

# 1121 NTOUCSE 程式設計 1C 期末考參考答案

112/12/26 (二)

1. 請問下面四列程式分別會產生哪些編譯時的錯誤? [8] (下面所列的是 Visual C/C++ 的編譯錯誤訊息, 沒有按照順序, 有些程式列會產生多個錯誤, 請以錯誤編號 C##### 作答), 請解釋該錯誤的原因並且修改成為可以編譯正確的敘述?[12]

```
01. int n=3, array1[n];
02. int array2[3]={1,2,3,4};
03. int *array3="abcd";
04. array1 = array2;
```

error C2440: '正在初始化': 無法由 'const char [5]' 轉換為 'int \*'

指向的型別沒有相關; 轉換必須有 reinterpret\_cast、C-Style 轉換或函式樣式 轉換

error C2133: 'array1': 未知的大小

error C2466: 無法配置常數大小為 0 的陣列, 常數必須為大於 0 的整數

error C2106: '=': 左運算元必須是左值 (l-value)

error C2440: '=': 無法由 'int [3]' 轉換為 'int []'

error C2057: 必須是常數運算式

error C2078: 初始設定式太多

**Sol:**

01. `int n=3, array1[n];`

error **C2057**: 必須是常數運算式

error **C2466**: 無法配置常數大小為 0 的陣列, 常數必須為大於 0 的整數

error **C2133**: 'array1': 未知的大小

在很多編譯器裡, `array1[n]` 這個 VLA 擴充語法是不被接受的, 定義陣列的時候大小需要是常數  
可以改成 `int array1[3];`

或是 `#define SIZE 3`

```
int array1[SIZE];
```

或是 `const int n=3;`

```
int array1[n];
```

如果真的需要動態地配置陣列大小, 可以使用 `malloc()`, 例如

```
int array1=(int *)malloc(sizeof(int)*3);
```

...

```
if(array1) free(array1), array1=0; // 不再需要時釋放掉, 可以在相同函式裡或是在
```

```
// 能夠取得此指標值的任何地方, 請注意在 GNU C
```

```
// 的 VLA 擴充語法中 int array[n], 在函式結束時
```

```
// 陣列一定會被釋放掉而不能在離開函式以後繼續使用
```

02. `int array2[3]={1,2,3,4};`

error **C2078**: 初始設定式太多

陣列 `array2[]` 有三個元素, 但是初始設定式裡面有四個元素, 應該要讓初始化的資料數量和陣列元素個數相同, 或是陣列元素個數讓編譯器根據初始化的資料數量來決定, 例如

```
int array2[3]={1,2,3}; 或是
```

```
int array2[4]={1,2,3,4}; 或是
```

```
int array2[]={1,2,3,4};
```

```
03. int *array3="abcd";
```

error C2440: '正在初始化': 無法由 'const char [5]' 轉換為 'int \*'

指向的型別沒有相關; 轉換必須有 reinterpret\_cast、C-Style 轉換或函式樣式 轉換  
錯誤訊息中 const char [5] 指的就是字串常數 "abcd", 在 C/C++ 中字串常數就是一個字元陣列, 這個陣列中有 5 個元素, 看不到的就是字串的結束字元 '\0', 如果指標是 char \* 型態或是 const char \*, 初始化為 "abcd" 語法就是正確的, "abcd" 就是那個字元陣列第一個元素的記憶體位址, 錯的就是上面的 int \*, 下面的敘述就會是對的  
char \*array3="abcd"; 或是  
const char \*array3="abcd";

```
04. array1 = array2;
```

error C2106: '=': 左運算元必須是左值 (l-value)

error C2440: '=': 無法由 'int [3]' 轉換為 'int []'

array1 是打算定義成一個陣列的, 不能用 array1=array2; 這樣的語法, array1 只是一個位址常數, 並不是一個變數, 如果想要複製兩個陣列內容的話, 需要一個迴圈來複製  
for (i=0; i<3; i++)

```
array1[i] = array2[i];
```

如果只要讓 array1 成為 array2 的別名, 那麼 array1 需要定義為指標 int \*array1; array1 = array2; 才會是一個合法的敘述, 但是如此就只有一個陣列而已。

2. [5] 在 C 語法中該如何用指標語法替整數陣列 array4 定義一個別名 array5, 如此相同的一個陣列可以有兩個名稱?

```
int i, array4[20];  
_____  
for (i=0; i<20; i++)  
array5[i] = i;
```

**Sol:**

int \*array5=array4; 或是 int \*const array5=array4; 都可以使得 array5 成為 array4 的別名, 以 array4[i] 存取 array4 的每一個元素都可換成 array5[i] 來存取相同的記憶體, 第二個定義中 \* 後面的 const 目的是讓 array5 這個指標變數內容在設定為 array4 之後無法再修改為其它的位址, 上面的 for 迴圈裡透過 array5[i] 來修改陣列的內容, 所以 const int \*array5=array4; 或是 const int \*const array5=array4; 這兩個宣告都是不可以的, int 前面的 const 就是讓 array5[i] 成為資料內容不能修改的變數。

3. (a) [5] 請運用 struct 語法定義一個 CreditCard 的結構來描述一張信用卡的資訊, 包含 1. 16 位數卡號, 2. 有效月/有效年, 3. 持卡人姓名最長 50 個字元, 4. 3 位數驗證碼 (CVV/CVC), 資料範例如 1. 5412 3456 7890 1232, 2. 12/25 3. WANG, DAVID, 4. 911, 各個資料欄位的名稱以及資料型態要求如下: 1. cc\_number 字元陣列, 2.1 expiration\_month 整數, 2.2 expiration\_year 整數, 3. user\_name 字元陣列, 4. cvv 字元陣列

**Sol:**

```
struct CreditCard {  
char cc_number[17];  
int expiration_month, expiration_year;  
char user_name[51];  
char cvv[4];
```

```
};
```

請注意 cc\_number 有 17 個字元，user\_name 有 51 個字元，cvv 有 4 個字元，都比需要的個數多一個字元是給結束字元 '\0' 用的。

(b) [4] 請以 struct CreditCard 型態定義一個結構變數 mycard，並且以上述資料範例初始化該變數  
Sol:

```
struct CreditCard mycard = {"5412345678901232", 12, 25, "WANG, DAVID", "911"};
```

請注意 char cc\_number[17]="5412345678901232"; 的初始化語法中是讓編譯器自己把字串常數切開成為 {'5','4','1','2','3','4','5','6','7','8','9','0','1','2','3','2'}

(c) [6] 假設已經有一個 struct CreditCard 型態的陣列 ccArray，其元素個數紀錄在整數變數 nCards 中，請完成下列程式片段，請運用動態的記憶體配置新的陣列 ccArraySorted，並且複製 ccArray 的內容到新的陣列中

```

#include <stdlib.h>
/* definition of struct CreditCard */
/* inside main() */
    struct CreditCard * ccArraySorted=NULL;
    ccArraySorted = (struct CreditCard *) malloc(nCards* sizeof(struct CreditCard));
    if (ccArraySorted == NULL) {
        printf("memory allocation failed\n");
        return 1;
    }
    for (int i=0; i<nCards; i++)
        ccArraySorted[i] = ccArray[i];

```

sizeof(mycard) 或是 sizeof(\*ccArray) 或是 sizeof(\*ccArraySorted)  
ccArraySorted == 0 或是 !ccArraySorted  
\*(ccArraySorted+i) = \*(ccArray+i);

(d) [5] 接題 (c)，請運用 stdlib.h 中的 qsort 函式以及 string.h 中的 strcmp 函式將此 ccArraySorted 陣列依照信用卡卡號由小到大排序

```
qsort(ccArraySorted, nCards, sizeof(struct CreditCard), (int (*)(const void*,const void*))strcmp);
```

/\* 如果信用卡卡號欄位在 CreditCard 結構的第一個的話，不需要另外定義比對函式 \*/

(e) [6] 請完成下列程式片段，運用 stdlib.h 中的 bsearch 函式，在已經排序好的 ccArraySorted 陣列中搜尋題 (b) 定義的 mycard，如果找到的話列印其持卡人姓名，否則列印想要搜尋的卡號

```

    struct CreditCard * result;
    result = (struct CreditCard *) bsearch(&mycard, /*參數 2~5 和上面的 qsort 參數完全一樣*/);
    if (result != NULL)
        printf("%s\n", result->user_name);
    else
        printf("Card %s not found!\n", mycard.cc_number);

```

result 或是 result != 0

(f) [4] 請問這個動態配置的陣列在不需要使用的時候該如何釋放? (因為釋放時指標是 NULL 或是所指到的記憶體先前已經被釋放掉了，都會導致記憶體管理的錯誤，因此習慣上會加上一些檢查)

```
if (ccArraySorted != NULL) { free(ccArraySorted); ccArraySorted=NULL; }
```

ccArraySorted 或是 ccArraySorted != 0

4. 請以 C 語法回答下列問題

(a) [4] 第 02 列如何定義一個 11 個整數的陣列 array，使得下列第 03~04 列程式中迴圈可以直接運用 array[-5], array[-4], ..., array[0], array[1],..., array[5] 來存取這 11 個元素的資料?

```
01 int i;
02 int array0[11], *array=array0+5 ;
03 for (i=-5; i<=5; i++)
04     array[i] = i;
```

(b) [4] 第 06 列如何在記憶體中定義一個連續存放 6\*11 個整數的二維陣列 array2，希望第一維的範圍是 50~55，第二維的範圍是-5~5，使得下列第 07~09 列程式中迴圈可以直接運用 array2[50][-5], array2[50][-4], ..., array2[50][5], array2[51][-5], array2[51][-4], ..., array2[51][5], ..., array2[55][-5], array2[55][-4], ..., array2[55][5],來存取這 66 個元素的資料

```
05 int i, j;
06 int arraymem[6][11], (*array2)[11]=(int (*)(11))(int *(arraymem-50)+5);
07 for (i=50; i<=55; i++)
08     for (j=-5; j<=5; j++)
09         array2[i][j] = i*11+j;
```

(c) [5] 接續題 (a) 與題 (b) 中 array 與 array2 之程式，請問下列敘述列印的結果為

```
printf("%d ", *(array-2)); -2
printf("%d ", array[-4]+5); 1
printf("%d ", **(array2+50)); 550
printf("%d ", (*(array2+51)-5)); 556
printf("%d ", array2[55]-array2[50]); 55
```

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

5. [16] 深度優先的搜尋最直接的實作就是固定層數的多層迴圈，遞迴的實作則可以得到層數可以調整的多層迴圈，例如二進位四個位元有 16 種，一般列舉的順序就是四位數字由小到大的順序，下面的四層迴圈可以列舉右圖的 16 種二進位數字

```
for (i[0]=0; i[0]<=1; i[0]++)
    for (i[1]=0; i[1]<=1; i[1]++)
        for (i[2]=0; i[2]<=1; i[2]++)
            for (i[3]=0; i[3]<=1; i[3]++) {
                for (j=0; j<4; j++)
                    printf("%d ", i[j]);
                printf("\n");
            }
```

0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000

請完成下面程式來列舉出右圖的 Gray Code，Gray Code 的編碼順序從第二個位元開始有一點不太一樣，前半段是先 0 再 1，後半段則是先 1 再 0，第三位元和第四位元也是這樣 01, 10, 01, 10 交替出現)

```
01 #include <stdio.h>
02
03 void gray(int n, int x[], int p) {
04     if (p>=n)
05         for (int i=0; i<n; i++)
06             printf("%d%c", x[i], "\n "[i<n-1]);
```

```

07     else {
08         gray(n, x, p+1);
09         x[p] = 1-x[p];
10         gray(n, x, p+1);
11     }
12 }
13
14 int main() {
15     int i, n, x[10];
16     while (1==scanf("%d", &n)) {
17         for (i=0; i<n; i++) x[i]=0;
18         gray(n, x, 0);
19     }
20     return 0;
21 }

```

6. [16] 請運用下面描述的規則撰寫遞迴函式來有效率地計算  $x^n$

(a)  $n$  是偶數時  $x^n = (x^{n/2})^2$ ;  $n$  是奇數時  $x^n = x(x^{(n-1)/2})^2$

```

01 #include <stdio.h>
02
03 double power1(double x, int n) {
04     double y;
05     if (n==0) // 遞迴結束條件
06         return 1;
07     else if (n%2==1) // n 是奇數
08         return x * power1(x, n-1);
09     else { // n 是偶數
10         y = power1(x, n/2);
11         return y*y;
12     }
13 }
14
15 int main() {
16     printf("%f\n", power1(2.5,9));
17     return 0;
18 }

```

(b)  $n$  是偶數時  $x^n = (x^2)^{n/2}$ ;  $n$  是奇數時  $x^n = x(x^2)^{(n-1)/2}$

```

01 #include <stdio.h>
02
03 double power2(double x, int n) {
04     if (...) // 同題 (a)
05         ...// 同題 (a)
06     else if (...) // 同題 (a)
07         ...// 同題 (a)
08     else
09         return power2(x*x, n/2);
10 }
11
12 int main() {
13     printf("%f\n", power2(2.5,9));
14     return 0;
15 }

```