

P3-3

例 要知道常值或變數所占記憶體的长度，可用 sizeof()函式來查詢： (sizeof.c)

```

printf("char 資料常值占用的記憶體大小： %d \n", sizeof('A'));
printf("int 資料常值占用的記憶體大小： %d \n", sizeof(123));
printf("double 資料常值占用的記憶體大小： %d \n", sizeof(123.45));

```

```

char 資料常值占用的記憶體大小： 4
int 資料常值占用的記憶體大小： 4
double 資料常值占用的記憶體大小： 8

```



例 要知道常值或變數所占記憶體的长度，可用 sizeof()函式來查詢： (sizeof.c)

```

char c = 'A'; // c 為字元變數, 變數數值為'A'
printf("char 資料變數占用的記憶體大小： %d \n", sizeof(c)); // c 為變數
printf("int 資料常值占用的記憶體大小： %d \n", sizeof(123)); // 123 為常值
printf("double 資料常值占用的記憶體大小： %d \n", sizeof(123.45)); // 123.45 為常值

```

```

char 資料變數占用的記憶體大小： 1
int 資料常值占用的記憶體大小： 4
double 資料常值占用的記憶體大小： 8

```

P3-10

1. 一元運算子 (Unary Operator)：-(負)、++(遞增)、--(遞減)，如：-5、++x、y--。

P3-16

```

例 int x=2, y=3;
x >= y && x != y || x * 2 > y == 1

```

P3-19

題目 (二)

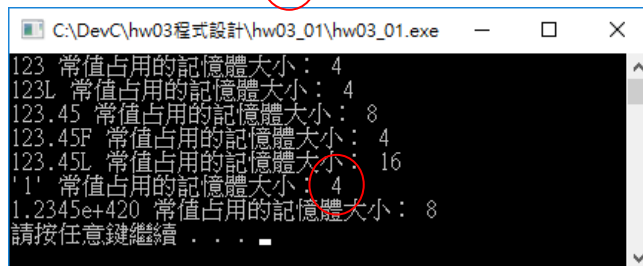
若 a、b、c、d、e 均為整數變數，下列哪個算式計算結果與 a+b*c-e 計算結果相同？

- (A) (((a+b)*c)-e) (B) ((a+b)*(c-e)) (C) ((a+(b*c))-e) (D) (a+((b*c))-e) (106 年檢測試題)

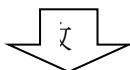
P3-23

1. 撰寫一個程式，使用 sizeof() 函式顯示下列常值所佔的記憶體長度。

123、123L、123.45、123.45F、123.45L、'1'、1.2345e+420

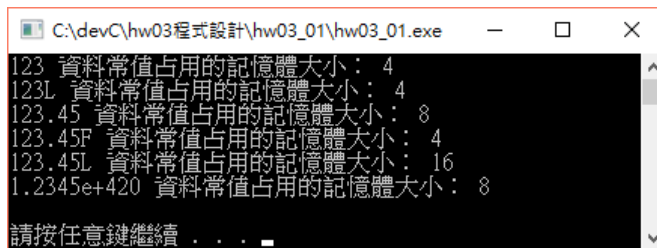


```
C:\DevC\hw03程式設計\hw03_01\hw03_01.exe - _ □ ×
123 常值占用的記憶體大小： 4
123L 常值占用的記憶體大小： 4
123.45 常值占用的記憶體大小： 8
123.45F 常值占用的記憶體大小： 4
123.45L 常值占用的記憶體大小： 16
'1' 常值占用的記憶體大小： 4
1.2345e+420 常值占用的記憶體大小： 8
請按任意鍵繼續 . . .
```



1. 撰寫一個程式，使用 sizeof() 函式顯示下列常值所佔的記憶體長度。

123、123L、123.45、123.45F、123.45L、1.2345e+420



```
C:\devC\hw03程式設計\hw03_01\hw03_01.exe - _ □ ×
123 資料常值占用的記憶體大小： 4
123L 資料常值占用的記憶體大小： 4
123.45 資料常值占用的記憶體大小： 8
123.45F 資料常值占用的記憶體大小： 4
123.45L 資料常值占用的記憶體大小： 16
1.2345e+420 資料常值占用的記憶體大小： 8
請按任意鍵繼續 . . .
```

P4_21

二. 問答題

1. 使用 printf() 函式將十進制整數 253，改由八進制與十六進制具有前導字元的格式輸出？

① 八進制：_____ 輸出結果：_____

② 十六進制：_____ 輸出結果：_____

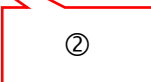
2. 使用 printf() 函式輸出下列兩行字串？

① 輸出內容：「"Nice to meet you! "」，頭尾要有雙引號。

答：_____

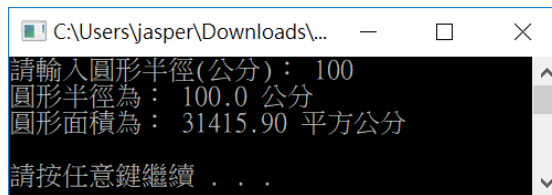
① 輸出內容：「環保民意支持度有 Δ75%」，Δ 代表空格。

答：_____



P4-22

4. 設計可計算圓形面積的程式，程式一開始詢問圓形半徑，輸入後顯示圓形的半徑和面積。(假設圓周率為 3.14159)



```
C:\Users\jasper\Downloads\... - □ ×
請輸入圓形半徑(公分)： 100
圓形半徑為： 100.0 公分
圓形面積為： 31415.90 平方公分
請按任意鍵繼續 . . .
```

P5-1

5.1 結構化程式設計

C 語言是一種「結構化程式設計」的程式語言。這種程式設計的技術，是透過程式的模組化和程式的結構化，來簡化程式設計的流程，降低邏輯錯誤發生的機率。這種程式設計的觀念，是由上而下的程式設計，將程式中可以有獨立功能的程式區塊分割出來使成為「模組」(Module)，這些模組最後再組合成一個大的完整程式軟體。「結構化程式設計」採用「循序結構」、「選擇結構」、「重複結構」這三個基本流程架構來設計程式。在前面章節所撰寫的程式架構是採用由上而下一行接著一行執行的「循序結構」。本章所要介紹的程式流程是因條件的不同而會執行不同的程式區塊，這種流程有選擇性的架構稱為「選擇結構」。下一章，我們再來接觸「重複結構」，這種流程會在條件成立的情況下重複執行相同的程式區塊。

P5-2

5.2 結構化程式設計 關係運算式

P5-4

5.3 邏輯運算式

一個關係運算式就是一個條件式，當要把多個條件式一起做判斷時，便需要「邏輯運算子」來連結運算。這個運算式稱為「邏輯運算式」，也可簡稱為條件式。其運算後也是有兩種結果，就是「1」(true)和「0」(false)，或者稱為「真」和「假」。C 語言提供的邏輯運算子如下：

P5-12

5.5 多重條件選擇 switch

switch 也是一個多向選擇結構，但與 if ... else if ... else 不同。if ... 的多向選擇結構使用多個不同的 (條件式) 來選擇執行的敘述區段。switch 的多向選擇是使用一個運算式，再根據運算式的結果(value)來判斷所要執行的 case 敘述區段。switch 的語法及流程圖如下：

P6-3

第 5 次	5	$i \leq 5$ (成立)	印出 5	6
第 6 次	6	$i \leq 6$ (不成立)	離開迴圈	

6 更正為 5

P7-23

2. 所輸入的數值依序存入陣列元素的註標順序為：X[2],X[3],X[4] ... X[9],X[0],X[1]

P7-24



說明

1. 答案是 (D)。
2. $a[1] = b[1] + a[0] = 1 + 0 = 1$
 $a[2] = b[2] + a[1] = 2 + 1 = 3$
 $a[3] = b[3] + a[2] = 3 + 3 = 6$
⋮
 $a[30] = b[30] + a[29] = 30 + 435 = 465$
⋮
 $a[50] = b[50] + a[49] = 50 + 1225 = 1275$
故 $a[50] - a[30] = 1275 - 465 = 810$

P7-25

 題目 (六)

定義 $a[n]$ 為一個陣列(array)，陣列元素的指標為 0 至 $n-1$ 。若要將陣列中 $a[0]$ 的元素移到 $a[n-1]$ ，下列程式片段的空白處該填入何種運算式？

```
01 int i, hold, n;  
02 ...  
03 for(i=0; i<=_____ ; i=i+1) {  
04     hold = a[i];  
05     a[i] = a[i+1];  
06     a[i+1] = hold;  
07 }
```

P7-25

 題目 (七)

下列程式片段主要功能為：輸入六個整數，檢測並印出最後一個數字是否為六個數字中最小的值。然而，這個程式片段是錯誤的！

```
01 #define TRUE 1  
02 #define FALSE 0  
03 ...  
04 int d[6], val, allBig;  
05 for(i=1; i<=5; i=i+1) {  
06     scanf("%d", &d[i]);  
07 }  
08 scanf("%d", &val);  
09 allBig = TRUE;  
10 for (i=1; i<=5; i=i+1) {  
11     if(d[i]>val) {  
12         allBig=TRUE;  
13     } else {  
14         allBig=FALSE;  
15     }  
16 }
```

P7-26

 題目 (八)

下列程式片段執行過程的輸出為何？

```
01 int i, sum, arr[10];  
02 for(i=0; i<10; i=i+1)  
03     arr[i]=i;  
04 sum=0;
```

```
05 for(i=1; i<9; i=i+1)
06     sum = sum-arr[i-1]+arr[i]+arr[i+1];
07 printf("%d", sum);
```

(A) 44 (B) 52 (C) 54 (D) 63

P8-3

floor	語法：double floor(double x); 說明：傳回不大於 x 的最大整數。 例：ceil(3.2) 傳回 3, ceil(4.7) 傳回 4, ceil(-0.5) 傳回 -1
-------	-------------------------------------------------------------------------------------------------------

ceil 皆更正為 floor

P8-15

二. 傳址呼叫

函式呼叫時，若採用「傳址呼叫」。編譯器會將實引數和虛引數所佔用的記憶體位址設為一樣，如此引數間的資料傳遞是雙向道，即

呼叫敘述實引數的資料內容 ↔ 自定函式虛引數

當呼叫敘述中的實引數傳入資料給自定函式的虛引數，若自定函式內虛引數內容有改變，則原呼叫敘述實引數的內容也跟著變數動。傳址呼叫的傳遞方式會使用傳遞整個陣列或指標變數，有關整個陣列的傳遞本章會提到，而指標變數的傳遞在第 11 章再來探討。

P8-25

題目 (三)

給定 g() 如下，g(13) 的傳回值為何？

```
01 int g(int a) {
02     if (a>1) {
03         return g(a-2)+3;
04     }
05     return a;
06 }
```

 題目 (四)

下列程式碼是一個計算 n 階層的函式，請問要如何修改才能得到正確的結果？

```

01 int fun(int n) {
02     int fac = 1;
03     if (n >= 0) {
04         fac = n * fun(n-1);
05     }
06     return fac;
07 }

```

- (A) 第 02 行，改為 `int fac = n;` (B) 第 04 行，改為 `fac = n * fun(n+1)`
 (C) 第 03 行，改為 `if (n > 0) {` (D) 第 04 行，改為 `fac = fac * fun(n-1)`
 (105 年檢測試題)

 說明

1. 答案是 (C)。
2. 若 $n = 0$ ，則 `fac` 最後會乘到 0，結果傳回值會為 ~~會為~~ 0。

2.
$$j = f(1) + f(2) + f(3)$$

$$f(1) = 1 + f(2) = 1 + 3 + f(3) = 1 + 3 + 1 + f(4) = 1 + 3 + 1 + 1 = 6$$

$$f(2) = 3 + f(3) = 3 + 1 + f(4) = 3 + 1 + 1 = 5$$

$$f(3) = 1 + f(4) = 1 + 1 = 2$$

$$j = 6 + 5 + 2 = 13$$

 題目 (六)

下列 $F()$ 函式的 `return` 回傳運算式該如何寫，才會使得 $F(14)$ 的回傳值為 40？

 說明

1. 答案是 (B)。
2.
$$F(14) = 14 + F(11)$$

$$= 14 + 11 + F(8)$$

$$= 14 + 11 + 8 + F(5)$$

$$= 14 + 11 + 8 + 5 + F(2)$$

$$= 14 + 11 + 8 + 5 + 2$$

$$= 40$$

題目 (十三)

給定一陣列 $a[10]=\{1,3,9,2,5,8,4,9,6,7\}$ ，i.e.， $a[0]=1, a[1]=3, \dots, a[8]=6, a[9]=7$ ，以 $f(a,10)$ 呼叫下列函式後，回傳值為何？

```

01 int f(int a[], int n) {
02     int i, index=0;
03     for(i=1; i<=n-1; i=i+1) {
04         if (a[i] >= a[index]) {
05             index = i;
06         }
07     }
08     return index;
09 }

```

- (A) 1 (B) 2 (C) 7 (D) 9

(105 年檢測試題)



說明

1. 答案是 (C)。
2. 在函式定義主體內， $index$ 變數值的變化依序為 1,2,7。

#ifdef 巨集名稱

敘述區段 1

/*若指定巨集已經定義，就編譯敘述區段 1*/

#else

敘述區段 2

/* 更正為譯 編譯敘述區段 2*/

#ifndef 巨集名稱

敘述區段 1

/*若指定巨集尚未定義，就編譯敘述區段 1*/

#else

敘述區段 2

/* 更正為譯 編譯敘述區段 2*/

P11-1

11.1.1 指標與取址運算子

所以什麼是指標(Pointer)？

所以什麼是指標(Pointer)？我們在程式中宣告的每一個變數，系統都會在記憶體中安

，則被稱為「直接取值」。兩行的執行結果都是 15。

P11-2

接取值」。第 3 行以變數名稱存取資料，則被稱為「直接取值」。

```
.....*指標;... /* 取值運算 */
```

變數的空指標。也可以同時宣告多個指標變數，語法如下：

P11-2

時不同時賦與初始值，最好將指標指向 NULL 方便識別。這時我們變數的空指標。

```
.....資料型別 *指標變數名稱 1, *指標變數名稱 2, ...;
```

x 的指標變數。宣告指標變數時同時指定初始值時，語法如下：

P11-3

x 的指標變數。

```
.....資料型別 *指標變數名稱 1, 0, 變數名稱 ..
```

觀察變數宣告後，記憶體配置的分佈情形，發現先放入的資料所對應的記憶體位址較大；後放入的資料記憶體位址愈小。這種先進後出(First In Last Out：FILO)的模式，是使用堆疊管理記憶體的特徵。另外，記憶體位址可能會和本書執行不同。

P11-4



1. 觀察變數宣告後，記憶體配置的分佈情形，先放入的資料所對應的記憶體位址較大；後放入的資料記憶體位址愈小。這種先進後出(First In Last Out：FILO)的模式，是使用堆疊管理記憶體的特徵。

P11-5

將等號右邊的位址設定

一、指

將等號右邊的值設定給左邊的指標變數。例如：下面敘述將 iptr1 和 iptr2 指標變

P11-6

```
·04· printf("a[%d]=%x\tptr+%d=%x\n", i, &a[i], i, ptr+i);
·05· ptr += 3; /* 指標往高位址移動 3 個資料長度，也就是 12 */
```

，也就是 12 */

P11-6



說明

1. 第 4 行：以**陣列的方式及**指標的方式

1. 第 4 行：以**陣列的方式及**指標的方式取出陣列元素的位址。

指向**個**各自的

P11-8

指向**個**各自的陣列，再以 ary 指向整個陣列。所以要取得位址可以下列語法：

```
(1) ... 指標變數 = 指標變數[m] + n ...
```

P11-8

[例] x=*(ary[1]+2); //取得**第二維第三個元素**的內容

```
(1) 變數 = *(陣列名稱[m] + n);
```

```
[例] x=*(ary[1]+2);
```

P11-10

```
swap(&a, &b);
```

```
swap(&a, &b); /* 實引數加上&符號，表示傳實引數位址給虛引數 */
```

P11-19



說明

2. 觀察函式可發現其**的**功能為

1. 答案是(C)。

2. 觀察函式可發現其的功能為循序檢視陣列元素，求得最大值的註標，所以以陣列值代

P11-21

二. 程式設計

提示：1 公里=0.62137 英里=0.53995 海里。

1. 試撰寫一函式，函式功能：公里換算成英里及海里，函式輸入整數變數公里，傳回值有兩個皆為浮點數英里及海里。提示 1 公里=0.62137 英里=0.53995 海里。

P12-6

下表中所介紹的字串函式可以讓您將**數值**字串轉換

12.3.1 字串轉換函式

下表中所介紹的字串函式可以讓您將字串轉換成倍精確浮點數、整數、長整數型別的資料，以方便在程式中計算。其說明如下：

P12-9

```

strlen
簡例：① char str1[]="C 語
.....n=strlen(str1);/* 中文字佔用兩個字元，所以 n1=5 */
② n=strlen("Hello, World!");/* n=13 */
/* n=13 (含空白字元) */

```

P12-10

```

strxfrm
/* 傳回 "VB 入門" 字串長度 6，接著再將 s1 "C#基礎必修課" 字串前面 */
strxfrm(s1, "VB 入門", 2);
/* 傳回 "VB 入門" 字串長度 6，接著再將 s1 字串前面 */
/* 2 個字元換成 "VB 入門" 字串最前面 2 個字元 */
/* 結果 s1 為 "VB 基礎必修課" 字串 */

```

P13-2

1. 定義結構和宣告結構變數分開敘

說明：

一個結構資料型別後，便可以使用下面語法來宣告一個屬於這個結構資料型別的變數：

```

..struct struct_name
..{
→ → ...
..};
→ ...
..struct struct_name 結構變數 1, 結構變數 2, ...;

```

定義結構和宣告結構變數分開敘述。

P13-2

2. 定義結構同時宣告結構變數 \Rightarrow :

```

--struct struct_name+
--{+
→ → ...+
--}結構變數 1, 結構變數 2, ...,+

```

定義結構同時宣告結構變數。

P13-2

3. 定定義結構時不賦與結構名稱，這種定義方式只能同時宣告結構變數 \Rightarrow :

```

--struct+
--{+
→ → ...+
--}結構變數 1, 結構變數 2, ...,+

```

定義結構時不賦與結構名稱，這種定義方式只能同時宣告結構變數。

P13-4

06 = {

```

05 struct student+
06 +
07 ... char no[8];+

```

索引註標值

P13-12

4. 第 30~35 行：由結構陣列的第一個項目，循序以 strcmp() 函式比對結構陣列欄位 no 和使用者輸入的資料，如果相同，則變數 flag 設成索引值，並結束迴圈。

P13-24

結構欄位：

下一項

函式功能 \Rightarrow 為



- 第 5~9 行：定義多項式結構欄位：使用鏈結串列來表示多項式，因此定義 poly 結構型別用來表示多項式的一個項，poly 結構成員包括 coef 表示係數；exp 表示指數(即次方)；next 指標變數表示鏈結欄位，用來指定串列的下一個項的節點。
- 第 43~56 行：定義 printpoly() 函式：函式功能，為顯示多項式，呼叫此函式時必須傳

while 迴圈 ~~→~~ 迴圈會執行

P13-25

7. 第 99~128 行: while 迴圈, 迴圈會執行到 poly1 及 poly2 指標移至該串列尾端為止(99 行)。

P13-33

3. struct a1 {

int id;

struct { int y

① date.m = 13 ② r.m = 13

char age;

}r;

上列結構中, 要存取變數 m, 應該如何敘述?

① a1.date = 10 ② r.date = 10 ③ r.date.m = 13 ④ 錯誤敘述

P14-2

/* 預設定值為 1 */

/* 預設定值為 5 */

4. 若要在定義時同時設定常數值, 只要直接使用 = 等號直接給予設定值即可, 其寫法如下:

```
enum weekday
{
    SUN=1, /* 預設定值為 1 */
    MON, /* 預設定值為 2 */
    TUE, /* 預設定值為 3 */
    WED=5, /* 預設定值為 5 */
    THU, /* 預設定值為 6 */
}
```

P14-9

/* 相當於 float a=7.0, b; */

[例 1] 使用 typedef 更改 float 資料型別名稱為 REAL 識別字。寫法如下:

```
typedef float REAL;
REAL a=7.0, b; /* 相當於 float a,b; */
```

P15-1 **的不同資料檔**

寫回指定的資料檔內。如此，資料不但容易維護，而且同一份程式可處理相同資料格式不同的資料檔，增加程式的使用彈性。

ffileno

P15-5 Fileno	語法：int fileno(FILE *) 功能：傳回所開啟檔案對應的檔案處理代碼(handle_no)。
-----------------	----------------------------------------------------------

P15-6

[例 1] 開啟一個和執行檔在相同路徑下 sample.txt 文字資料檔。

```
FILE *fptr;
fptr=fopen("sample.txt", "r");
```

[例 2] 開啟一個位於 C 磁碟 test 資料夾下的 sample.txt 文字資料檔。

```
FILE *fptr;
fptr=fopen("C:\\test\\sample.txt", "r"); //寫法 1
fptr=fopen("C:/test/sample.txt", "r"); //寫法 2
```

P15-26~P15~27

一. 選擇題

- 使用 fopen() 函式開檔失敗會傳回？
 - NULL
 - 1
 - 非 0 值
 - 檔案指標
- 使用 fclose() 函式關檔失敗會傳回？
 - NULL
 - 1
 - 非 0 值
 - 0
- ~~fread() 函式宣告於哪個標頭檔？~~

~~⊕ stdlib.h ⊕ stdio.h ⊕ math.h ⊕ io.h~~
- 一個資料檔的大小可透過什麼函式來取得？
 - filesize
 - filelen
 - filelength
 - lengthoffile
- fopen("檔名", "w+") 請問開啟檔案的模式？
 - 開啟已經存在的檔案
 - 開啟可以讀寫資料的檔案，若檔案存在，檔案內的資料會被刪除成一個空檔

③ 開啟二進位檔

④ 開啟可以讀寫資料的檔案，新增的資料會寫到檔案的最後面

~~6. 下面哪一個不是標準輸出入函式讀寫檔案內資料的方法？~~

~~⊕ 字元 ⊕ 字串 ⊕ 陣列 ⊕ 結構~~

5. 以結構化輸出入來存取資料檔，其檔案模式可以指定以下哪一個格式？

① w ② r ③ wb ④ 以上皆是

6. fileno 函式可用來取得？

① 檔案代碼 ② 檔案指標 ③ 開檔資訊 ④ 檔案大小