
            echo "Sorry, file already exists.";
            $uploadOk = false;
        }

        if($uploadOk){
            $didUpload = move_uploaded_file($fileTmpName, $uploadPath);
            if ($didUpload) {
                echo "The file " . basename($fileName) . " has been uploaded";
            } else {
                echo "An error occurred. Please contact the administrator.";
            }
        }
    }
?>
```

三、請說明下列程式設計概念的差異。（每小題5分，共20分）

㈠請說明傳址（call-by-reference）與傳值（call-by-value）參數傳遞的差異。

㈡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。

㈢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。

㈣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）？

試題評析	本試題測驗一些程式語言的觀念，包含傳值呼叫(call by value)與傳位址呼叫(call by address)的差異度，一個是傳變數值而另一個是傳變數位址給函數。靜態變數與動態變數最主要差異在靜態變數放在永久性的heap資料結構，而動態變數放在暫時性的stack資料結構。至於錯誤(error)的問題則與編譯程式(compiler)有關聯度，仔細回答可得高分。
考點命中	《高點・程式設計概要講義》第一回，許振明平編撰，頁103-109、230。

答：

(一)傳值呼叫(call by value)與傳址呼叫(call by address)

相同點：都是由主程式(或函數)傳遞資料給函數。

相異點：傳值呼叫(call by value)傳遞的是值，函數不會影響主程式(或函數)的變數內容值。傳址呼叫(call by address)傳遞的是變數位址，函數可以影響主程式(或函數)的變數內容值。

(二)1.靜態及動態記憶體(static memory allocation)

生命期：所在的函數(或區塊)

存取範圍：所在的函數(或區塊)

配置的資料結構：stack

2.動態記憶體(dynamic memory allocation)
生命期：程式開始到結束
存取範圍：主程式外宣告，程式開始到結束。
函數內宣告，存取範圍只能在函數內。
配置的資料結構：heap

(三)1.語法錯誤(syntax error)：語法錯誤就是我們所撰寫的程式碼中有不符合程式語言(如C語言)語法的規定。
2.語意錯誤(semantic error)：又稱為邏輯錯誤(logic error)，當程式本身的語法沒有錯誤，但是執行的結果不符合我們的預期，此時可能就是犯了語意上的錯誤，也就是程式有邏輯上的錯誤。
3.執行錯誤(run-time error)：編譯程式編譯通過，但執行時發生錯誤。如除法分母為0。。

(四)編譯程式可以發現的錯誤：語法錯誤(syntax error)。

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,C3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）

㈠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。

㈡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;
}
```

試題評析	本題合併集點卡程式要求計算合併卡後可得到最多與最少的點數。要得到最多點數需要每次從點數最多的兩張卡開始合併，如果要得到最少的點數則需要每次從最少的兩張卡開始合併，因此需要使用插入排序法(insertion sort)將每次合併後的點數放回並排序。因此，只要抓到這個重點，本試題就容易獲得高分。
考點命中	《高點・程式設計概要講義》第一回，許振明平編撰，頁221-222。

答：

(一)

```
int best_case () {
    int i,j,item,data,fee,x[101];
```

```
	for(i=0;i<n;i++)
	 x[i]=a[i];
	// sort small to large
	for(i=1; i<n; i++){
	 item=x[i];
	 for(j=i-1; j>=0; j--){
		if(x[j]>item){
			x[j+1]=x[j];
		}else{
			x[j+1]=item;
			break;
		}
	 }
	}
	if(j<0) x[0]=item;

	for(i=n-1;i>0;i--){
	 fee=(x[i-1]%10==0?x[i-1]/10:x[i-1]/10+1);
	 data=x[i]+x[i-1]-fee;
	 // insert data
	 for(j=i-2; j>=0; j--){
		if(x[j]>data){
			x[j+1]=x[j];
		}else{
			x[j+1]=data;
			break;
		}
	 }
	 if(j<0)	x[0]=data;
	}
	return data;
}
(二)
int worst_case() {
	int i,j,item,data=0,fee,x[101];

	for(i=0;i<n;i++)
	 x[i]=a[i];
	// sort large to small
	for(i=1; i<n; i++){
	 item=x[i];
	 for(j=i-1; j>=0; j--){
		if(x[j]<item){
			x[j+1]=x[j];
		}else{
			x[j+1]=item;
			break;
		}
	 }
```

```
    if(j<0)         x[0]=item;
   }

   for(i=n-1;i>0;i--){
    fee=(x[i]%10==0?x[i]/10:x[i]/10+1);
    data=x[i]+x[i-1]-fee;
    // insert data
    for(j=i-2; j>=0; j--){
            if(x[j]<data){
                    x[j+1]=x[j];
            }else{
                    x[j+1]=data;
                    break;
            }
    }
    if(j<0)         x[0]=data;
   }
   return data;
}
```