

《程式設計概要》

一、根據以下C程式碼，回答下列問題：

```

1 #include <stdio.h>
2 void g(int *x, int *y) {
3     *x = *y - 1;
4     *y = *y - 2;
5     printf("*x = %d, *y = %d\n", *x, *y);
6 }
7
8 void f(int *z, int *w) {
9     int n = 13;
10    g(w, &n);
11    printf("*z = %d, *w = %d, n = %d\n", *z, *w, n);
12 }
13
14 int main() {
15     int i = 21;
16     int j = 5;
17     int n = 137;
18     f(&i, &j);
19     printf("print 1: i = %d, j = %d, n = %d\n", i, j, n);
20     g(&n, &n);
21     printf("print 2: i = %d, j = %d, n = %d\n", i, j, n);
22     return 0;
23 }

```

(一) 請說明程式碼第8行至第12行中，每一行之運算子符號「*」及「&」操作所代表的意義。

(10分)

(二) 請問該程式碼執行後輸出的結果為何？(15分)

試題評析	本試題是測試C語言指標的方法，C語言的指標用法是C語言最重要的語法，利用記憶體位址控制程式的重要規則，其中最常見的用法是呼叫函數時使用傳位址呼叫(Call By Address)，本試題就是測驗這個重點，平常用功的同學容易取得高分。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁157-165。

答：

(一)

```

void f(int *z, int *w){ // 使用整數指標z, w接收傳入的整數變數的位址
    int n=13;           // 宣告區域變數n並初始化為13
    g(w, &n);          // 呼叫函數g並傳入w內容值(call by value)與n的位址(&n)(call by address)
    printf("*z=%d, *w=%d, n=%d\n", *z, *w, n); // 列印z與w指標控制的變數內容值，列印變數n內容值
}

```

(二)

```

*x=12, *y=11
*z=21, *w=12, n=11
print 1:i=21,j=12,n=137
*x=134, *y=134
print 2:i=21,j=12,n=134

```

【版權所有，重製必究！】

二、根據以下C程式碼，回答下列問題：

```

1 #include <stdio.h>
2 int f(int a, int b) {
3     if (b=1) return a;
4     else {
5         int value = a + f(a, b-1);
6         return value;
7     }
8 }
9 int main() {
10    printf("%d\n", f(5, 3));
11    printf("%d\n", f(0, 0));
12 }
```

- (一) 請問那一行是此遞迴程式的中止條件？(5分)
 (二) 請說明第10行的輸出，以及其運作邏輯。(6分)
 (三) 請說明第11行的輸出，以及其運作邏輯。(7分)
 (四) 請說明函式f的主要功能，包含輸入參數與輸出結果的關係。(7分)

試題評析	本試題測驗C語言的遞迴用法，遞迴(Recursive)是函數直接呼叫函數本身來完成工作，遞迴具有邏輯性，程式撰寫比較短，但執行時間比較長，因此大多會將遞迴程式轉換為迴圈寫法以減少執行時間，仔細作答可以獲取高分。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁131-139。

答：

- (一)
第03行
if(b==1) return a;
(二)
呼叫f(5,3)，將傳回結果使用printf()函數印出
 $f(5,3) = 5 + f(5,2) = 5 + 5 + f(5,1) = 5 + 5 + 5 = 15$
(三)
呼叫f(0,0)，將傳回結果使用printf()函數印出
f(0,0) 無傳回值
(四)
 $f(a,b) = a * b, b >= 1$

三、關於搜尋法程式：

- (一) 說明循序 (Sequential) 搜尋法，以及二元 (Binary) 搜尋法的優缺點。(8分)
 (二) 以下二元搜尋程式碼有部分錯誤，若要修正為正確程式，請說明「最少」需修改程式碼行數、原因與修改方法。(17分)

```

1 public class BinarySearch{
2     public int faultyBinarySearch(int[] arr, int x) {
3         int l=0, r=arr.length-1;
4         int m=(1+r)/2;
5         while(l<=r) {
6             m=(1+r)/2;
7             if(arr[m] = x) return m;
8             if(arr[m] > x) l=m+1;
9             if(arr[m] < x) r=m-1;
10        }
```

11	return -1;
12	}
13	}

試題評析	二分搜尋法(Binary Search)是資料結構當中重要的演算法，主要在資料已排序的前提下可以迅速完成搜尋動作，時間複雜度是 $O(\log 2n)$ ，用功的同學可得高分。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁200-201。

答：

(一)

1.循序(Sequential)搜尋法

優點：演算法簡單

缺點：時間複雜度高， $O(n)$

2.二元(Binary)搜尋法

優點：時間複雜度小， $O(\log 2n)$

缺點：資料須事先排序

(二)

```
public static int faultyBinarySearch(int []arr, int x){
    int l=0, r=arr.length-1;
    int m=(l+r)/2;
    while(l<=r){
        m=(l+r)/2;
        if(arr[m]==x) return m;
        if(arr[m]>x) r=m-1;           // x比較小，往左半部繼續搜尋
        if(arr[m]<x) l=m+1;         // x比較大，往右半部繼續搜尋
    }
    return -1;
}
```

四、請依據以下Java程式碼回答下列問題：

```
01 import java.util.Scanner;
02 public class MyJavaCode {
03     public static void main(String[] args) {
04         Scanner in = new Scanner(System.in);
05         int a = in.nextInt();
06         Animal dog = new Dog();
07         dog.run(a);
08         dog.run();
09     }
10 }
11
12 class Dog extends Animal {
13     public void run(int units) {
14         System.out.println("Dog's running speed is " + units);
15     }
16     public void run() {【版權所有，重製必究！】
17         System.out.println("Dog's default running speed is 20");
18     }
19 }
```

(一)請說明Java中class、abstract class與interface之間的主要差異？(10分)

(二)請說明上述程式碼那幾行會產生編譯錯誤，以及錯誤的原因。(5分)

(三) 不改變原1-19行程式碼順序的條件下，請修正問題(二)之編譯問題，使之可以正常執行。修正不包含左右大括號「{}」，新增或修改之指令行數不可以超過3行。(10分)

試題評析	本試題測試物件導向程式中的抽象類別用法，父類別宣告抽象函數成員，具體子類別必須覆寫繼承於父類別的抽象函數後，才可以產生物件使用，對物件導向清楚的學員可以獲取高分。
考點命中	《高點·高上程式設計概要講義》第一回，許振明編撰，頁203-205。

答：

(一)

Class：具體類別，所有函數都必須定義內容，具體類別可以產生物件。

abstract class：可以有抽象函數的類別，抽象類別不可產生物件，需要子類別繼承抽象類別後實作抽象函數形成具體子類別才能產生物件。

interface：所有函數都是抽象函數的類別，子類別若實作父介面部分函數則形成抽象類別，如果完整實作所有函數則形成具體子類別。

(二)

第06行

第08行

第12行

此三行都是因為不存在父類別Animal而產生的錯誤。

(三)

```
abstract class Animal{
    public abstract void run(int units);
    public abstract void run();
}
```

【版權所有，重製必究！】