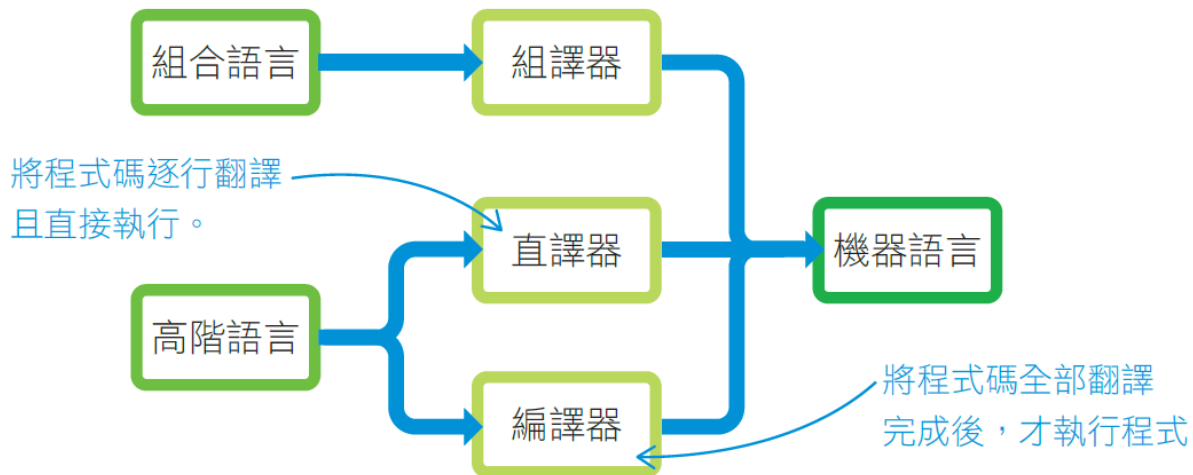


1. 不同程式語言產生可執行碼的過程



2. 流程圖符號

名稱	符號	說明
起始 / 結束符號		程式的起始 / 結束
資料符號		資料的輸入 / 輸出
動作符號		執行某個動作
決策符號		條件判斷用
流程符號		表示程序執行的方向
連接符號		程序連接用

3. 運算子的表示式

表示式	敘述	說明
中置式	$a \% b$	取得 a 除以 b 之後的餘數，運算子必須置於兩數之間
前置式	$++a$	遞增運算子，用來表示 $a = a + 1$
後置式	$a--$	遞減運算子，用來表示 $a = a - 1$
複合運算式	$a*=b$	運算子置於等號之前，用來表示 $a = a * b$
複合運算式	$a=b++$	先將 b 值指定給 a ，然後 b 再遞增
複合運算式	$a=++b$	b 值先遞增，再指定給 a

- $\text{pow}(2,31)=2^{31}$ ，自然指數 $e = 2.718281828459045$ ， $\text{exp}(1)=e^1$ ，
- 常見的關係運算子有等於(==)、不等於(!=)、大於(>)、小於(<)、大於等於(>=)、小於等於(<=)。
- 軟體發展生命週期將系統發展過程依序劃分為系統規劃、系統分析、系統設計、系統發展、系統建置等五個步驟

7. 資料型態

變數 v1 宣告為 int 整數型態、其值 = 2147483647

變數 v2 宣告為 short 短整數型態、其值 = 32767

變數 v3 宣告為 char 字元型態、其值 = A

變數 v4 宣告為 bool 布林型態、其值 = 1

變數 v5 宣告為 float 浮點數型態、其值 = 2.71828

變數 v6 宣告為 double 雙精密度浮點數型態、其值 = 2.718281828459045

- ❖ 變數 v1 為整數型態，占記憶體空間 4 Bytes (共 32bits)，其值的範圍介於 $-(2^{31}) \sim +(2^{31}-1)$ 之間。
- ❖ 變數 v2 為短整數型態，占記憶體空間 2 Bytes (共 16bits)，其值的範圍介於 $-(2^{15}) \sim +(2^{15}-1)$ 之間，也就是介於 $-32768 \sim +32767$ 之間。
- ❖ 變數 v3 為長整數型態，占記憶體空間 4 Bytes，與整數型態相同。
- ❖ 變數 v4 為字元型態，僅占記憶體空間 1 Byte，因此其值介於 $-128 \sim +127$ 之間。
- ❖ 變數 v5 為布林型態，僅占記憶體空間 1 Byte，其值為 0 或 1。
- ❖ 變數 v6 為浮點數型態，占記憶體空間 4 Bytes，其值的範圍為 $\pm 3.4 \times 10^{\pm 38}$ 。
- ❖ 變數 v7 為雙精密度浮點數型態，占記憶體空間 8 Bytes，其值的範圍為 $\pm 1.7 \times 10^{\pm 308}$ 。

8. 1 Byte 位元組 = 8 bits 位元

9. ASCII = American Standard Code for Information Interchange, 美國標準資訊交換碼

b ₇ b ₆ b ₅	000	001	010	011	100	101	110	111
b ₄ b ₃ b ₂ b ₁	0	1	2	3	4	5	6	7
0000	0 NUL	DLE	SP	0	@	P	`	p
0001	1 SOH	DC1	!	1	A	Q	a	q
0010	2 STX	DC2	"	2	B	R	b	r
0011	3 ETX	DC3	#	3	C	S	c	s
0100	4 EOT	DC4	\$	4	D	T	d	t
0101	5 ENQ	NAK	%	5	E	U	e	u
0110	6 ACK	SYN	&	6	F	V	f	v
0111	7 BEL	ETB	'	7	G	W	g	w
1000	8 BS	CAN	(	8	H	X	h	x
1001	9 HT	EM	)	9	I	Y	i	y
1010	A LF	SUB	*	:	J	Z	j	z
1011	B VT	ESC	+	;	K	[	k	{
1100	C FF	FS	,	<	L	\	l	
1101	D CR	GS	-	=	M	]	m	}
1110	E SO	RS	.	>	N	^	n	~
1111	F SI	US	/	?	O	_	o	DEL

優先順序	運算子
1	()
2	正號(+)、負號(-)、++、--、!、~、sizeof
3	*、/、%
4	加法(+)、減法(-)
5	<<、>>
6	>、<、>=、<=
7	==、!=
8	&
9	^
10	
11	&&
12	
13	=、+=、-=、\=、%=

10. 常見的算術運算子有加(+)、減(-)、乘(*)、除(/) 及餘數(%)。常用的位元運算子有且(AND &)、或(OR |)、互斥或(XOR ^)、補數(NOT ~)。其真值表及符號運算式如下：

AND 運算		
A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

OR 運算		
A	B	A B
0	0	0
0	1	1
1	0	1
1	1	1

XOR 運算		
A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

NOT 運算	
A	~A
0	1
1	0

11. 為隱藏式的轉換(Implicit Conversion)。其轉換順序如下表：

資料型態	優先權高低
long double	最高
double	次高
float	高
long	低
int	次低
char	最低

1. 程式架構包含前置處理指令 `#include`、主程式 `main()` 以及名稱空間 `using namespace std`。

2. 基本輸出函式 `cout` 的指令格式如下：

```
cout << 變數或字串 1 << 變數或字串 2...<< 變數或字串 n;
```

3. 基本輸入函式 `cin` 的指令格式如下：

```
cin >> 變數 1 >> 變數 2...>> 變數 n;
```

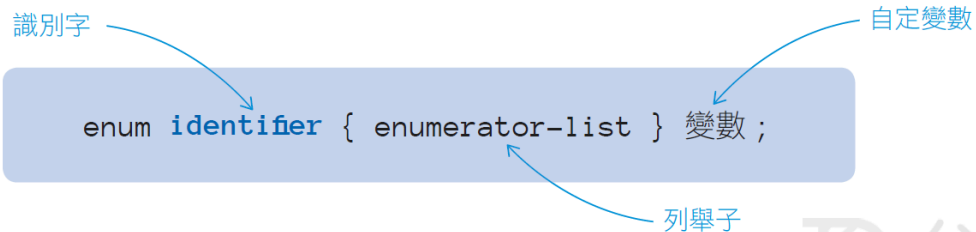
4. 變數的宣告格式如下：

```
資料型態 變數名稱1 = 初值, 變數名稱2 = 初值, ... ;
```

5. 常數的宣告以關鍵字 `const` 宣告，其值不可以被更改或重新設定。指令格式如下：

```
const 資料型態 常數名稱1 = 數值1, 常數名稱2 = 數值2, ... ;
```

12. 列舉資料型態是由一組具名整數常數所成的集合，指令格式如下：



13. 常見的流程指令格式整理如下：

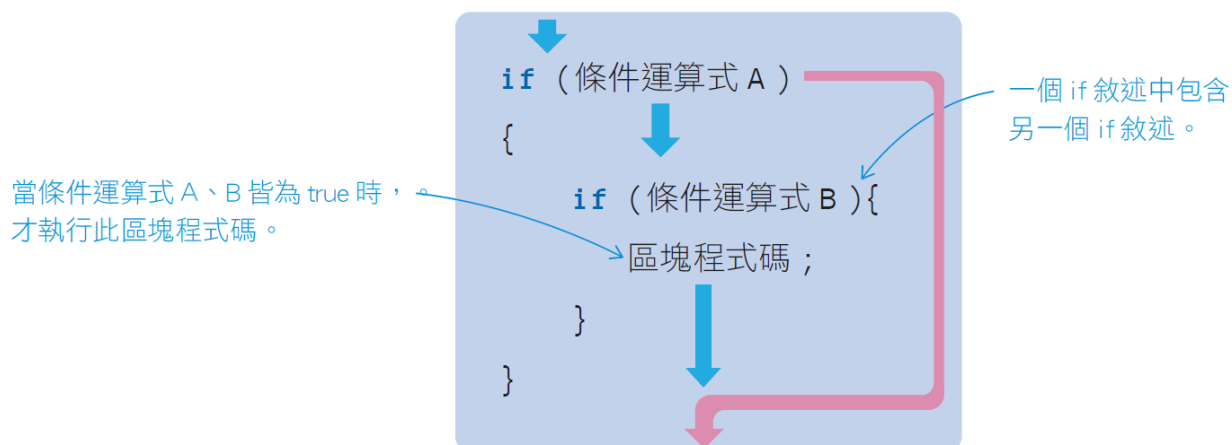
if 指令敘述

```
if ( 條件運算式 )
{
    敘述 _1;
    敘述 _2;
    ...
    敘述 _n;
}
```

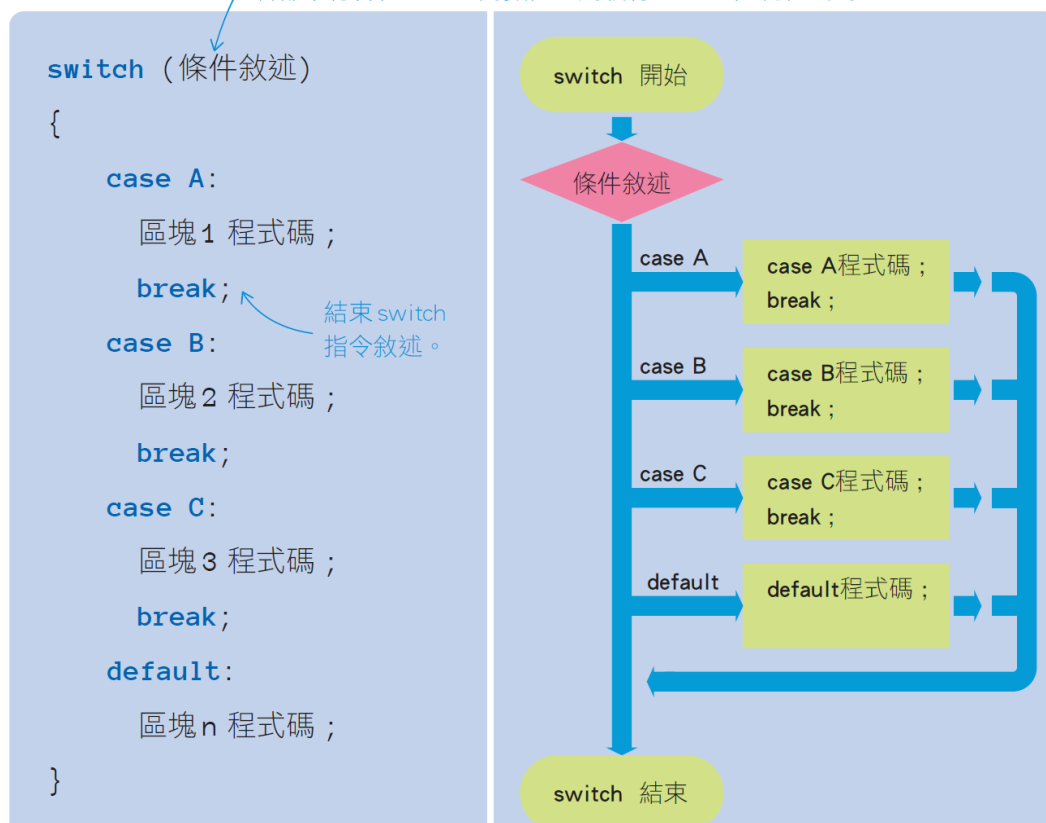
if-else 指令敘述

```
if ( 條件運算式 ) {
    區塊 A 程式碼;
}
else {
    區塊 B 程式碼;
}
```

當 if 指令敘述中包含另外一個 if 指令敘述時，我們稱之為巢狀結構。其格式如下：



當條件敘述的結果等於某個 case 的值，程式就執行該區塊的程式碼，若都不符合任一 case 的數值，則執行 default 區塊程式碼。



14. 常見的迴圈指令格式整理如下：

for 迴圈指令

```
for ( 初始運算式 ; 條件運算式 ; 控制運算式 )  
{  
    區塊程式碼 ;  
}
```

while 迴圈指令

```
while ( 條件運算式 )  
{  
    區塊程式碼 ;  
}
```

do-while 迴圈指令

```
do {  
    區塊程式碼 ;  
} while ( 條件運算式 );
```

```
1  #include<iostream>  
2  #include<iomanip>  
3  using std::cin;  
4  using std::cout;  
5  using std::endl;  
6  main()  
7  {  
8      int a,b,c,d;  
9      cout<<"請輸入第1個正整數:";  
10     cin>>a;  
11     cout<<"請輸入第2個正整數:";  
12     cin>>b;  
13     //for(c=a;c>=1;c--)    if (a % c == 0 and b % c == 0) break;  
14     c=a;  
15     while (c>=1){  
16         if (a % c == 0 and b % c == 0) break;  
17         c--;  
18     }  
19     cout<<"最大公因數GCD是"<<c<<endl;  
20  
21     //for(d=a;d<=a*b;d++)    if (d % a == 0 and d % b == 0) break;  
22     d=a;  
23     do{  
24         if (d % a == 0 and d % b == 0) break;  
25         d++;  
26     }while (d<=a*b);  
27     cout<<"最小公倍數LCM是"<<d<<endl;  
28     system("Pause");  
29 }
```

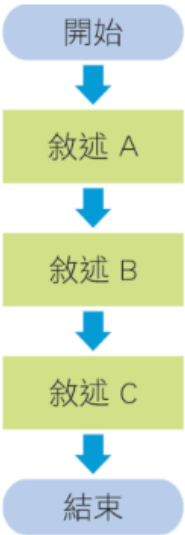
請輸入第1個正整數:12
請輸入第2個正整數:16
最大公因數GCD是4
最小公倍數LCM是48
請按任意鍵繼續 . . .

15. 維陣列與二維陣列的宣告格式如下：

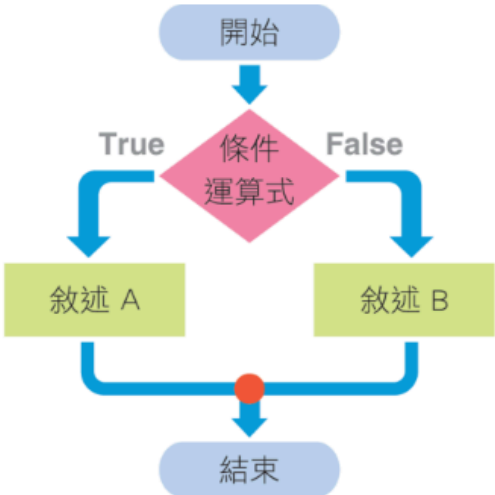
資料型態 變數名稱 [陣列大小] ; 資料型態 變數名稱 [陣列大小] [陣列大小] ;

16. 結構化

循序結構
程式必須依序由上而下逐一執行。



選擇結構
程式可以依條件判斷，選擇適當的路徑。



迴圈結構
程式可以搭配條件判斷，重複執行同一個區塊的程式碼。

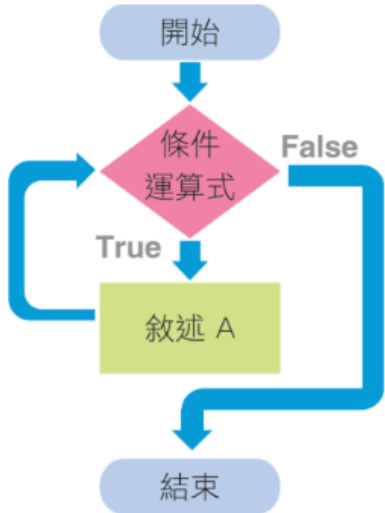
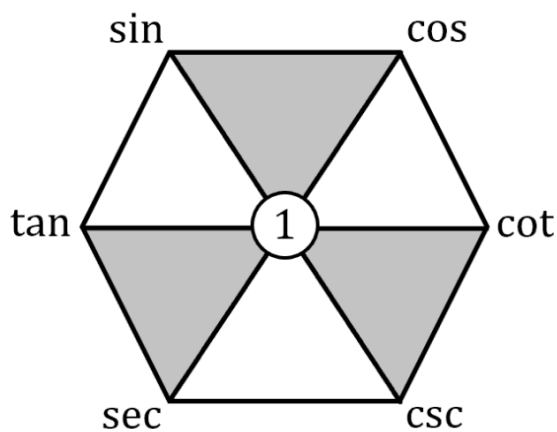


圖 2-4 結構化程式語言之三種基本結構

17. 微軟Office 包含 **Word**文書編輯、**Excel**試算表、**PowerPoint**簡報、**Outlook**郵件、**OneNote**數位筆記本、**Access**資料庫。開源的辦公室軟體OpenOffice，包括**Writer**文書編輯、**Impress**簡報、**Calc**試算表、**Base**資料庫、繪圖(**Draw**)、數學公式(**Math**)。
18. **Include**包含、**Input**輸入、**Output**輸出、**Stream**串流、**Main**主要、**Project**專案、**math**數學、**Std**=standard 標準、**Cout**=Char output 字元輸出、**Endl**=end of Line 行的結束(換行)、**Windows Application** 視窗應用程式、**Console Application**主控台應用程式、**System**系統

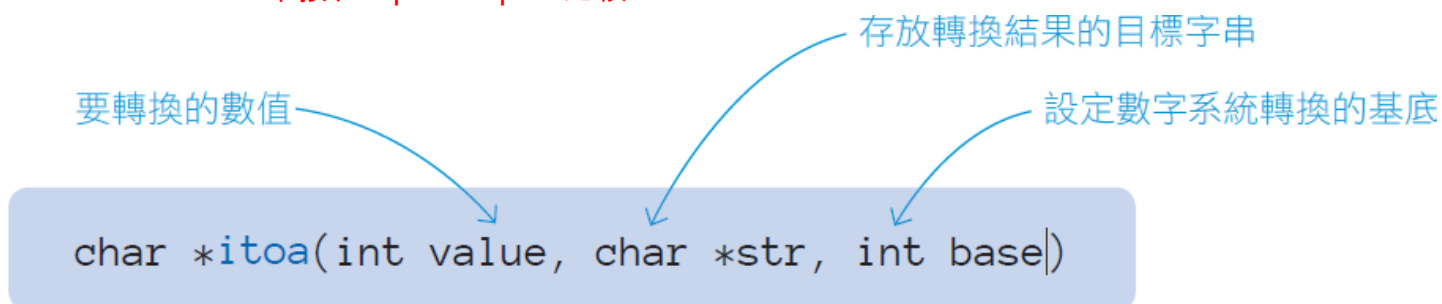
函 式	說 明
void srand (unsigned int seed)	初始化亂數產生函式
int rand (void)	亂數產生函式（介於 0~32767 的亂數值）
double ceil (double x)	將 x 值無條件進位
double floor (double x)	將 x 值無條件捨去
double round (double x)	將 x 值四捨五入
double abs (double x)	將 x 值取絕對值



函 式	
double sin (double x)	正弦函式
double cos (double x)	餘弦函式
double tan (double x)	正切函式
double asin (double x)	反正弦函式
double acos (double x)	反餘弦函式
double atan (double x)	反正切函式

函 式	說 明
size_t strlen (char *str)	取得字串的長度
char * strcpy (char *destination, char *source)	複製字串
char * strncpy (char *destination, char *source, size_t num)	複製部分字串
char * strcat (char *destination, char *source)	字串的串接
char * strncat (char *destination, char *source, size_t num)	串接部分字串
int strcmp (char *str1, char *str2)	• 當 str1=str2 時回傳 0 - 字串比較：• 當 str1>str2 時回傳 1 • 當 str1<str2 時回傳 -1
char * itoa (int value, char *str, int base)	將整數轉換成字串 註：該函式定義於<stdlib>標頭檔中

size大小、str=string字串、len=length長度、cpy=copy複製、destination目的、source來源、
cat= concatenation串接、cmp= compare比較



19. 要會用人腦計算，統測考型

行	程式碼
1	#include <iostream>
2	#include <iomanip>
3	using std::cout;
4	using std::endl;
5	
6	main()
7	{
8	int i,sum,v1,v2,v3,v4,v5;
9	
10	v1=3;sum=0;
11	v2=v1++;
12	v3=++v2;
13	v4=v3--;
14	v5=--v4;
15	for(i=0;i<10;i++)
16	sum=sum+i;
17	system("Pause");
18	}
19	

請以偵錯模式逐行執行，並追蹤變數的數值，完成下表。

第 11 行執行完畢，v1 = 4、v2 = 3。

第 12 行執行完畢，v2 = 4、v3 = 4。

第 13 行執行完畢，v3 = 3、v4 = 4。

第 14 行執行完畢，v4 = 3、v5 = 3。

當 sum=10 時，i = 4。

當 sum=36 時，i = 8。

程式執行完畢後，sum = 45、i = 10。

1 工場安全衛生及程式應用

1. 工業安全標示的意義

形狀	正三角形 △	倒三角形 ▽	方形 □	圓形 ○
危險程度	第二級	第三級	×	第一級
標示	警告	注意	說明	禁止

顏色	紅色 ■	橙色 ■	黃色 ■	綠色 ■	藍色 ■	紫色 ■
標示	危險	警告	注意	安全	禁止	放射危險
	消防設備			安全急救設備、 逃難疏散指示。	禁止使用待修機具。	

2. 火災類型與適用滅火器

分類	名稱	引起原因	可用滅火器
A (甲) 類	普通火災	紙張、木材等可燃性固體。	水、泡沫、乾粉
B (乙) 類	油類火災	石油、有機溶劑等可燃性液體。	泡沫、二氧化碳、乾粉
C (丙) 類	電氣火災	電器配線及電器設備。	二氧化碳、乾粉
D (丁) 類	金屬火災	鈉、鉀、鎂等可燃性金屬。	金屬

3. 資訊安全

- (1) 資訊安全是為了保護資訊的機密性、完整性、可用性。
- (2) 正確的資安觀念可以避免資料於未授權的情況下被非法存取、竄改、破壞。
- (3) 資訊安全 4D: 嚇阻 (Deter)、偵測 (Detect)、禁制 (Deny)、延遲 (Delay)。

4. 正確的資安觀念

- (1) 使用正版軟體。
- (2) 隨時更新軟體的修補程式。
- (3) 安裝防毒軟體並隨時更新病毒碼。
- (4) 養成備份的好習慣。
- (5) 勿點擊來路不明的連結。

2 程式架構

1. 軟體分類



2. 程式語言發展

- (1) 機器語言→組合語言→高階語言。
- (2) 機器語言由 0、1 所構成。
- (3) 組合語言使用助憶碼提高程式的可讀性。
- (4) 不同的硬體使用不同的組合語言。
- (5) 高階語言具有跨平台的特性。

3. 語言處理程式

- (1) 組譯程式 (Assembler)：將組合語言轉為機器碼。
- (2) 編譯程式 (Compiler)：將高階語言轉為機器碼。
- (3) 直譯程式 (Interpreter)：將高階語言轉為機器碼。

4. 專案開發

開發流程：

5. 程式架構

- (1) 程式區塊分為：前置處理器、全域變數宣告區、函式原型宣告區、主程式區、自訂函式區。
- (2) 前置處理器：編譯器會在編譯程式之前就先處理的前置作業。
- (3) 全域變數宣告區：當變數是所有函式共用時，則應該宣告於此區域中。
- (4) 函式原型宣告區：函式在使用之前必須先定義。

6. 除錯 (Debug)

- (1) 除錯就是找到程式錯誤的地方，修正為正確答案的過程。
- (2) 語法錯誤：使用錯誤的語法，編譯器就會無法編譯而產生錯誤。
- (3) 語意錯誤：語意錯誤又稱之為邏輯錯誤，程式可以執行但是結果錯誤。

3 變數與常數

1. 結構化程式

- (1) 結構化程式的邏輯性是**由上而下**之架構。
- (2) 結構化程式的三種基本結構為：循序結構、選擇結構、重複結構。
- (3) 每一種結構都只有**單一入口**和**單一出口**。
- (4) 程式應避免使用 **GOTO** 指令，避免破壞程式的可讀性及結構性。

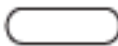
2. 演算法 (Algorithm)

- (1) 演算法的定義：能在有限的時間內完成明確界定的演算過程。
- (2) 演算法可以使用文字、流程圖、虛擬碼等方式敘述。
- (3) 演算法的五個特點：輸入、輸出、有限性、明確性、有效性。

3. 虛擬碼 (Pseudo Code)

- (1) 虛擬碼是將演算法以熟悉**的文字**或部分**程式碼**呈現。
- (2) 虛擬碼強調的是**程式的邏輯思維**，不受限於某一種程式語言的框架。

4. 常用的流程圖符號

名稱	圖示	說明	名稱	圖示	說明
起始／結束符號		程式的開始或結束	決策符號		條件判斷用
資料符號		程式的輸入或輸出	流程符號		表示程序執行的方向
動作符號		執行某個動作	連接符號		程序連接用

5. 變數 (Variables)

- (1) 變數是以一個名稱來代表記憶體的某個位址。
- (2) 變數宣告前必須先宣告資料型態。
- (3) 變數必須宣告後才能夠使用。
- (4) 變數只有兩種操作動作：存入數值、讀取數值。
- (5) 如果要將變數宣告為常數變數，使用 `#define` 或 `const` 關鍵字宣告。

6. 識別字 (identifiers)

- (1) 變數的命名又稱為識別字。
- (2) 變數的命名只能使用英文字母、數字及底線命名。
- (3) 數字不可以放在名稱的第一個位置。

7. 關鍵字 (keywords)

- (1) 關鍵字又稱保留字，是程式語言已事先賦予的識別字。
- (2) 關鍵字有其特殊意義，使用者不能再將這些字作為變數名稱使用。

4 資料型態

1. 資料型態整理

資料型態	名稱	儲存空間	數值範圍
bool	布林	1 Byte	true 或 false
char	字元	1 Byte	-128 ~ 127
short	短整數	2 Bytes	-32768 ~ 32767
unsigned short	無號數短整數	2 Bytes	0 ~ 65535
int	整數	4 Bytes	-2147483648 ~ 2147483647
unsigned int	無號數整數	4 Bytes	0 ~ 4294967295
long	長整數	4 Bytes	-2147483648 ~ 2147483647
float	浮點數	4 Bytes	$\pm 3.4 \times 10^{\pm 38}$
double	雙精密度浮點數	8 Bytes	$\pm 1.7 \times 10^{\pm 308}$

2. 資料型態轉換

- (1) 隱藏式轉換：編譯器會自動將運算元轉換成相同的資料型態再做運算。
- (2) 明確式轉換：在變數前面加上一對小括號，中間填入要轉換的資料型態，就可以進行資料型態的轉換。

3. 列舉 (enumeration)

- (1) 列舉為使用者定義的列舉清單。
- (2) 列舉是由一組具名整數常數所成的集合。
- (3) 集合內的元素都是已知且具名的整數識別字。
- (4) 整數識別字預設值為 0，依序往後編碼。
- (5) 內容後面也可以加上指定對應的整數值。

5 運算式及運算子

1. 運算式

- (1) 運算子 (Operator)：用來運算的符號。
- (2) 運算元 (Operand)：常數或變數。
- (3) 運算式 (Expression)：由運算元與運算子組成的式子。

2. 指定運算子

- (1) 指定運算子(=)是將運算式的結果指定給變數。
- (2) 指定運算子的左邊一定要是變數。
- (3) 指定運算子可以連續指定。

3. 算術運算子

- (1) 常見的算術運算子有：加(+)、減(-)、乘(*)、除(/)、取餘數(%)。
- (2) 運算優先權依序為：正負號→乘、除、餘數→加、減。

4. 遞增(減)運算子

- (1) 遞增運算子(++)可以將變數的數值加1。
- (2) 遞減運算子(--)可以將變數的數值減1。
- (3) 前置(prefix)或後置(postfix)結果不一樣。

5. 複合指定運算子

- (1) 同時執行運算及指定兩個動作。
- (2) 複合指定運算子的優先權低於算術運算子。

6. 邏輯運算子

- (1) 常見的邏輯運算子有：AND(&&)、OR(||)、NOT(!)。
- (2) 邏輯運算的結果只有true(真)與false(假)兩種結果。
- (3) 邏輯運算子採用短路計算法(Short-circuit evaluation)。

7. 位元運算子

- (1) 常見的位元運算子有：AND(&)、OR(|)、XOR(^)、補數(~)。
- (2) 常見的移位運算子有：左移(<<)、右移(>>)。

8. 關係運算子

- (1) 用來判斷兩個數字之間的關係。
- (2) 常見的關係運算子有：等於(==)、不等於(!=)、大於(>)、大於等於(>=)、小於(<)、小於等於(<=)。
- (3) 關係運算子的結果只有true(真)與false(假)兩種結果。

6 流程指令及迴圈

1. 結構化程式

結構化的三種基本結構分別為：循序結構、選擇結構、重複結構。

循序結構	選擇結構	重複結構
由上而下依序執行。	根據條件的成立與否，決定執行哪個程式區塊。	根據條件的成立與否，決定程式的執行次數。

2. 流程指令

單一選擇	重複結構
<pre>if(條件式) { 條件成立的程式敘述; }</pre>	<pre>if(條件式){ 條件成立的程式敘述; } else{ 條件不成立的程式敘述; }</pre>

2. 指定運算子

- (1) 指定運算子(=)是將運算式的結果指定給變數。
- (2) 指定運算子的左邊一定要是變數。
- (3) 指定運算子可以連續指定。

3. 算術運算子

- (1) 常見的算術運算子有：加(+)、減(-)、乘(*)、除(/)、取餘數(%)。
- (2) 運算優先權依序為：正負號→乘、除、餘數→加、減。

4. 遞增(減)運算子

- (1) 遞增運算子(++)可以將變數的數值加1。
- (2) 遞減運算子(--)可以將變數的數值減1。
- (3) 前置(prefix)或後置(postfix)結果不一樣。

5. 複合指定運算子

- (1) 同時執行運算及指定兩個動作。
- (2) 複合指定運算子的優先權低於算術運算子。

6. 邏輯運算子

- (1) 常見的邏輯運算子有：AND(&&)、OR(||)、NOT(!)。
- (2) 邏輯運算的結果只有true(真)與false(假)兩種結果。
- (3) 邏輯運算子採用短路計算法(Short-circuit evaluation)。

7. 位元運算子

- (1) 常見的位元運算子有：AND(&)、OR(|)、XOR(^)、補數(~)。
- (2) 常見的移位運算子有：左移(<<)、右移(>>)。

8. 關係運算子

- (1) 用來判斷兩個數字之間的關係。
- (2) 常見的關係運算子有：等於(==)、不等於(!=)、大於(>)、大於等於(>=)、小於(<)、小於等於(<=)。
- (3) 關係運算子的結果只有true(真)與false(假)兩種結果。

6 流程指令及迴圈

1. 結構化程式

結構化的三種基本結構分別為：循序結構、選擇結構、重複結構。

循序結構	選擇結構	重複結構
由上而下依序執行。	根據條件的成立與否，決定執行哪個程式區塊。	根據條件的成立與否，決定程式的執行次數。

2. 流程指令

單一選擇	重複結構
<pre>if(條件式) { 條件成立的程式敘述; }</pre>	<pre>if(條件式){ 條件成立的程式敘述; } else{ 條件不成立的程式敘述; }</pre>

多重選擇	switch 敘述
<pre> if(條件式1){ 條件式1成立的程式敘述; } else if(條件式2){ 條件式2成立的程式敘述; } else{ 當所有條件都不成立 的程式敘述; } </pre>	<pre> switch(變數名稱或運算式){ case 符合的數字或字元: 執行的敘述; break; case 符合的數字或字元: 執行的敘述; break; case 符合的數字或字元: 執行的敘述; break; default: 執行的敘述; } </pre>

3. 迴圈指令

(1) for 迴圈

- ① for 迴圈屬於前測試迴圈（先判斷再執行）。
- ② for 迴圈一般應用於明確次數的迴圈設計。
- ③ for 迴圈的參數設定有變數初始設定、結束條件式、調整控制變數。

```

for(變數初始設定;條件式;調整控制變數)
{
    程式敘述;
}

```

(2) while 迴圈

- ① while 迴圈屬於前測試迴圈（先判斷再執行）。
- ② while 迴圈適用於明確次數與不明確次數的迴圈設計。

```

while(條件式)
{
    程式敘述;
}

```

(3) do-while 迴圈

- ① do-while 迴圈屬於後測試迴圈（先執行再判斷）。
- ② do-while 迴圈的程式至少執行一次。
- ③ do-while 迴圈適用於明確次數與不明確次數的迴圈設計。

```

do
{
    程式敘述;
}while(條件式);

```

7 陣列及指標

1. 陣列 (Array)

- (1) 陣列是一群**相同資料型態**元素的集合。
- (2) 陣列元素**共用一個變數名稱**，而每一個元素都有一個對應的**索引值 (index)**。
- (3) 陣列內的元素，在記憶體空間中的位址是**連續**的。
- (4) 陣列占用的記憶體空間大小 = (元素個數) × (資料型態的所占位元數)。
- (5) 一維陣列 $A[m]$ ，假設起始位址為 K 、每個元素占 S 個記憶體空間大小。則 $A[y]$ 的記憶體位址為： $K+y*S$ 。
- (6) 二維陣列 $A[m][n]$ ，假設起始位址為 K 、每個元素占 S 個記憶體空間大小。則 $A[x][y]$ 的記憶體位址為： $K+(x*n+y)*S$ 。

2. 指標 (Pointer)

- (1) 指標是一種變數，儲存的是**記憶體位址**。
- (2) 指標的資料型態，必須與指向的變數資料型態相同。
- (3) 取址運算子 (&)：可以將變數的**記憶體位址**取出。
- (4) 指位運算子 (*)：可以**使用指標間接存取變數的值**。

3. 陣列與指標

- (1) **陣列名稱**就是陣列**第一個元素的位址**。
- (2) 指標可以搭配**算術運算子 (+、-)**，間接存取陣列的每一個元素。
- (3) 可以利用指標進行陣列的**動態配置**。
- (4) **陣列指標 (Pointer-to-Array)**：指向陣列的指標。
- (5) **指標陣列 (Array-of-Pointer)**：由指標構成的陣列。

4. 陣列應用：資料搜尋

- (1) 「搜尋」就是在一堆資料中找出所要的資料 (**鍵值 key value**)。
- (2) 搜尋的核心動作為「比對」，當資料筆數越多時，搜尋的效率越低、時間越久。
- (3) 常見的搜尋演算法有：**循序搜尋法**、**二分搜尋法**、**插補搜尋法**等。

5. 陣列應用：資料排序

- (1) 排序是將一堆**無序**的資料重新整理成一個**有序**的序列。
- (2) 依照鍵值 (key value) 可以進行**由大到小**或**由小到大的**排序。
- (3) 常見的搜尋演算法有：**插入排序法**、**氣泡排序法**、**選擇排序法**等。

8 公用函式及函式

1. 函式

- (1) 將程式獨立成一個模組，稱之為**函式**。
- (2) 函式的優點：**提高程式碼再利用率**、**模組化設計**、**容易維護**、**容易移植**、**提高程式可讀性**、**縮短程式開發時間**。

2. 常見的公用函式：(1) 亂數函式。(2) 整數求解函式。(3) 三角函式。(4) 字串函式。

3. 自訂函式

- (1) 自訂函式的架構如下。

```

回傳值的資料型態 函式名稱(參數){
    函式本體;
}
    
```


- (2) 函式執行後，是使用關鍵字 `return` 將值傳回。
- (3) 不需要回傳值時，則回傳值資料型態應宣告為 `void`。
- (4) 在 C 語言中，函式的呼叫只要直接輸入函式名稱即可使用。

4. 函式原型

- (1) 函式在使用之前必須先定義。
- (2) 習慣上會將主程式寫在前面，自訂函式寫在後面。
- (3) 為了讓編譯器正確執行，需宣告函式原型，才能將自訂函式移到主程式之後。

5. 變數的生命周期

- (1) 變數宣告在主程式及函式之內時，稱之為區域變數 (Local Variable)。
- (2) 當變數宣告於主程式及函式之外時，稱之為全域變數 (Global Variable)。
- (3) 區域變數：僅能於函式內存取，當函式執行完畢後，變數立即失去功能。
- (4) 全域變數：所有的函式都可以使用該變數，直到主程式結束為止。

6. 引數的傳遞方式

- (1) 傳值呼叫：將變數的數值傳遞出去。主程式與函式對應到同一個變數值。
- (2) 傳址呼叫：透過指標將位址傳遞出去。主程式與函式指到同一個記憶體位址。
- (3) 傳參考呼叫：將位址傳遞出去。主程式與函式是同一個記憶體位址。

7. 遞迴函式

- (1) 當函式自己呼叫自己的時候，稱該函式為遞迴函式。
- (2) 遞迴必須滿足兩個條件：
 - ① 函式每經過一次遞迴處理，都必須將問題的範圍縮小。
 - ② 函式必須有一個終止條件以結束遞迴。

9 結構及類別

1. 結構

- (1) 結構是使用者自訂型態，以關鍵字 `struct` 定義。
- (2) 結構能夠結合多個不同資料型態的變數 (成員資料) 在一個名稱之下。
- (3) 結構的成員資料預設是公用的 (`public`)。
- (4) 如果要存取成員資料，應使用成員存取運算子 (`.`) 進行存取。

2. 結構與陣列

- (1) 如果要宣告多筆相同屬性的結構變數，可以使用陣列宣告。
- (2) 宣告陣列的同時，可以同時設定初始值。
- (3) 陣列名稱相當於指向陣列第一個結構的指標。

3. 結構與指標

- (1) 可以利用結構的型態來宣告指標變數 (指向結構)。
- (2) 陣列名稱相當於指向陣列第一個結構的指標。
- (3) 指標的資料型態，必須與指向的結構變數資料型態相同。
- (4) 取址運算子 (`&`)：可以將結構變數的記憶體位址取出。
- (5) 指位運算子 (`*`)：可以間接存取結構變數。
- (6) 通過指標存取成員運算子 (`->`)：又稱結構反參考運算子，取得指標所指向的結構的某個變數。

4. 結構與函式

- (1) 與基本變數型態相同，我們可以將結構變數當成引數傳至函式。
- (2) 引述傳遞的方式同樣有傳值、傳位址與傳參考等三種。

5. 類別

- (1) 類別是一種使用者自訂型態，以關鍵字 `class` 定義。
- (2) 類別是將不同的成員變數與函式組織在同一個模組的設計方法。
- (3) 類別的資料成員預設是私有的 (`private`)。
- (4) 如果要存取公有成員的資料，應使用成員存取運算子 (`·`)。
- (5) 如果要存取私有成員的資料，必須透過自訂函式的呼叫才能完成存取動作。
- (6) 物件是類別實體化的結果，而類別是抽象的定義。
- (7) 當函式只在類別內宣告函式原型，而函式本體定義在類別之外時，需以類別名稱及範圍解析運算子 (`::`) 讓編譯器知道類別與成員函式的對應關係。

6. 建構子

- (1) 建構子可以用來進行物件初始化；當物件建立後，程式會自動執行建構子。
- (2) 建構子的名稱與類別名稱需相同。
- (3) 一個類別可以有多個建構子 (建構子多載)。
- (4) 建構子無傳回值。

7. 解構子

- (1) 當物件不再使用時，可以使用解構子釋放物件的記憶體空間。
- (2) 解構子的名稱與類別名稱相同，名稱前要加上波浪符號 (`~`)。
- (3) 建構子無傳回值。

8. 物件導向程式設計

- (1) 物件導向比結構化程式設計更符合人類的思維模式，將日常生活中物品運用的方法 (Method)、屬性 (Property) 及事件觀念運用在程式設計中。
- (2) 物件的三個要素：描述物體所擁有的特性 (屬性)、行為 (方法) 以及事件。
- (3) 物件導向的三個要素：封裝、繼承、多型。
- (4) 物件導向的優點有：容易維護、容易擴充、再利用率高、高靈活度的特性。
- (5) 物件是類別實體化的結果，而類別是抽象的定義。
- (6) 物件導向中，應先定義類別，才能使用物件。
- (7) 存取修飾詞是用來設定類別的存取等級。
 - ① 公用型態 (`public`)：沒有存取限制。
 - ② 私有型態 (`private`)：只能在自己的類別中存取。
 - ③ 保護型態 (`protected`)：只能在自己類別與子類別中存取。
- (8) 常見的物件導向程式語言有 `C++`、`C#`、`Java`、`Python` 等。

9. 堆疊 (Stack)

- (1) 堆疊是一種先進後出 (FILO) 的資料結構。
- (2) 堆疊只有一個出入口，所有動作都在同一端 (`top`) 進行。
- (3) 堆疊的動作有：加入 (`push`)、取出 (`pop`)、頭端 (`top`)、判斷是否為空堆疊 (`isempty`)、取得堆疊大小 (`getsize`)。

10. 佇列 (Queue)

- (1) 佇列是一種先進先出 (FIFO) 的資料結構。
- (2) 佇列有一個出口 (`front`)、一個入口 (`rear`)。
- (3) 佇列的動作有：加入 (`enqueue`)、取出 (`dequeue`)、判斷是否為空佇列 (`isempty`)、判斷是否為滿佇列 (`isfull`)。

類	名稱	引起原因	較適當的滅火原理	可用之滅火器
A (甲)	普通火災	紙張、木材等可燃性固體	水冷卻、降低燃燒溫度	水 泡沫 乾粉 ABC
B (乙)	油類火災	石油、有機溶劑等可燃性液體	覆蓋缺氧氣、使火窒息	泡沫 二氧化碳 乾粉 ABC
C (丙)	電器火災	電器配線及電器設備	斷電後，看情況做 A 或 B 類火災方式處理	二氧化碳 乾粉 ABC
D (丁)	金屬火災	鈉、鉀、鎂等可燃性金屬	燃燒溫度高，需要特種金屬化學乾粉	乾粉 D



35. 關於實習工場安全與衛生的敘述，下列何者錯誤？
(A) 通電中的變壓器起火燃燒，可以使用二氧化碳滅火器來撲滅
(B) 實習工場的消毒酒精起火燃燒，此為B類火災
(C) 燒燙傷急救的實施步驟依序為「沖、脫、泡、蓋、送」
(D) 心肺復甦術的實施步驟依序為「叫、叫、A、B、C、D」
36. 以C語言開發程式的敘述，下列何者錯誤？
(A) 程式需要經過編譯及連結產生可執行檔，才能夠執行
(B) 使用#include前置處理命令時，命令結尾需要加上分號
(C) main()是一個函式，程式執行時從main()函式開始執行
(D) 單行註解可用2個斜線(//)開頭
37. 關於C程式語言中，使用define建立常數的方式，下列何者正確？
(A) define PI=3.14; (B) define PI 3.14;
(C) #define PI=3.14 (D) #define PI 3.14
38. 關於C程式語言的資料型態，下列敘述何者錯誤？
(A) float資料型態可以儲存浮點數，數值精確度跟double資料型態相同
(B) 宣告int資料型態可以儲存整數資料
(C) double資料型態可以儲存浮點數值
(D) 宣告char資料型態可以儲存字元符號
39. 下列C語言程式碼片段執行後，x與y的結果為何？
int x, a=7, b=2;
float y;
x=a/b;
y=(float)a/b;
(A) x為3，y為3 (B) x為3，y為3.5 (C) x為3.5，y為3 (D) x為3.5，y為3.5
40. 下列C語言程式碼執行後，其結果為何？
#include <stdio.h>
int main()
{
int a=9, b=7;
printf("%d", a^b);
return 0;
}
(A) 1 (B) 2 (C) 14 (D) 15
41. 依C語言之運算子優先權順序，下列運算式的結果，何者為真(true)？
(A) !(1!=3)||1==3 (B) 1!=3&&!(1==3)
(C) !(1<3)||1>=3 (D) 1<3&&!(1>=3)
42. 下列C語言程式碼片段執行後，變數y的值為何？
int y, a=45;
if(a>=60)
y=a+1;
else if(a>=50)
y=a+2;
else
y=a+3;
(A) 45 (B) 46 (C) 47 (D) 48

47. 下列C語言程式碼片段執行後，變數result的值為何？

```
int a=5, b=3;
double result=0;
result=Func(a, b);
```

```
double Func(int w, int h)
{
    double A;
    A=w*h/2.0;
    return A;
}
```

- (A) 0.0 (B) 7.0 (C) 7.5 (D) 15.0

48. 若要利用C語言寫一個BMI函式，此一函式接收傳入兩個整數資料，經計算後回傳的數值必須有小數點後至少兩位數精確度，BMI函式的原型宣告應為下列何者？

- (A) double BMI(); (B) void BMI(int, int);
(C) int BMI(int, int); (D) float BMI(int, int);

49. 關於物件導向程式語言的敘述，下列何者錯誤？

- (A) 物件導向程式語言具有「多型」、「繼承」與「封裝」的特性
(B) 物件導向程式語言的「多型」可以達到資訊隱藏的目的
(C) 「繼承」的特性讓物件導向程式語言具有避免重複撰寫相同程式碼的優點
(D) 物件導向程式語言可以簡化大型程式的開發流程

50. 關於C語言中結構(structures)資料型態的敘述，下列何者錯誤？

- (A) 結構是一種自行定義的資料型態
(B) 宣告結構資料型態的變數時，只能使用已定義的結構名稱宣告
(C) 定義結構只能使用不同資料型態的變數
(D) 定義結構只能使用struct關鍵字

題號	答案	題號	答案	題號	答案	題號	答案	題號	答案
1	D	11	B	21	C	31	D	41	D
2	D	12	C	22	A	32	C	42	D
3	C	13	B	23	C	33	A	43	C
4	B	14	A	24	A	34	A	44	D
5	C	15	D	25	A	35	D	45	C
6	B	16	B	26	A	36	B	46	D
7	D	17	D	27	A	37	D	47	C
8	D	18	B	28	C	38	A	48	D
9	B	19	A	29	A	39	B	49	B
10	A	20	C	30	B	40	C	50	C

35. 有關電腦教室的安全注意事項，下列敘述何者正確？
 (A) 電腦教室內發生火災，應立即離開教室但不用通報學校環安室或老師
 (B) 教室內發現同學因觸電倒地，應立即對他進行 CPR 急救
 (C) AED 裝置是急救的電擊設備，用於 CPR(叫叫 CABD)中的 D 步驟
 (D) 同學在教室內打翻可燃性塗料引起火災，應立即潑水灌救
36. 曉華寫了下列一段 C 語言程式，想要測試程式執行時如何透過作業系統的終端機(Console)指令取得參數(Arguments)，但發現無法成功進行編譯，應採取下列哪一個方案來解決這個問題？

```

1
2  #include <stdio.h>
3  //void sub(int i, char *s);
4  int main(int argc, char *argv[]) {
5      sub(argc, argv[2]);
6      return 0;
7  }
8
9  void sub(int i, char *s){
10     printf("total %d arguments, and the 2nd one is %s\n", i, s);
11 }

```

- (A) 將行號4中main(int argc, char *argv[])改為main()
 (B) 去掉行號3最前面的註解標記//
 (C) 將行號1的空白行刪除
 (D) 在行號1新增#include <stdlib.h>
37. 關於C語言中語法錯誤(Syntax error)以及程式執行過程的錯誤(Run-time error)的敘述，下列何者正確？
 (A) 語法錯誤在編譯過程中就會被編譯器發現
 (B) 除以0屬於一種語法錯誤
 (C) 指標未經初始化就指定其內容時可能會發生語法錯誤
 (D) 在敘述(Statement)結束時，沒有分號是一種程式執行過程的錯誤

▲閱讀下文，回答第 38-39 題

小芳在一個原本可以編譯(Compile)成功的程式中，在 main()主程式內再加入行號 1 至行號 6 的程式碼，但加入後發生編譯錯誤的情況。

```

1  #define Value1 100
2  #define Value2 (Value1 - 1)
3  const int Value3;
4  int CheckValue = 0;
5  Value3 = Value2;
6  CheckValue = Value1 + Value3;

```

38. 小芳刪除行號 1 至行號 5 中的哪一個部分後，可以使程式編譯成功？
 (A) (Value1 - 1) (B) Value3 = Value2;
 (C) const (D) #define Value2 (Value1 - 1)
39. 程式修正後，當程式執行完行號 6 的時候，CheckValue 的值為下列何者？
 (A) 200 (B) 199 (C) 198 (D) 100

曉華想要了解 C 語言程式區域變數 (Local variable) 和全域變數 (Global variable) 的數值變化情形，撰寫了下列的程式。

```

1  #include <stdio.h>
2  int sum=1, x=10;
3  int inc(int xin){
4      int sum=2;
5      sum = sum + xin;
6      xin++;
7      return (sum);
8  }
9  int main(){
10     int sum = 3;
11     sum=inc(x);
12     printf("%d, %d", sum, x);
13     return 0;
14 }

```

40. 下列何者為程式執行結果？
(A) 11, 11 (B) 13, 10 (C) 12, 11 (D) 12, 10
41. 在執行到行號 12 的時候，想要讓 x 的值隨著行號 6 中 xin 的值更新，下列修改程式的方式何者正確？
(A) 行號 11 的 x 改為 &x，並將函式 inc() 中所有的 xin 全部改為 *xin
(B) 行號 11 的 x 改為 *x，並將函式 inc() 中所有的 xin 全部改為 &xin
(C) 行號 11 的 x 改為 &x，並將函式 inc() 中所有的 xin 全部改為 &xin
(D) 行號 11 的 x 改為 *x，並將函式 inc() 中所有的 xin 全部改為 *xin
42. 關於行號 2、行號 4、以及行號 10 的變數 sum 的敘述，下列何者正確？
(A) 行號 2 的 sum 是全域變數，行號 4 的 sum 是區域變數
(B) 行號 2 的 sum 是區域變數，行號 4 的 sum 是全域變數
(C) 行號 2 的 sum 和行號 10 的 sum 都是區域變數
(D) 行號 2 的 sum 和行號 10 的 sum 都是全域變數
43. 阿華想要了解 C 語言程式中不同資料型態佔據記憶體空間的大小，於是使用 C 語言中的 sizeof() 函式並撰寫如下程式來測試，所使用的電腦為 64 位元的個人電腦，下列何者為該程式的執行結果？

```
1  #include <stdio.h>
2
3  int main() {
4      char *s="ABC";
5      int i=10,CharPtrSize,CharSize;
6      CharPtrSize = (int)sizeof(s);
7      CharSize = (int)sizeof(*s);
8      printf("%d, %c", CharSize, *(s+2));
9
10     return 0;
11 }
```

- (A) 8, B (B) B, 1 (C) 1, C (D) C, 8

44. 執行完下列片段程式後，Num1 與 Num2 的數值分別為何？

```
1  int Num1 =10, Num2 = 5;
2  int Num3 =3;
3  Num1 = Num1 << Num3-1;
4  Num2 = Num2 * Num1 >> 1;
```

(A) Num1=79、Num2=197

(B) Num1=79、Num2=195

(C) Num1=40、Num2=200

(D) Num1=40、Num2=100

45. 阿華想要了解C語言程式if條件敘述中常用的運算子&與&&的不同，撰寫如下程式，下列何者為程式執行結果？

```
1  #include <stdio.h>
2
3  int main() {
4
5      int a=0x0a;
6      int b=0x05;
7
8      if(a & b)
9          printf("a&b=%d\n", a&b);
10     else
11         printf("a&&b=%d\n", a&&b);
12
13     return 0;
14 }
```

(A) a&&b=1

(B) a&&b=0

(C) a&b=1

(D) a&b=0

46. 曉華想要知道三角函數 $\sin(x)$ 在 $x=0$ 之後遞增的變化情形，寫了如下的C語言程式碼，卻發現迴圈內行號8和行號9的程式碼只執行了一次，下列哪一種修改程式的方式可以讓迴圈內的程式碼多執行幾次？(提示： $\sin(1)=0.8415$)

```
1  #include <stdio.h>
2  #include <math.h>
3  int x = 100;
4  int main(){
5      int x = 0;
6      double y = 0.0;
7      do{
8          y = 10*sin(x);
9          printf("x=%d, y=%lf\n", x, y);
10     } while(++x <= y);
11     printf("end of program\n");
12     return 0;
13 }
```

(A) 把行號3中的 $x=100$ 改為 $x=0$

(B) 把行號10中的 $++x$ 改為 $x++$

(C) 把行號6中 y 的初始值改為 -1.0

(D) 把行號3中 x 的初始值改為1

47. 在跑馬燈的設計上，可以由陣列中取出文字，並且顯示於螢幕上。如下片段程式執行完後 ShowMessage 字串為 "ILoveTaiwan"，則 Count 的初始值應為下列哪一個？

```
1  const int Count = ? ;
2  char Dictionary[50] = "IWhLoaorevYTeouTW5aM7iynwyuaTYn?";
3  char ShowMessage[12] = {0};
4  char *P = &Dictionary[0];
5  int Index = 0, Search = 0;
6  for(Index = 0; Index < 11 ; Index++)
7  {
8      ShowMessage[Index] = P[Search];
9      Search += Count;
10 }
```

- (A) 0 (B) 1 (C) 2 (D) 3

48. 有關指標的程式設計，如下片段程式可以在【1】的位置加入哪一行程式碼，編譯時不會產生任何錯誤或警告訊息？

```
....
1  int *p;
2  char MyName[] = {'A'};
3  int MyAge;
4  double MyWeight;
5  【1】
```

- (A) p = MyName; (B) p = &MyName[0];
(C) p = MyWeight; (D) p = &MyAge;

49. 小文設計了一個計算矩形面積的類別 CalculateArea，其宣告如下，則下列哪一個成員函式(Member Function)定義正確？

```
1  class CalculateArea{
2  private:
3      double Length, Width, Area;
4  public:
5      void SetPara(double, double);
6      double GetLength();
7      double GetWidth();
8      double GetArea();
9  };
```

- (A) void CalculateArea:: SetPara(double L, double W){Length = L; Width = W;}
(B) double GetLength(){return Length;}
(C) CalculateArea:: GetWidth(){return Width;}
(D) double CalculateArea:: GetArea(){ Area = Length * Width;}

50. 在沒有使用存取修飾字(private、public或protected)的情況下，下列敘述何者正確？

- (A) class 內的所有成員變數都可在 main() 主程式內存取
(B) struct 宣告的物件不可以做為 class 的成員
(C) 任何函式內的 struct 型態變數中的成員，在該函式的範圍內都可以被存取
(D) class 內的成員函式不可存取同一個 class 內的成員變數

題號	答案	題號	答案	題號	答案	題號	答案	題號	答案
1	A	11	B	21	D	31	A	41	A
2	C	12	C	22	D	32	B	42	A
3	AC	13	B	23	D	33	A	43	C
4	B	14	B	24	B	34	D	44	D
5	C	15	C	25	C	35	C	45	A
6	D	16	B	26	A	36	B	46	B
7	D	17	C	27	A	37	A	47	D
8	A	18	D	28	B	38	C	48	D
9	C	19	D	29	A	39	B	49	A
10	D	20	A	30	B	40	D	50	C

35. C++ 程式語言片段如下，小燕想要將 StepName[9][4] 內的文字，依燒燙傷急救步驟順序輸出到螢幕，Select[5] 內的?值為下列何者？

```

1  int TotalSteps = 5;
2  int Count = 0;
3  char StepName[9][4] = { "拖", "脫", "送", "沖", "蓋", "泡", "送" };
4  int Select[5] = { ? };
5  for (Count = 0; Count < TotalSteps; Count+=1)
6  {
7      std::cout << StepName[Select[Count]];
8  }

```

- (A) 4, 2, 5, 6, 3 (B) 4, 1, 6, 5, 7 (C) 3, 1, 5, 4, 6 (D) 3, 1, 4, 5, 2

36. 針對任意實係數二次多項式 $f(x)=ax^2+bx+c$ ，曉華想要計算並輸出 $f(x)$ 的函數值，因此寫了下列片段的C語言程式，發生程式編譯錯誤，主要原因以及可以採取更正措施為下列何者？

```

1  #include <stdio.h>
2
3  float f(float x){
4      return(a*x*x+b*x+c);
5  }
6  int main(){
7      float x, a=1, b=0, c=-1;
8      for(x=-10; x<=10; x=x+0.1)
9          printf("f(%.1f)=%.1f\n", x, f(x));
10 }

```

- (A) 變數 x, a, b, c 不可以宣告為 float，若宣告為 double 可以解決此問題
 (B) 變數 a, b, c 的初始值是整數，若改為包含小數位數的實數可以解決此問題
 (C) 變數 a, b, c 屬於 main() 中的區域變數 (Local Variable)，將變數 a, b, c 移到行號 2 宣告可以解決此問題
 (D) 變數 x, a, b, c 屬於全域變數 (Global Variable)，改宣告為在函式 f() 中的區域變數 (Local Variable) 可以解決此問題

37. 有一個非負整數的集合，每個整數都小於 10，曉華想利用如下的 C 語言程式來計算該集合的平均值且取整數為 average，依據 $average = (\sum_{i=0}^9 score[i]) / 10 = \sum_{i=0}^9 (score[i] / 10)$ 等式來寫出這個程式，並且使用一個陣列 score 來儲存這個整數的集合，可是發現不論 score 陣列裡面的數字增加或減少，平均值 average 都為 0，下列何者可以解決此問題？

```
1  #include <stdio.h>
2  int main(){
3      int score[10] = {1, 2, 3, 4, 5, 6, 7, 0, 0, 0};
4      int i;
5      double sum = 0;
6      int average;
7      for(i=0; i<10; i++){
8          sum = sum + score[i]/10;
9      }
10     average = sum;
11     printf("average=%d", average);
12     return 0;
13 }
```

- (A) sum 和 average 均改宣告為 double，且行號 11 的 %d 改為 %f
(B) 行號 5 的 sum 改宣告為 int 整數型態且行號 10 改為 average = (int) sum;
(C) 行號 8 改為 sum = sum + score[i]; 且行號 10 改為 average = (int) (sum/10);
(D) average 改宣告為 double，且行號 11 的 %d 改為 %f
38. 有關宣告陣列與指標的片段程式，下列敘述何者正確？
- ```
1 int Data[10] = {1,2,3,4,11,12,13,15};
2 int *pData = &Data[2];
```
- (A) if (pData[1] == 1) 判斷式結果為真  
(B) if (\*(pData + 3) == Data[5]) 判斷式結果為真  
(C) if (Data[9] == 0) 判斷式結果為假  
(D) if (Data[3] == (pData[5] - \*(pData+2))) 判斷式結果為假
39. 如下 C 語言程式，當程式執行完畢後，輸出為何？

```
1 #include <stdio.h>
2
3 int main(){
4 unsigned char i=3;
5 switch ((i&0x0e) % 5){
6 case(1):
7 printf("%c", '0'+i);
8 break;
9 case(2):
10 printf("%c", '0'+i*i);
11 case(3):
12 printf("%c", 'a'+i*i);
13 default:
14 printf("%c", 'z');
15 }
16 return(0);
17 }
```

- (A) 9jz                      (B) 927z                      (C) 9270                      (D) 9

40. C++程式語言片段如下，程式由行號1執行到行號9後，下列敘述何者正確？

```
1 enum States { Run = 1, Stop, Pause = 3, Exit };
2 States StateMachine = Pause;
3 int Command = 0;
4 std::cin >> Command;
5 if (Command == 1) StateMachine = Stop;
6 if (Command == 2 && StateMachine == Pause)
7 StateMachine = Run;
8 if (Command == 3 || StateMachine == Stop)
9 StateMachine = Exit;
```

- (A) 當Command為1，StateMachine 的值為 Exit
- (B) 當Command為2，StateMachine 的值為 Pause
- (C) 當Command為3，StateMachine 的值為 Stop
- (D) 當Command為4，StateMachine 的值為 Run

41. 有關運算子的優先順序，假設所有的變數都宣告為整數型態，下列哪一個C語言敘述運行的結果都是偶數？

- (A)  $\text{Result} = (\text{A} - 5 \gg 2) \mid 0x4;$
- (B)  $\text{Result} = ((\text{A} + 8) * \text{A} - 13) \& 0x1B;$
- (C)  $\text{Result} = (\text{A} - 15) / 2 + 6;$
- (D)  $\text{Result} = ((\text{A} + 124) \& 2) + 2 \% 5;$

42. 針對任意實係數一次多項式 $f(x)=ax+b$ ，曉華想要計算當 $x$ 落在 $[m, n]$ 範圍內時是否存在 $f(x)=0$ ，寫了如下的C語言程式，若變數found為1表示該範圍內存在 $f(x)=0$ ，則行號11內的if判斷式中，??可以為下列何者？

```
1 #include <stdio.h>
2 float a=1, b=0, m=-11, n=12;
3 float f(float x){
4 return(a*x+b);
5 }
6 int main(){
7 float x;
8 unsigned char found=0;
9 scanf("%f",&a); scanf("%f",&b);
10 scanf("%f",&m); scanf("%f",&n);
11 if(?? <=0)
12 found = 1;
13 printf("found=%d\n", found);
14 }
```

- (A)  $f(m) * f(n)$
- (B)  $f(m) + f(n)$
- (C)  $f(m) - f(n)$
- (D)  $f(m) \% f(n)$



### ▲閱讀下文，回答第 43-44 題

C 程式語言中的 typedef 關鍵字，可以將複雜的資料型態用簡單的別名來取代。例如下列資料型態宣告與程式片段，以 id 來儲存一個學生的學號，score 來儲存該學生的成績。假設班上共 N 個學生，第 i 個學生的資料儲存在 student[i-1] 裡面。

```
1 #include <stdio.h>
2 #define N 50
3 void main() {
4 typedef struct studentScore {
5 int id; //學號
6 float score; //成績
7 } SCORE;
8 SCORE student[N], *p;
9 float sscore;
10 p = student+28;
11 ...
```

43. 在行號 10 之後，若要取得學號 28 的學生成績放到變數 sscore，下列程式碼何者正確？  
(A) sscore = student[27].score;      (B) sscore = SCORE[27].student;  
(C) sscore = student->score[27];    (D) sscore = student->score[29];
44. 使用泡沫排序演算法來將 student 陣列中的成績 score 排序時，關於此演算法需經幾次的成績數值比較，可得排序結果？  
(A) 50                              (B) 1225                              (C) 24550                              (D) 245050

### ▲閱讀下文，回答第 45-46 題

下列程式片段為計算管道內水流量的全域類別，其中 ToatlFlow() 為計算流量值的成員函式。

```
1 enum Item{_FlowRate, _Time };
2 class Volume {
3 private:
4 double FlowRate, Time;
5 static double Offset;
6 public:
7 Volume () { FlowRate =0.0, Time = 0.0;}
8 Volume (double In1, double In2)
9 { FlowRate = In1; Time = In2; }
10 void SetOffset(double offs) { Offset = offs; }
11 double GetParameter(Item item)
12 { return (item == _Time) ? Time : FlowRate ;}
13 void SetPara(double FR,double T)
14 { FlowRate =FR; Time = T;}
15 protected:
16 double ToatlFlow() {return FlowRate*Time;}
17 } T1(1.0,2.3), T2;
18 double Volume :: Offset = 0.0;
```

45. 若要在 main 主程式內使用 T1 或 T2 物件來編寫程式，下列程式敘述何者正確？  
(A) double Value 1 = T1->GetParameter (\_Time);  
(B) T1->SetPara (100.5,50);  
(C) T2.FlowRate = 12.4;  
(D) T2.SetOffset (-3.2);

46. 根據Volume類別的宣告，程式碼都在main內執行，下列敘述何者正確？
- (A) 執行T2.SetOffset(0.5)；則T1內的Offset也會被同時修改為0.5
  - (B) T1與T2內的FlowRate預設值都為0.0
  - (C) 執行double X = T1.ToatlFlow()；可以將流量值回傳給X
  - (D) 執行float Y = T2.GetParameter((Item)1)；可以將FlowRate值回傳給Y

▲閱讀下文，回答第 47-48 題

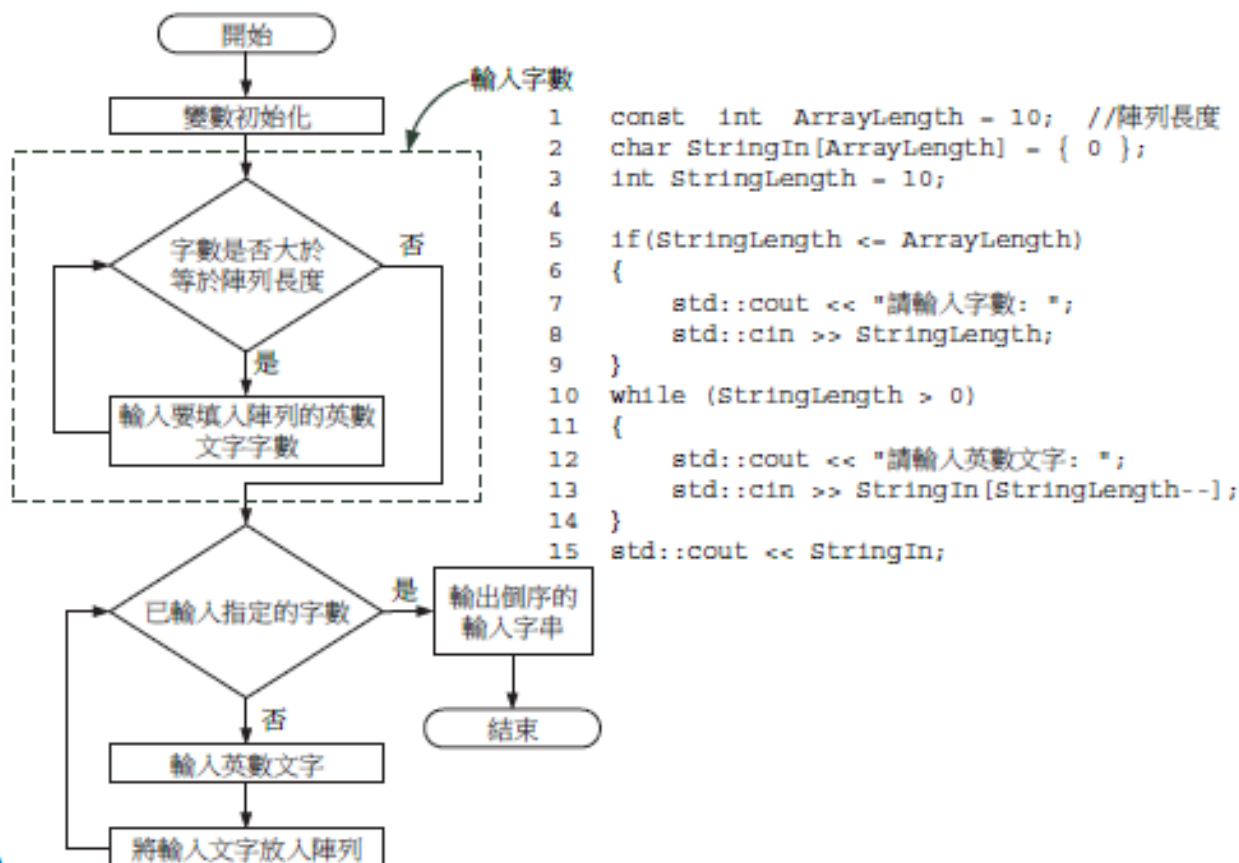
曉華寫了如下的 C 語言程式實現泡沫排序演算法，目的是將 N 個整數進行排序，其中字元 'a' 的 ASCII 碼為 97。

```
1 #include <stdio.h>
2 #define N 11
3 void swap(int a, int b){
4 int tmp;
5 tmp=a;
6 a=b;
7 b=tmp;
8 }
9 void main(void){
10 int numbers[N]={1,3,5,7,9,2,4,6,8,0,'a'};
11 int tmp, i, min;
12 //min=0;
13 for(min=0; min<N; min++){
14 for(i=0; i<N; i++){
15 if(numbers[i]<numbers[min]){
16 //swap(numbers+i, numbers+min);
17 tmp=numbers[min];
18 numbers[min]=numbers[i];
19 numbers[i]=tmp;
20 }
21 }
22 }
23 for(i=0; i<N; i++){
24 printf("%d ", numbers[i]);
25 }
```

47. 程式輸出結果為何？
- (A) a 9 8 7 6 5 4 3 2 1 0
  - (B) 0 1 2 3 4 5 6 7 8 9 a
  - (C) 97 9 8 7 6 5 4 3 2 1 0
  - (D) 0 1 2 3 4 5 6 7 8 9 97
48. 曉華想要把交換整數資料的程式碼寫成副程式，因此把行號17,18,19的程式改為註解，並且將行號16的註解拿掉以便啟用函式呼叫swap(.)，結果發現程式無法執行並出現錯誤訊息expected 'int' but argument is of type 'int \*'，錯誤原因為何？
- (A) 行號16呼叫swap()時，使用的引數資料型態與副程式不一致
  - (B) 行號16的numbers是陣列指標，不能和整數i, min相加
  - (C) 行號10的陣列宣告中，字元'a'和swap(.)函式中的整數變數a名稱上有衝突
  - (D) 行號12註解，導致min沒有初始值

### ▲閱讀下文，回答第 49-50 題

小明根據老師提供的流程圖(如圖十所示)，利用 C++ 語言程式完成在 main 函式中的程式碼片段。



圖(十)

49. 小明應如何修改行號 5，可以實現輸入字數部分的流程圖？

- (A) `if (StringLength < ArrayLength)`
- (B) `if (StringLength != ArrayLength)`
- (C) `while (StringLength >= ArrayLength)`
- (D) `while (StringLength != ArrayLength)`

50. 小明輸入 6 個英數文字，希望程式執行後可以印出倒序的字串，下列敘述何者正確？

- (A) 行號 10 改為 `while (StringLength != 0)`
- (B) 行號 10 改為 `while (StringLength < 0)`
- (C) 行號 13 改為 `std::cin >> StringIn[--StringLength];`
- (D) 行號 13 改為 `std::cin >> StringIn[StringLength++];`

| 題號 | 答案 | 題號 | 答案 | 題號 | 答案 | 題號 | 答案 | 題號 | 答案 |
|----|----|----|----|----|----|----|----|----|----|
| 1  | A  | 11 | C  | 21 | D  | 31 | D  | 41 | D  |
| 2  | D  | 12 | A  | 22 | D  | 32 | A  | 42 | A  |
| 3  | B  | 13 | A  | 23 | D  | 33 | A  | 43 | A  |
| 4  | C  | 14 | B  | 24 | D  | 34 | D  | 44 | B  |
| 5  | C  | 15 | D  | 25 | A  | 35 | C  | 45 | D  |
| 6  | B  | 16 | C  | 26 | B  | 36 | C  | 46 | A  |
| 7  | D  | 17 | B  | 27 | A  | 37 | C  | 47 | C  |
| 8  | B  | 18 | B  | 28 | A  | 38 | B  | 48 | A  |
| 9  | B  | 19 | D  | 29 | B  | 39 | A  | 49 | C  |
| 10 | D  | 20 | B  | 30 | C  | 40 | A  | 50 | C  |