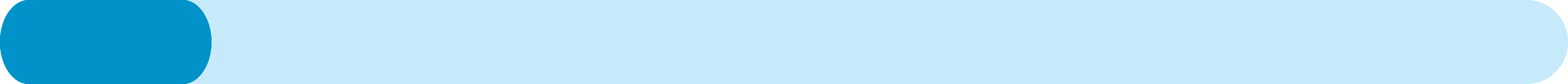
# 06



## 函式

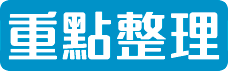
* 1. 公用函式
  2. 自訂函式
  3. 函式的資料傳遞

程式設計實習



6-1

公用函式



1. **函式(Function)**或**副程式(Subroutine)**：將特定功能或常用的程式碼獨立成一個區塊，使用函式有以下優點：
   1. 程式模組化、結構化。
   2. 增加程式的可讀性。
   3. 使程式易於維護。
2. 主函式 main( )也是一種函式，是程式執行的進入點。
3. cIc++語言提供標準函式庫，稱為公用函式或內建函式，一個函式庫就是多個同類型函式的集合。
4. **標頭檔(head file)**：副檔名為.h，用來描述函式庫內所有函式的特性和名稱，例如 math.h

是數學函式庫的標頭檔。

1. ANSI c 中常見的標準函式庫如下表：

|  |  |  |
| --- | --- | --- |
| 函式庫 | C 標頭檔 | C丑丑標頭檔 |
| 數學運算 | <math.h> | <cmath> |
| 輸入、輸出 | <stdio.h> | <iostream> |
| 日期、時間 | <time.h> | <ctime> |
| 數值轉換、亂數、動態記憶體 | <stdlib.h> | <cstdlib> |
| 字串處理 | <string.h> | <cstring> |

1. 在使用函式前，需先使用前置處理器指令引用函式庫，語法如下：

#include <函式庫標頭檔>

1. 常見的函式：

<math.h>的數學函式

|  |  |  |
| --- | --- | --- |
| 函式 | 說明 | 範例 |
| double exp(double x) | 指數 ex | exp(1) = 2.71828 |
| double log(double x) | 自然對數 logex(lnx) | log(2.71828) = 1 |
| double log10(double x) | 10 為底的對數 log10x | log10(100) = 2 |
| double pow(double x， double y) | xy | pow(10， 2) = 100 |
| double sqrt(double x) | x 的平方根 | sqrt(100) = 10 |
| double ceil(double x) | 大於或等於 x 的最小整數 | ceil(5.4) = 6  ceil(-5.4) = -5 |
| double floor(double x) | 小於或等於 x 的最大整數 | floor(5.4) = 5  floor(-5.4) = -6 |
| double hypot(double x， double y) | x2  y2 | hypot(6， 8) = 10 |

#### 6-2

Chapter 6 函式

|  |  |  |
| --- | --- | --- |
| 函式 | 說明 | 範例 |
| double round(double x) | 四捨五入 | round(4.4) = 4  round(4.5) = 5  round(-4.5) = -5 |
| double abs(double x) | 絕對值 | abs(-3) = 3 |
|  |  | 計算 sin 45°： |
|  |  | 角度 45°，則徑度 |
| double sin(double A) | 正弦函式 sinA，參數 A 需以徑度表示 |  |
|  |  | = 45\*3.14/180 = 0.785 |
|  |  | sin(0.785) = 0.707 |
| double cos(double A) | 餘弦函式 cosA，參數 A 需以徑度表示 | 計算 cos 45°： cos(0.785) = 0.707 |
| double tan(double A) | 正切函式 tanA，參數 A 需以徑度表示 | 計算 tan 45°： tan(0.785) = 1 |
| double asin(double A) | 反正弦函式 sin-1A，回傳徑度 | asin(0.707) = 0.785 |
| double acos(double A) | 反餘弦函式 cos-1A，回傳徑度 | acos(0.707) = 0.785 |
| double atan(double A) | 反正切函式 tan-1A，回傳徑度 | atan(1) = 0.785 |

<stdlib.h>的亂數函式

|  |  |  |
| --- | --- | --- |
| 函式 | 說明 | 範例 |
| void srand(unsigned int seed) | 初始化亂數函式，指定亂數種子 | **srand(time(NULL));**  time(NULL)為從 1970 年 1 月 1 日到目前為止的秒數，以此當做亂數種子 |
| int rand(void) | 隨機產 介於0 到32767 之間的整數 | **rand( )%100;** 產 0~99 之間的亂數值  **rand( )%47;** 產 0~46 之間的亂數值  **rand( )%47+1;** 產 1~47 之間的亂數值 |

<string.h>的字串處理函式

|  |  |  |
| --- | --- | --- |
| 函式 | 說明 | 範例 |
| size\_t strlen(char s[ ]) | 回傳字串長度，不包含結束  字元’\0’ | 三個字元陣列如下：  char s1[20] = "LeBron"; char s2[20] = "James"; char s3[20] = "";  **strlen(s1);** 取得 s1 的長度，回傳值為  6 (不包含結束字元’\0’) **strcpy(s3, s1);** s3 的值為 LeBron **strncpy(s3, s1, 2);** s3 的值為 Le  **strcat(s1, s2);** s1 的值為 LeBronJames **strncat(s1, s2, 2);** s1 的值為 LeBronJa **strcmp(s1, s2);** 比較字串 s1 和 s2，回傳值為 1(L 的 ASCII 碼為 7610，J 的 ASCII 碼為 7410，故 s1>s2) |
| char[ ] strcpy(char s1[ ], char s2[ ]) | 複製 s2 到 s1 |
| char[ ] strncpy(char s1[ ], char s2[ ],  size\_t n) | 複製 s2 的前n 個字元到 s1 |
| char[ ] strcat(char s1[ ], char s2[ ]) | 將 s2 連接到 s1 之後 |
| char[ ] strncat(char s1[ ], char s2[ ],  size\_t n) | 將 s2 的前 n 個字元，連接  到 s1 之後 |
| int strcmp(char s1[ ], char s2[ ]) | 由左至右逐一比較 s1 和 s2  的每個字元(ASCII 碼)， s1=s2，回傳值=0 s1>s2，回傳值>0 s1<s2，回傳值<0 |

#### 6-3

程式設計實習

以 strcpy ( )為例，另—種表示方式是 char ＊strcpy(char ＊s1， char ＊s2)，其中 char ＊s1 和 char[ ] s1 都是表示「指向字元的指標j，但使用 char[ ] s1 可以更清楚的表示是「指向字元陣列的第 1 個元素j。



註

### 01

******

設計一程式，由使用者輸入任意角度，計算其 sin、cos、tan 等三角函數後輸出。

程式碼：

|  |
| --- |
| 1. #include <stdio.h> |
| 2. #include <math.h> |
| 3. |
| 4. int main(){ |
| 5. const double PI=3.14; |
| 6. double degree, radian; |
| 7. double sinVal, cosVal, tanVal; |
| 8. |
| 9. printf("請輸入角度："); |
| 10. scanf("%lf", &degree); |
| 11. |
| 12. radian = degree\*PI/180; |
| 13. sinVal = sin(radian); |
| 14. cosVal = cos(radian); |
| 15. tanVal = tan(radian); |
| 16. |
| 17. printf("sin%d = %lf\n", degree, sinVal); |
| 18. printf("cos%d = %lf\n", degree, cosVal); |
| 19. printf("tan%d = %lf\n", degree, tanVal); |
| 20. } |

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 2 | 引用<math.h>函式庫標頭檔。 |
| 5 | 宣告常數 PI = 3.14。 |
| 9 ~ 10 | 使用者輸入角度。 |
| 12 | 計算徑度，徑度=角度＊3.14I180。 |
| 13 ~ 14 | 使用數學函式庫<math.h>提供的函式，計算三角函數。 |

Chapter 6 函式

### 02

******

設計一剪刀、石頭、布的猜拳程式，由電腦隨機出拳，和使用者猜拳後輸出勝負結果。

程式碼：

|  |
| --- |
| 1. #include <stdio.h> |
| 2. #include <stdlib.h> |
| 3. #include <time.h> |
| 4. |
| 5. int main(){ |
| 6. int pc, user; |
| 7. |
| 8. srand(time(NULL)); |
| 9. pc=rand()%3; |
| 10. |
| 11. printf("請輸入數字(0:剪刀 1:石頭 2:布)"); |
| 12. scanf("%d", &user); |
| 13. |
| 14. switch(pc){ |
| 15. case 0: |
| 16. printf("電腦出剪刀,"); |
| 17. if (user == 0) printf("你出剪刀，平手\n"); |
| 18. else if (user == 1) printf("你出石頭，你贏了！\n"); |
| 19. else if (user == 2) printf("你出布，你輸了！\n"); |
| 20. break; |
| 21. case 1: |
| 22. printf("電腦出石頭\n"); |
| 23. if (user == 0) printf("你出剪刀，你輸了\n"); |
| 24. else if (user == 1) printf("你出石頭，平手！\n"); |
| 25. else if (user == 2) printf("你出布，你贏了！\n"); |
| 26. break; |
| 27. case 2: |
| 28. printf("電腦出布\n"); |
| 29. if (user == 0) printf("你出剪刀，你贏了\n"); |
| 30. else if (user == 1) printf("你出石頭，你輸了！\n"); |
| 31. else if (user == 2) printf("你出布，平手！\n"); |
| 32. break; |
| 33. } |
| 34. } |

程式設計實習

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 1 ~ 3 | 引用函式庫標頭檔。 |
| 6 | 變數 pc 代表電腦出的拳，user 代表使用者出的拳。 |
| 8， 9 | 利用 srand( )和 rand( )函式，隨機產 0 到 2 之間的整數，0、1、2 分別代表剪刀、石頭、布。 |
| 11， 12 | 使用者輸入數字，0、1、2 分別代表剪刀、石頭、布。 |
| 14 ~ 33 | 判斷勝負並輸出結果。 |



( c ) 1. 關於 c 語言公用函式的描述，下列何者錯誤？

1. 函式庫的標頭檔，副檔名為.h
2. 使用#include 指令，引用函式庫標頭檔
3. 函式一定有回傳值 (D)函式不一定要有參數。

( A ) 2. 程式敘述 rand( )％3，不可能會回傳下列何值？ (A)3 (B)2 (c)1 (D)0。

( A ) 3. c 語言公用函式 sqrt(9)的回傳值為何？ (A)3 (B)2 (c)1 (D)0。

( D ) 4. c 語言公用函式 pow(-4， 2)的回傳值為何？ (A)2 (B)-2 (c)-16 (D)16。

( c ) 5. 佩佩要使用 c 語言的公用函式 sin()函式，計算 sin45空的結果，她必需先引用哪一個函式庫？ (A)string.h (B)stdlib.h (c)math.h (D)stdio.h。

( D ) 6. c 語言公用函式 floor(-3.7)的回傳值為何？ (A)4 (B)3 (c)-3 (D)-4。

( c ) 7. c 語言公用函式 ceil(-3.7)的回傳值為何？ (A)4 (B)3 (c)-3 (D)-4。

( D ) 8. 在 c 語言中執行 rand()％4，指令，下列哪一個值不可能出現？ (A)1 (B)2 (c)3 (D)4。

( B ) 9. 以下何者的值與其它 3 者不同？

* 1. abs(-5) (B)ceil(-5.3) (c)floor(5.3) (D)round(4.6)。

( A ) 10. 一個 *RC* 串聯電路，計算電容充電時的暫態電壓公式為*VC*  *E*(1 *e*

完成這個算式，下列哪一個函式最適合被使用？

1. exp( ) (B)log( ) (c)sqrt( ) (D)pow( )。

 *t*

*RC* )，要使用程式

Chapter 6 函式

( B ) 11. 下列程式片段執行後的輸出為何？

(A)9 (B)10 (C)11 (D)12。

int sum=0;

for (int i=-2;i<=2;i++){ sum = sum + pow(i,2);

}

printf("%d", sum);

( B ) 12. 在 C 語言中有兩個字串，str1[ ]="Orange"，str2[ ]="Apple"，則函式 strcmp(str1, str2)

的回傳值，最可能是下列何者？

(A)0 (B)1 (C)-1 (D)皆不可能。

( B ) 13. 承上題，執行 strcpy(str1, str2)後，str1 的內容為何？

(A)AppleOrange (B)Apple (C)Orange (D)OrangeApple。

( B ) 14. 接續上一題，若繼續執行 strcat(str1, str2)後，str1 的內容為何？

(A)AppleOrange (B)AppleApple (C)OrangeOrange (D)OrangeApple。

( C ) 15. 一函式 S( )如下，S(3, 4)的回傳值為何？ (A)24 (B)10 (C)25 (D)5。

double S(double x, double y){ return pow(hypot(x, y),2);

}

( D ) 16. 若函式 rand( )的回傳值為一介於 0 和 10000 之間的亂數，下列那個運算式可產 介於 100 和 1000 之間的任意數(包含 100 和 1000)？

(A) rand( ) % 900 + 100 (B) rand( ) % 1000 + 1

(C) rand( ) % 899 + 101 (D) rand( ) % 901 + 100。

( C ) 17. 有兩個字串 s1 和 s2，分別為” SuzanHola”和”AmigoFriend”，執行 strncat(s1,s2,5)後，下列何者正確？

(A)s1 = "AmigoFriendSuzan" (B) s2 = "AmigoFriendSuzan"

(C)s1 = "SuzanHolaAmigo" (D)s2 = "SuzanHolaAmigo"。

( A ) 18. 在 C 語言中宣告 2 個字元陣列分別為 a[]="baseball"，b[]="basketball"，請問執行 strcmp(a, b); 後，回傳值為何？

(A)-1 (B)0 (C)1 (D)2。

( D ) 19. 關於 C/C++語言的公用函式，下列說明何者正確？ (A)abs(double a)：回傳 a 的四捨五入值 (B)floor(double a)：回傳2:a 的最小整數 (C)round(double a)：回傳 a 的絕對值

(D)sqrt(double a)：回傳 。



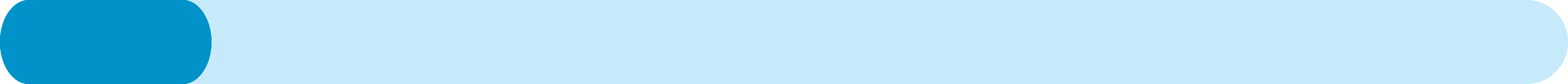
a

( C ) 20. 指令 sqrt(abs(-49));執行後的輸出為何？ (A)5 (B)6 (C)7 (D)9。

程式設計實習

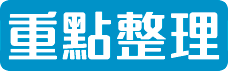
 打表示有詳解

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1.(C) | 2.(A) | 3.(A) | 4.(D) | 5.(C) | 6.(D) | 7.(C) | 8.(D) | 9.(B) | 10.(A) | 11.(B) |
| 12.(B) | 13.(B) | 14.(B) | 15.(C) | 16.(D) | 17.(C) | 18.(A) | 19.(D) | 20.(C) |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 1. (A)5 (B)-5 (C)5 (D)5。 2. 使用 exp()函式計算指數 e。   12. 比較第 1 個字元的 ASCII 碼，'O'的 ASCII 碼為 79，'A'的 ASCII 碼為 65，str1 大於 str2，回傳值>0。  17. 將 s2 的前 5 個字元，串接到 s1。  18. e 的 ASCII 碼為 10110，k 的 ASCII 碼為 10710，故 s1<s2，回傳-1。 | | | | | | | | | | | |



6-2

自訂函式



1. 函式是獨立的程式碼區塊，可以輸入外部資料，也可以將資料輸出。
2. 在程式中呼叫函式的時機：
   1. 在主函式 main( )中呼叫函式。
   2. 在函式 A( )中呼叫函式 B( )。
   3. 在函式中呼叫自已，又稱為遞迴函式。
3. 引數與參數：
   1. 呼叫函式時，傳遞給函式的值，稱為**引數(Argument)**。
   2. 函式被呼叫時，負責接收資料的變數，稱為**參數(Parameter)**。
4. 當函式有回傳值時，在函式的程式碼區塊內，使用關鍵字 return 將資料回傳，回傳後函式隨即結束，而寫在 return 之後的程式碼，將不會被執行。
5. 若函式沒有回傳值，則在函式名稱之前使用 void 關鍵字表示沒有回傳值。
6. 自訂函式包含宣告和定義(實作)2 個部份，語法如下：
   1. **宣告**函式原型：規範回傳值與參數的資料型態，需在主函式 main( )與函式定義之前宣告。

回傳恒資料型態 函式名稱(參數資料型態, …);

/\* 宣告函式原型 C()，1 個浮點數型態的參數，回傳整數 \*/

3. int C(float);

2. void B(int, int); /\* 宣告函式原型 B()，2 個整數型態的參數，沒有回傳恒 \*/

/\* 宣告函式原型 A()，沒有參數，也沒有回傳恒 \*/

1. void A();

Chapter 6 函式

* 1. **定義**函式：負責參數宣告與程式區塊實作，若定義寫在主函式 main( )之前，則可以省略上述的函式原型宣告。

回傳恒資料型態 函式名稱(參數資料型態 參數名稱, …){ 程式碼區塊 }

4. void A(){ … }

/\* 宣告 1 個浮點數參數x，同時實作程式碼區塊 \*/

6. int C(float x){ … }

/\* 宣告 2 個整數參數a 和b，同時實作程式碼區塊 \*/

5. void B(int a, int b){ … }

1. 呼叫函式時，會暫停目前執行的程式碼，待函式執行結束後，再返回原來的程式碼位置繼續執行，呼叫函式的語法如下(引數的數量與型態，需符合函式原型的規範)，

函式名稱(引數, …);

9. int i=C(3.14); /\* 呼叫函式C()，引數是浮點數 3.14，回傳恒指定給整數變數i \*/

/\* 呼叫函式B()，引數是整數 10 和 20 \*/

8. B(10, 20);

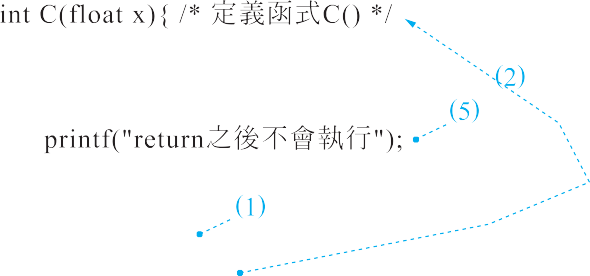
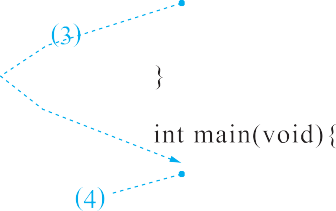
/\* 呼叫函式A()，沒有引數 \*/

7. A();

1. 以下範例說明函式呼叫的過程，







1. 從主函式 main()開始執行。
2. 呼叫函式 C(3.14)，傳入的引數為 3.14，函式 C()被呼叫後，會宣告一參數 x 並指定為

3.14(float x = 3.14)。

1. 執行函式 C()，直到遇到 return 指令，回傳 x\*100，函式執行結束，返回原程式碼繼續執行。
2. 將函式回傳值指定給變數 i(int i=314)。
3. 寫在 return 之後的程式碼，不會執行。

#### 6-9

程式設計實習

1. 區域變數與全域變數

|  |  |  |
| --- | --- | --- |
|  | 區域變數 | 全域變數 |
| 種類 | * if、for、while 結構內宣告的變數 * 函式的參數 * 函式內部宣告的變數 | 函式外部宣告的變數 |
| 存取權 | 只能在所在的程式區塊(即左右大括號{ }包  圍的部份)內被存取 | 在程式的任何位置都可以存取 |
| 生命週期 | 程式區塊結束即消失 | 程式結束才消失 |

1. 程式執行時，每個變數存在的時間不同，稱為變數的 **命週期**。下列程式碼宣告 1 個全域變數 a 和 3 個區域變數 b、c 和 i，變數建立(作業系統配置記憶體)和消失(作業系統回收記憶體)的時間順序如下，其中 for 迴圈內的區域變數 c，每執行一次迴圈即消失(5)，進入迴圈時再重新建立(4)，共重覆 10 回合。
2. 建立全域變數 a
3. 建立 main()的區域變數 b
4. 建立 A()的區域變數 i
5. 建立 for 迴圈的區域變數 c
6. for 迴圈的區域變數 c 消失
7. A()的區域變數 i 消失
8. main()的區域變數 b 消失
9. 全域變數 a 消失

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. int a=1; | /\* 全域變數 a \*/ (1) | | | | | |
| 2. | | | | | |  |
| 3. void A(){ | | | | | |  |
| 4. int i=0; | /\* 函式 A()的區域變數 i \*/ (3) | | | |  |  |
| 5. for(i=0; i<10; i++){ | | |  | | |  |
| 6. int c=0; | /\* for 迴圏的區域變數 c \*/ (4) | |  | | |  |
| 7. c=c+i;  × | |  |  | | |  |
| 8. } | (5)  × | |  | | |  |
| 9. } | (6) | | | |  |  |
| 10. | | | | | |  |
| 11. int main(void) | | | | | |  |
| 12. { | | | | | |  |
| 13. int b=0; | /\* 主函式 main()的區域變數 b \*/ (2) | | | |  |  |
| 14. A(); | × | | | × | |  |
| 15. } | (7) | | | | (8) | |

#### 6-10

Chapter 6 函式

1. 不同程式區塊的區域變數，因存在於記憶體的不同位置，因此允許使用相同的變數名稱，存取時，採由內而外的方式，優先存取同一程式區塊內的區域變數，若區域變數不存在，則往上一層尋找。下例的第 1、4、6、18 行皆宣告一名稱為 s 的變數，虛線指向存取的對象，第 13 行因函式 B()內無區域變數 s，故存取對象為上一層的全域變數 s。



1.

2.

3.

4.

5.

6.

7.

8.

9.

int s = 2;

/\* 全域變數 s \*/

void A(){

int s = 1;

for (int i=0;i<10;i++){ int s = 0;

s=s+1;

}

s=s+1;

/\* 函式 A()的區域變數 s \*/

/\* for 迴圏結構的區域變數 s \*/

10. }

1. void B(int a, int b)
2. {
3. s=s+1;

14. }

15.

1. int main(void)
2. {

|  |  |  |
| --- | --- | --- |
| 18. | int s = 3; | /\* 主函式 main()的區域變數 s \*/ |
| 19. | A(); |  |
| 20. | B(); |  |
| 21. | s=s+1; |  |
| 22. } |  |  |

1. **重載(Overload)**函式：C++允許在函式的參數個數或參數資料型態不同時，使用相同的函式

名稱，下例為 3 個名稱相同的重載函式，根據參數的不同，自動呼叫不同的函式。

void myFun(int, int, int) { … } void myFun(int) { … }

void myFun(double) { … }

C 語言不允許函式重載，但允許運算子重載，如運算子「-j可以是負號或減號兩種功能。



註

#### 6-11

程式設計實習

1. 函式自已呼叫自已，稱為遞迴函式(Recursive Function)，遞迴函式必須具備**至少一個終止條件**，否則函式將無法結束，下面是一個計算 n! (n 階層)的函式 K( )，執行 K(3)會回傳 3x2x1x1，函式執行的過程如下。



|  |
| --- |
| 1. int K(int n){ |
| 2. if (n == 0) /\* 終止條件：當n=0 時，不再呼叫自己 \*/ |
| 3. return 1; |
| 4. else |
| 5. return n \* K(n-1); /\* 呼叫自己 \*/ |
| 6. } |

1. 遞迴函式的優點是程式較簡潔明確且可讀性較高，缺點是執行效率低，會佔用較多記憶體



間。

***03 *** 

世界衛 組 建議以身體質量指數(BMI)來衡量肥胖程度，請建立一自訂函式 bmi( )，該函式會回傳 BMI 值。BMI 的計算公式為 BMI  體重(kg)

身高(m)2

程式碼：

|  |
| --- |
| 1. #include <stdio.h> |
| 2. #include <math.h> |
| 3. |
| 4. double bmi(double, double); /\* 宣告函式原型 \*/ |
| 5. |
| 6. double bmi(double height, double weight){ /\* 函式定義 \*/ |
| 7. double bmiVal; |
| 8. bmiVal=weight/pow(height/100, 2); |
| 9. return bmiVal; |
| 10. } |
| 11. |
| 12. int main(){ |
| 13. double height, weight; |
| 14. |
| 15. printf("請輸入身高(cm):"); |
| 16. scanf("%lf", &height); |
| 17. printf("請輸入體重(kg):"); |
| 18. scanf("%lf", &weight); |
| 19. |
| 20. printf("%lf", bmi(height, weight)); /\* 呼叫函式 bmi()，直接輸出回傳恒 \*/ |
| 21. } |

#### 6-12

Chapter 6 函式

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 2 | 引用數學函式庫<math.h>。 |
| 4 | 宣告 bmi( )函式原型：規範 2 個 double 型態的參數，回傳值也必需是 double 型態。 |
| 6 ~ 10 | 函式定義：實作 BMI 的計算公式，參數 height 和 weight 負責接收傳入的引數。 |
| 8 | 計算 BMI 值，利用 pow()函式進行次方運算。 |
| 9 | 回傳 BMI 值。 |
| 15 ~ 18 | 使用者輸入身高和體重。 |
| 20 | 呼叫 bmi( )函式，提供引數 height 和 weight。 |
| 6， 13 | bmi( )和 main( )都有 height 和 weight 變數，因為不同函式的區域變數彼此獨立，所以變數名稱相同並不會造成編譯錯誤。 |

***04 *** 

費式數列，由義大利人費先 提出，該數列由 0 和 1 開始，之後的每個數為前兩個數字相加，舉列如下：0， 1， 1， 2， 3， 5， 8， 13， 21， 34.，現在，費先 請你幫忙設計一遞迴函式 fib( )，該函式可以直接回傳一費式數列的值(例如 fib(5)會回傳 5)。

程式碼：

|  |
| --- |
| 1. int fib(int); |
| 2. |
| 3. int fib(int x){ |
| 4. if (x == 0){ /\* 終止條件— \*/ |
| 5. return 0; |
| 6. } |
| 7. else if (x == 1){ /\* 終止條件二 \*/ |
| 8. return 1; |
| 9. } |
| 10. else{ |
| 11. return fib(x-1)+fib(x-2); /\* 呼叫自己 \*/ |
| 12. } |
| 13. } |

#### 6-13

程式設計實習

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 1 | 函式原型宣告，規範 1 個 int 型態的參數，且需回傳一 int 資料。 |
| 3 ~ 13 | fib( )函式定義，有兩個終止條件，當 x==0 或 x==1 時，函式結束，並回傳 0 或 1。 |
| 11 | 不符合終止條件時，fib( )會呼叫自己，以 fib(5)為例，此遞迴函式的執行過程如下， |

***05 *** 

就讀國小六年級的表妹正在學習利用輾轉相除法來取得 2 個整數的最大公因數，請你幫表妹設計一遞迴函式 GcD( )，該函式會回傳 2 個整數的最大公因數(例如 GcD(9， 12)會回傳 3)。

程式碼：

|  |
| --- |
| 1. int GCD(int, int); |
| 2. |
| 3. int GCD(int x, int y){ |
| 4. int r; |
| 5. r = x % y; |
| 6. if (r == 0) /\* 終止條件 \*/ |
| 7. return y; |
| 8. return GCD(y, r); /\* 呼叫自己 \*/ |
| 9. } |

#### 6-14

Chapter 6 函式

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 1 | 函式原型宣告，規範 2 個整數型態的參數，且需回傳一整數資料。 |
| 3 ~ 9 | GcD( )函式定義，終止條件為 r==0，此時函式結束並回傳最大公因數 y。 |
| 11 | 不符合終止條件時，GcD( )會呼叫自己，以 GcD(9， 12)為例，此遞迴函式的執行過程如下， |



( D ) 1. 關於函式的描述，下列何者正確？

(A)函式一定要有回傳值 (B)遞迴函式只能有一個終止條件 (c)自訂函式一定要宣告函式原型 (D)函式可以傳入多個參數。

( B ) 2. 要從函式將值回傳，要使用哪一個關鍵字？ (A)back (B)return (c)tx (D)send。 ( c ) 3. 要建立自訂函式時，若該函式不需要回傳值，則在宣告函式原型時，回傳值的資料

型態為何？ (A)null (B)nothing (c)void (D)no。

( D ) 4. 下列程式碼中，哪一個變數是全域變數？ (A)m (B)n (c)k (D)sum。

int sum=0; void F(int n){

int m=n; sum=sum\*m;

}

int main(){

int k=10; F(k);

}

( D ) 5. 關於 c++語言的函式，下列哪一個說法是錯誤的？ (A)函式自己呼叫自己，稱為遞迴函式

(B)在函式內部宣告的變數，稱為區域變數 (c)全域變數與區域變數的名稱可以相同

(D)在 main()中呼叫函式時所傳的值，稱之為參數。

#### 6-15

程式設計實習

( B ) 6. 下列有關 c 語言的敘述，哪一個是錯誤的？

(A)同一陣列的所有元素，一定是相同的資料型別 (B)函式一定要有回傳值 (c)使用變數前未宣告，會產 編譯錯誤 (D)使用編譯程式轉換成機器語言。

( c ) 7. 小豪宣告一個宣式原型 int PI(double， double)，該函式不可能回傳哪一個數值？ (A)0 (B)-1 (c)3.14 (D)1。



( B ) 8. 下列 c 程式執行後的輸出為何？ (A)0 (B)1 (c)2 (D)3。

int A(int a){ a--;

return a;

}

void B(int b){ b++;

printf("%i", A(b));

}

int main(void)

{

int x=1; B(x);

}

( A ) 9. 執行下列程式片段後的輸出結果為何？ (A)5 0 (B)5 5 (c)0 0 (D)5 5。

int x=0;

void Fun(int a){ int x=0; x=x+a;

printf("%i", x);

}

int main(void)

{

Fun(5);

printf("%i", x);

}

#### 6-16

Chapter 6 函式

( B ) 10. 執行下列程式片段後的輸出結果為何？ (A)0 (B)1 (c)10 (D)11。

int s = 1; void Y(){

int s = 1;

for (int i=0;i<10;i++){ s=s+1;

}

}

int main(void)

{

Y();

printf("%i", s);

}

( B ) 11. 下列程式的輸出結果為何？ (A)1024 (B)512 (c)256 (D)2048。

#include <stdio.h> int a=128;

int X(int b)

{

int i;

for( i=0;i<4;i++) b\*=2;

return (b);

}

int main()

{

int a=32; a=X(a);

printf("%d\n", a);

}

( c ) 12. 一程式片段如下，F(7)的結果為何？ (A)10 (B)11 (c)12 (D)13。

int F(int x){ if (x<0)

return 1; else

return F(x-2) + F(x-3);

}

#### 6-17

程式設計實習

( B ) 13. 下列程式執行後的輸出為何？ (A)55 (B)50 (c)0 (D)10。

int s=50;

void K(int a){ int s=0;

for(int i=1;i<=a;i++){ s=s+i;

}

}

int main(){

int a=10; K(a);

printf("%d", s);

}

( B ) 14. 下列函式兩個回傳式分別該如何撰寫，才能正確計算並回傳兩參數 a， b 之最大公因數(Greatest common Divisor)？

(A)a， GcD(b，r) (B)b， GcD(b，r) (c)a， GcD(a，r) (D)b， GcD(a，r)。

int GCD (int a, int b) { int r;

r = a % b; if (r == 0)

return ; return ;

}

( B ) 15. 下列給定下列程式，其中 s 有被宣告為全域變數，請問程式執行後輸出為何？

(A) 1，6，7，7，8，8，9 (B) 1，6，7，7，8，1，9 (c) 1，6，7，8，9，9，9 (D) 1，6，7，7，8，9，9。

int s = 1; // 全域變數

void add (int a) { int s = 6;

for( ; a>=0; a=a-1) {

printf("%d,", s); s++;

printf("%d,", s);

}

}

int main() {

printf("%d,", s); add(s); printf("%d,", s); s = 9;

printf("%d,", s); return 0;

}

#### 6-18

Chapter 6 函式

( D ) 16. 下列程式關於 c 語言函式的敘述，何者正確？

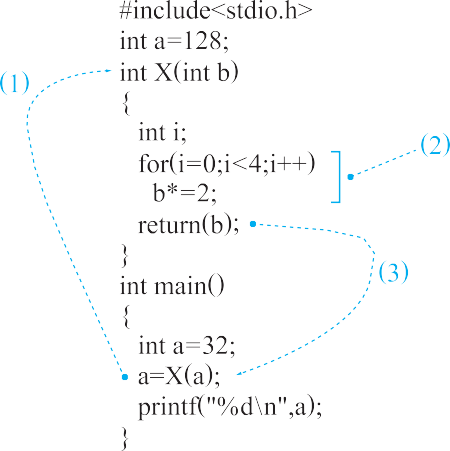
(A)hypot()為自訂函式

(B)呼叫自訂函式時，需在函式前面加上 call 關鍵字

(c)函式若有傳回值，則需在定義時，在函式的前面加上 void 關鍵字

(D)在自訂函式中，可以呼叫自己本身這個函式。

打 表示有詳解



1.(D) 2.(B) 3.(c) 4.(D) 5.(D) 6.(B) 7.(c) 8.(B) 9.(A) 10.(B)

11.(B) 12.(c) 13.(B) 14.(B) 15.(B) 16.(D)

1. (A)函式不一定要有回傳值 (B)可以有多個終止條件 (c)不一定要宣告函式原型。

1. (D)呼叫函式時所傳入的值，稱為引數。

7. 回傳值需為整數。

1. 區域變數與全域變數存在於記憶體的不同位置，可視為獨立的變數，允許使用相同名稱。此程式在函式

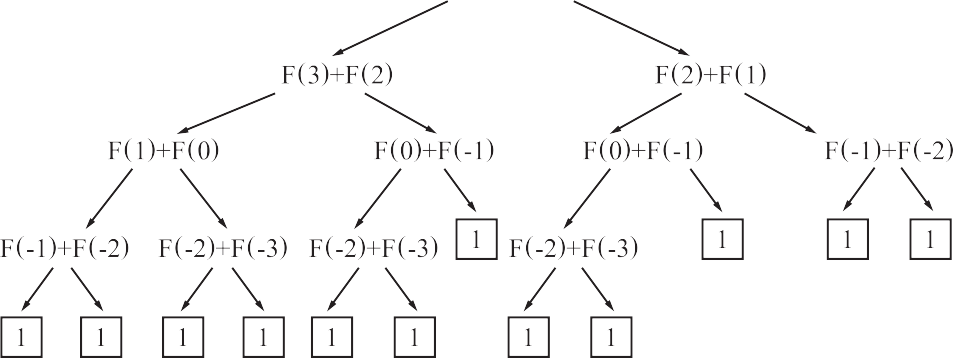
Fun( )內對其區域變數 x 進行運算，並不會影響全域變數 x。

1. 參考第 9 題。
2. 程式執行步驟如下，
   1. 呼叫 X(a)，引數為區域變數 a 的值，函式 X()被呼叫後，參數 b 的值等於 32，
   2. 運算後，b 等於 512，
   3. 函式 X()結束，返回原程式位置，回傳 512，並將回傳值指定給區域變數 a。

#### 6-19

程式設計實習

1. 呼叫 F(7)的執行過程如下，



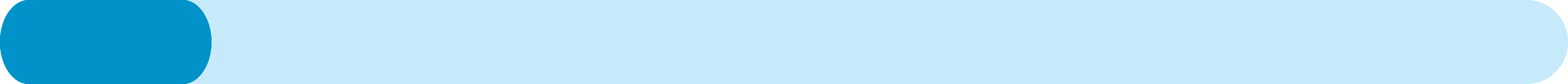
1. 參考第 9 題解析。
2. 參考單元 6-2 範例 5。
3. 程式執行步驟如下，
   1. 宣告全域變數 s，初始值為 1，
   2. 輸出全域變數 s，
   3. 呼叫 add(s)，傳入的引數為全域變數 s 的值，函式被呼叫後，參數 a 等於 1，
   4. 宣告區域變數 s，此變數只在 add()函式內有效，函式結束即消失，因為區域變數與全域變數的存取範圍不同，故允許同名，
   5. 輸出區域變數 s 的值：6 7 7 8，
   6. 函式 add()結束，返回原程式位置，
   7. 輸出全域變數 s 的值：1 9。
4. (A)hypot()為公用函式

(B)呼叫自訂函式時，不需在函式之前加上任何關鍵字

(c)函式若有傳回值，則需在定義時，在函式的前面加上傳回值的資料型態 (D)函式可以呼叫自己本身，稱為遞迴函式

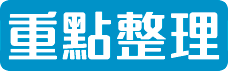
#### 6-20

Chapter 6 函式



6-3

函式的資料傳遞



1. 函式使用**參數(parameter)**來接收外部傳入的資料，資料可以是變數或陣列。
2. 傳遞變數的方式：

##### 傳值呼叫(call by value)

資料以**值**的型式傳遞，函式的參數是**一般變數**，其值等於**來源變數的值**，參數和來源變數分別存在於記憶體的不同位置，所以**參數的改變不會影響來源變數的值**。

以下列程式碼為例，在 byValue( )內部將參數 x 指定為 100，並不會改變 main( )中變數 a 的值(a 的值仍然是 10)。

|  |  |  |
| --- | --- | --- |
| 1. void byValue(int x){ | /\* | 參數 x 是—般變數 \*/ |
| 2. printf("%d\n", x); | /\* | 輸出 10 \*/ |
| 3. x = 100; | /\* | 指定參數 x 的恒為 100 \*/ |
| 4. } | | |
| 5. int main(){ | | |
| 6. int a = 10; | | |
| 7. byValue(a); | /\* | 呼叫byValue()，傳遞的引數是變數a 的恒 \*/ |
| 8. printf("%d\n", a); | /\* | 變數 a 的恒不變，依舊是 10 \*/ |
| 9. } | | |

程式執行時，變數在記憶體內的變化如下：第 6 行，宣告變數 a

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

第 7 行，呼叫函式 byValue(a)，採傳值呼叫，等同於執行 int x=a;

第 3 行，將參數 x 指定為 100



第 4 行，函式結束，參數 x 消失，返回第 8 行

##### **傳**址呼叫(call by address)



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

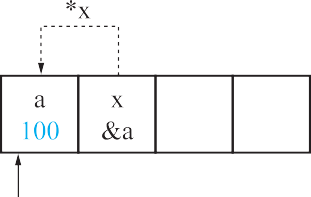
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

資料以**位址**的型式傳遞，函式的參數是**指標變數**，其值是**來源變數的位址**，當**參數指向的記憶體內容改變時，來源變數的值也會跟著改變**。

以下列程式碼為例，第 7 行呼叫byAddr( )時，傳入的引數是變數 a 的位址，此時 byAddr( )的參數 x 會指向來源變數 a，所以當 x 指向的記憶體內容被指定為 100 時，main( )中變數 a 的值也會改變(a 的值變成 100)。

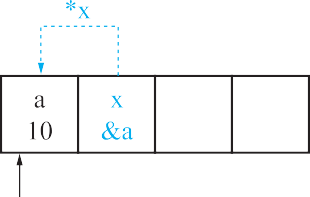
#### 6-21

程式設計實習



|  |
| --- |
| 1. void byAddr(int \*x){ /\* 參數 x 是指標變數 \*/ |
| 2. printf("%d\n", \*x); /\* 輸出 10 \*/ |
| 3. \*x = 100; /\* 指定參數 x 指向的記憶體位址的內容為 100 \*/ |
| 4. } |
| 5. int main(){ |
| 6. int a = 10; |
| 7. byAddr(&a); /\* 呼叫byAddr()，傳遞的引數是變數a 的位址 \*/ |
| 8. printf("%d\n", a); /\* 變數 a 的恒已經變成 100 \*/ |
| 9. } |

程式執行時，變數在記憶體內的變化如下：第 6 行，宣告變數 a



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

第 7 行，呼叫函式 byValue(&a)，採傳址呼叫，等同於執行 int \*x=&a;



第 3 行，將 x 指向的記憶體內容，指定為 100



第 4 行，函式結束，參數 x 消失，返回第 8 行





取址運算子「&j，取得某個變數的記憶體位址。

取恒運算子「\*j，存取指標變數指向的記憶體內容。



註

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

#### 6-22

Chapter 6 函式

1. 傳遞陣列資料時，**需同時提供陣列和元素個數**，函式有 2 種接收陣列資料的方式：
   1. 以陣列的方式接收，在函式內部使用陣列索引存取陣列元素

|  |
| --- |
| 1. /\* 宣告函式原型：規範參數的型態為整數陣列(用int[]表示)和整數變數 \*/ |
| 2. void G(int[], int); |
| 3. |
| 4. /\* 定義函式：宣告 2 個參數分�為整數陣列 x[]和整數變數 xLen \*/ |
| 5. void G(int x[], int xLen){ |
| 6. for (int i=0; i<xLen; i++){ |
| 7. x[i]=x[i]+1; /\* 以陣列索引的方式存取陣列元素 \*/ |
| 8. } |
| 9. } |

* 1. 以指標的方式接收，在函式內部使用指標存取陣列元素

|  |
| --- |
| 1. /\* 宣告函式原型：規範參數的型態為指標變數和整數變數 \*/ |
| 2. void F(int\*, int); |
| 3. |
| 4. /\* 函式定義：宣告 2 個參數分�為指標變數x 和整數變數 xLen \*/ |
| 5. void F(int \*x, int xLen){ |
| 6. for (int i=0; i<xLen; i++){ |
| 7. \*(x+i) = \*(x+i)+1; /\* 以指標的方式存取陣列元素 \*/ |
| 8. } |
| 9. } |

1. 呼叫上述函式 G( )或函式 F( )，均需要提供 2 個引數，分別為**陣列名稱(第 1 個元素的指標)**和**元素個數**，只要**在函式內改變陣列的內容，來源陣列的內容也會跟者改變。**

|  |
| --- |
| 1. int main(){ |
| 2. int a[5] = {1,2,3,4,5}; |
| 3. G(a, 5); /\* 呼叫函式 G()，函式結束後，陣列內容變成 2 3 4 5 6 \*/ |
| 4. F(a, 5); /\* 呼叫函式 F()，函式結束後，陣列內容變成 3 4 5 6 7 \*/ |
| 5. } |

#### 6-23

程式設計實習

***06 *** 

某量販店推出消費打 9 折，滿干再送百的快閃活動，即消費 額打 9 折，折後每滿 1000 元，再贈送 1 張 100 元的折價卷，請以傳址呼叫的方式設計一函式 shopback( )，計算打折後的消費

額，並顯示獲得的折價卷張數。

程式碼：

|  |
| --- |
| 1. #include <stdio.h> |
| 2. |
| 3. void shopback(int\*); /\* 函式原型宣告，規範參數是指標變數 \*/ |
| 4. |
| 5. void shopback(int \*x){ /\* 宣告參數 x 是指標變數 \*/ |
| 6. int coupon; |
| 7. |
| 8. \*x = (\*x)\*0.9; |
| 9. coupon = \*x / 1000; |
| 10. |
| 11. printf("獲得 %d 張折價卷\n", coupon); |
| 12. } |
| 13. |
| 14. int main(){ |
| 15. int m; |
| 16. |
| 17. printf("請輸入消費金額："); |
| 18. scanf("%d", &m); |
| 19. |
| 20. shopback(&m); /\* 呼叫shopback()，引數是變數m 的位址 \*/ |
| 21. printf("打折後的消費金額 %d", m); |
| 22. } |

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 3 | shopback( )函式原型宣告，規範參數的型態為指標變數。 |
| 5 ~ 12 | shopback( )函式定義，當第 20 行呼叫 shopback( )時，參數 x 會指向 main( )中的變數 m。 |
| 8 | 使用取值運算子「＊j，存取 x 指向的記憶體內容，當＊x 改變時，main( )中的變數 m 也會跟著改變。 |
| 20 | 呼叫 shopback( )函式，引數是變數 m 的位址（使用使用取址運算子「＆j）。 |

#### 6-24

Chapter 6 函式

### 07

******

設計一自訂函式 arrMax( )，傳入整數陣列，該函式會回傳陣列中最大值的索引，並使用此函式改寫 P5-5 範例 03。

程式碼：

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

#include <stdio.h>

int arrMax(int[], int);

int arrMax(int arr[], int len){ int max=1;

for (int i=2;i<=len;i++){

if (arr[i-1]>arr[max-1]) max = i;

}

return max;

12. }

13.

1. int main(){
2. int money[] = {900,850,925,1012,936,998,1019,899,986,978,1002,1011};
3. int max = arrMax(money, 12);
4. printf("營業額最高的月份：%d 月\n", max);
5. printf("營業額：%d 仟萬元", money[max-1]);

19. }

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 3 | arrMax( )函式原型宣告，規範參數的型態為整數陣列和整數變數。 |
| 5 ~ 12 | arrMax( )函式定義，宣告 2 個參數，分別為整數陣列 arr[ ]和整數變數 len，使用陣列索引的方式存取陣列，找出最高營業額的月份 max，將 max 回傳。 |
| 16 | 呼叫 arrMax( )函式，提供 2 個引數，分別為陣列名稱和元素個數。 |

#### 6-25

程式設計實習

### 08

******

寫出以下程式執行後之輸出結果？

|  |
| --- |
| 1. void F(char a[]) { |
| 2. int n, i; |
| 3. char t; |
| 4. if (a[0] != '\0') { |
| 5. for (n=0; a[n]!='\0'; n++); |
| 6. t = a[n-1]; |
| 7. for (i=n-1; i>0; i--) { |
| 8. a[i] = a[i-1]; |
| 9. } |
| 10. a[0] = t; |
| 11. F(a+1); |
| 12. } |
| 13. } |
| 14. int main() { |
| 15. char a[] = "HELLOWORLD"; |
| 16. F(a); |
| 17. printf("%s", a); |
| 18. return 0; |
| 19. } |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| 行數 | | | 說明 |
| 4 | | | 終止條件，當傳入陣列 a 的第 1 個元素 a[0]為結束字元’\0’時，條件式不成立，函式 F( )結束。 |
| 5 | | | 使變數 n 為傳入陣列 a 的大小（不包含結束字元’\0’）。 |
| 6 | | | 使變數 t 為傳入陣列 a 的最後一個字元。 |
| 7 ~ 9 | | | 將傳入陣列 a 的每一個元素都往後移一格（不包含結束字元’\0’）。 |
| 10 | | | 指定陣列 a 的第 1 個元素為 t（即將最後一個字元移到最前面）。 |
| 11 | | | F( )呼叫自己，引數為傳入陣列 a 的第 2 個元素的位址。 |

#### 6-26

Chapter 6 函式

行數

說明

16

遞迴函式 F( )的執行過程如下，最後字元陣列 a[ ]的內容為 DLROWOLLEH。

|  |  |  |
| --- | --- | --- |
| 呼叫 F（ ）的次數 | 傳入的陣列內容 | F（ ）執行後的陣列內容 |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 | 後續步驟請讀者自行練習 | |



( c ) 1. 今有一函式原型void K(int＊)和整數變數data，下列何者是呼叫函式 K( )的正確語法？ (A)K(data) (B)int x=K(data) (c)K(＆data) (D)int x=K(＆data)。

( D ) 2. 函式呼叫有兩種方式：傳值呼叫(call by value)和傳址呼叫(call by address)，下列何者不正確？

(A)傳址呼叫，不能將函式中宣告的變數位址回傳 (B)傳值呼叫，參數在函式的變化不會影響本來的變數

(c)傳址呼叫，參數在函式內的變化會直接改變本來的變數 (D)傳值呼叫，參數必需是指標變數。

( A ) 3. 執行函式呼叫時，若傳入的來源變數不希望被改變，可採用何種呼叫方式？

(A)傳值呼叫 (B)傳址呼叫 (c)傳名呼叫 (D)遞迴呼叫。

( A ) 4. 下列程式碼執行後的輸出為何？ (A)10，20 (B)20，10 (c)10，10 (D)20，20。

void swap(int a, int b){ int tmp;

tmp=a; a=b; b=tmp;

}

int main(){

int x=10,y=20; swap(x,y); printf("%d,%d",x,y);

}

#### 6-27

程式設計實習

( B ) 5. 下列程式碼執行後的輸出為何？ (A)10，20 (B)20，10 (c)10，10 (D)20，20

void swap(int \*a, int \*b){ int tmp;

tmp=\*a;

\*a=\*b;

\*b=tmp;

}

int main(){

int x=10,y=20; swap(&x, &y); printf("%d,%d",x,y);

}

( c ) 6. 一 c 語言的程式碼如下，其執行後的結果為何？ (A)5 5 (B)6 4 (c)5 4 (D)6 5。

void FUN(int x, int \*y){ x=x+1;

\*y=\*y-1;

}

int main(){

int x=5,y=5; FUN(x, &y);

printf("%d %d",x,y);

}

( B ) 7. cIc++語言的程式碼如下，其執行後 a[2]的值為何？ (A)0 (B)1 (c)2 (D)3。

void G(int a[]){ int tmp; tmp=a[0]; a[0]=a[1];

a[1]=a[2];

a[2]=tmp;

}

int main(){

int a[3]={1,2,3}; G(a);

}

( c ) 8. 一 c 函式 G(int ＊m， int n){ … }，下列何者是呼叫該函式的正確語法？ (A)G(x， y) (B)G(＆x， ＆y) (c)G(＆x， y) (D)G(x， ＆y)。

#### 6-28

Chapter 6 函式

( c ) 9. 程式碼如下，其執行結果為何？ (A)0 (B)110 (c)55 (D)45。

int ya(int \*arr, int n){ int k;

for (int i=0;i<n;i++)

k=k+\*(arr+i); return k;

}

int main(){

int data[10];

for (int i=0;i<10;i++) data[i]=i+1;

printf("%d", ya(data, 10));

}

( c ) 10. 濂將下濪 c 語言程式碼中的缺澜處代入下面四個選項，何者無法保證在畫面上濪印出 1 2 3 4 5？ (A) f(a， 5)， (B) f(＆a[0]， 5)， (c) f(a[5]， 5)， (D) f(a+0， 5)，。

void f(const int a[], int size)

{

for (int i=0; i<size; i+=1) printf("%d", a[i]);

}

int main()

{

int a[] = {1, 2, 3, 4, 5};

？

return 0;

}



1.(c)

2.(D)

3.(A)

4.(A)

5.(B)

6.(c)

7.(B)

8.(c)

9.(c) 10.(c)

1. 函式 K 的參數是指標變數，沒有回傳值。
2. (A)指定函式的回傳值型態為指標，即可回傳位址，下列程式將變數 b 的位址回傳。 int＊ K(){ I＊ 回傳值型態為整數指標 ＊I

int b=100，

return ＆b， I＊ 回傳變數 b 的位址 ＊I

}

int main(void){ int ＊ptr = K()，

printf("％d"， ＊ptr)， I＊ 輸出指標變數 ptr 指向的內容，100 ＊I

}

(D)傳值呼叫，參數必需是一般變數。

1. 傳值呼叫，原來的變數值不變。
2. 傳址呼叫，原來的變數值會被改變。

打 表示有詳解

#### 6-29

程式設計實習

1. 函式 FUN(int x， int ＊y){ . }的參數為整數變數 x 和指標變數 y，呼叫 FUN(x， ＆y)時的引數為整數 x 和位址＆y，分別為傳值呼叫和傳址呼叫。
2. 函式 G(int a[]){ . }的參數為陣列 a[]，呼叫 G(a)時的引數為陣列名稱 a（陣列第 1 個元素的位址），在函式內以陣列索引的方式存取陣列元素，會同時改變原始陣列的內容（因為傳入的引數是位址，故在函式內是對原始陣列進行存取）。
3. 函式 ya(int ＊arr， int n){ . }的參數為指標變數 arr 和整數變數 n，呼叫 ya(data， 10)時的引數為陣列名稱 data（陣列第 1 個元素的位址）和整數 n，在函式內以指標的方式存取陣列元素，會同時改變原始陣列的內容。
4. 函式 f(const int a[]， int size)的第一個參數是陣列 a[]，因此呼叫該函式時，第 1 個引數必須是陣列名稱（陣列第 1 個元素的位址）。



( A ) 1. 執行下列 c 程式片段後，變數 I 的值為何？

(A)505 (B)455 (c)555 (D)545。

int I,J,K; I=1;

for (J=1;J<=2;J++){

K=4\*pow(I,3); I=I+K;

}

( c ) 2. 執行完下列 c 語言的程式後，印出結果為何？

(A)P( 5， 6，2) (B)11 (c)121 (D)562。

int P(int A, int B, int C){ return pow(A+B, C);

}

int main(){

int X,Y,Z; X=5;Y=6;Z=2;

printf("%d", P(X,Y,Z));

}

( D ) 3. 對於 c++ 語言特性的敘述，以下那一個錯誤？

(A)變數和陣列的儲存特性是暫時性的儲存，該應用程式結束就會消失 (B)當超出區域變數範圍時，區域變數資料將無法使用

(c)檔案可以長期維持大量資料

(D)資料放在檔案中稱為短暫瞬間資料。

#### 6-30

Chapter 6 函式

( A ) 4. 以下是 c 語言的程式碼，請問執行時會印出"x = "。 (A)4 (B)5 (c)10 (D)11。

int main(void){ int x = 0; int i = 0;

for(i; i < 5; i++){ int x = 1; x++;

}

x += i;

printf("x = %d\n", --x); return 0;

}

( D ) 5. 下列何者是使用函式的優點之一？

(A)增加程式的可讀性 (B)讓程式較結構化 (c)使程式更易於維護 (D)以上皆是。 ( B ) 6. 下列何者對遞迴函式的描述是錯誤的？

(A)允許呼叫自己的函式 (B)可縮短程式執行時間

(c)可使程式較易撰寫 (D)運用遞迴函式可讓程式可讀性較高。 ( B ) 7. 要將某一變數四捨五入，可以使用哪一個函式？

(A)sqrt( ) (B)round( ) (c)abs( ) (D)ceil( )。 ( B ) 8. 下列哪一行程式敘述，執行後的回傳值是錯的？

(A)cel(8.7)，8 (B)floor(-8.7)，-8 (c)round(8.7)，9 (D)abs(-8.7)，8。 ( c ) 9. 在 c 語言中，使用 rand( )函澷必須包含那一個標頭檔？

(A)stdio.h (B)math.h (c)stdlib.h (D)randomize.h。

( c ) 10. round(-7.8)的回傳值是多少？ (A)8 (B)7 (c)-8 (D)-7。

( D ) 11. sqrt(abs(-9))的回傳值是多少？ (A)81 (B)9 (c)-3 (D)3。

( D ) 12. 函式 cube( )能回傳立方體的體積，則 格處需塡入哪一行敘述？ (A)sqrt(a) (B)hypot(a，3) (c)round(a) (D)pow(a， 3)。

int cube(int a){ return ;

}

( A ) 13. 這行程式敘述 int x=rand( )％10，執行後，x 的值不可能為何者？

(A)10 (B)9 (c)1 (D)0。

( c ) 14. 下列程式片段執行後，s1 的內容為何？

(A)Hello (B)World (c)HelloWorld (D)HelloHello。

char s1[20]="Hello"; char s2[20]="World"; strcat(s1,s2);

#### 6-31

程式設計實習

( D ) 15. 下列程式片段執行後，s1 的內容為何？

(A)LoveArduino (B)Ardu (c)LoveArdu (D)IloveArdu。

char s1[20]="ILove"; char s2[20]="Arduino"; strncat(s1,s2,4);

( A ) 16. 接續上一題，strlen(s1)的回傳值是多少？ (A)9 (B)10 (c)11 (D)12。

( A ) 17. 函式 F( )如下，F(7)的回傳值為何？ (A)9 (B)10 (c)11 (D)12。

int F(int x){ if (x<1)

return 1; else

return F(x-2) + F(x-3);

}

( B ) 18. 一函式 F( )如下，呼叫 F( )的結果為何？ (A)X (B)Y (c)X (D)編譯錯誤。

void F(){

char s1[20]="Hello, Mary!"; char s2[20]="Hello, Ken!"; if (strcmp(s1,s2) == 0)

printf("X");

else if (strcmp(s1,s2) >0) printf("Y");

else

printf("Z");

}

( B ) 19. 一程式片段如下，G(5)的結果為何？ (A)15 (B)14 (c)13 (D)12。

int G(int x){ if (x==1)

return 0; else

return x+G(x-1);

}

( B ) 20. 一函式 K( )如下，執行 K(2)後的輸出結果為何？ (A)2 2 2 (B)2 4 6 (c)2 4 6 4 2 (D)2 4 6 6 4 2。

void K(int x){ printf("%d",x); if (x>=6)

return; else

K(x+2);

}

#### 6-32

Chapter 6 函式

（ D ) 21. 一 c 語言程式如下，執行 rec(2，2)的結果為何？ (A)1 (B)2 (c)3 (D)4。

int rec(int a, int b)

{

if (a==0 || b==0) return 1;

else

return(rec(a-1,b)+rec(a-1,b-1));

}

( B ) 22. 一函式 A( )如下，執行 A(0，1)後第 3 個印出的數字為何？

(A)1 (B)2 (c)3 (D)4。

void A(int L, int C) { if(C < 5) {

printf("%d\n", C); int T = C + L; A(C, T);

}

}

( c ) 23. 下列程式執行後，data 的值為何？ (A)10 (B)1000 (c)10000 (D)100。

void add(int \*d){ int data;

\*d = pow(\*d,2);

}

int main(){

int data=100; int \*pData;

pData = &data; add(pData);

}

( B ) 24. 在具遞迴(recursive)能力的程式語言中，哪種資料結構常用來儲存呼叫程序

(procedure)的返回位址(return address)？

(A)陣列(array) (B)堆疊(stack) (c)佇列(queue) (D)潓列(list)。 ( A ) 25. 有一 c 語言程式如下，請問執行 rec(3，3)的結果為：

(A)1 (B)2 (c)3 (D)以上皆非。

long rec(int a, int b)

{

if (a==b 廿廿 b==0) return 1;

else return(rec(a-1,b)+rec(a-1,b-1));

}

#### 6-33

程式設計實習

( A ) 26. 以下為一段 c 的程式碼，請問哪一個變數的資料型態一定與其它不同？

(A)＊a (B)b (c)c (D)＆(a[0])。

void f(int b[3]) {

int \*c = b;

}

int main(void) {

int a[] = {10, 20, 30};

f(a); return 0;

}

( D ) 27. 費式(Fibonacci)數列的定義為 Fn  Fn1  Fn2 ，且 F0  0 、F1  1。假設以下 c 程式印出來的值為 a，Fib(2)被呼叫的次數為 b，請問 a + b = ？ (A)5 (B)6 (c)7 (D)8。 int Fib(int num) {

if(num == 0) return 0;

if(num == 1) return 1;

return Fib(num-1) + Fib(num-2);

}

int main(void) { int num = 5;

printf("%d\n", Fib(num)); return 0;

}

( B ) 28. 一個函式 Func1 的定義如下，請問呼叫 Func1(1001， 255)後，回傳的值會是什麼？ (A)256 (B)255 (c)250 (D)128。

int Func1(int x, int y)

{

x=x+y; y=x-y;

x=x-y; return x;

}

( A ) 29. 一個函式 Answering 的定義如下，請問呼叫 Answering(4)時，回傳的值會是什麼？ (A)480 (B)120 (c)4 (D)0。

int Answering (int x)。

{

if (x==0)

return (4); else

return ((x+1)\*Answering(x-1));

}

#### 6-34

Chapter 6 函式

( D ) 30. 給定以下遞迴函式 fun1，請問呼叫 fun1(13597， 80)的回傳結果為何？ (A)80 (B)40 (c)13593 (D)20。

int fun1(int a, int b){ if (a % 3 == 0)

return(b); else

return(fun1(a-2 , b/2));

}

( A ) 31. 執行完下列 c 語言函式 ack(2， 2)，回傳之值為何？ (A)7 (B)8 (c)9 (D)10。

int ack(int m, int n) {

if (m==0) return (n+1);

else if (n==0) return ack(m-1, 1); else return ack(m-1,ack(m,n-1));

}

( c ) 32. 執行下列 c 語言函式 call(5)，回傳之值為何？(A)200 (B)100 (c)7 (D)5。

void f(int a, int b){ a=100; b=200;

}

int call(int a){ int b=7; f(a, b); return b;

|  |  |  |
| --- | --- | --- |
|  | | } |
| ( B ) | 33. | 以下何者，不可以是 c 語言函式的回傳型態(return type)？ |
| ( c ) | 34. | (A)void (B)int[] (c)int ＊ (D)int ＊＊。  執行下列 c 程式片段，請問會輸出什麼？ |
|  | | (A)x=60，y=60 (B)x=29，y=29 (c)x=29，y=60 (D)x=60，y=29。 |
| void swap(int a, int b) |
| { |

int tmp; tmp = a;

a = b;

b = tmp;

}

int main(void)

{

int x = 29, y = 60; swap(x ,y);

printf("x=%d,y=%d", x , y);

}

#### 6-35

程式設計實習

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ( c ) | 35. | 執行下列 c 程式片段，請問會輸出什麼？  int F1(int n)  {  if (n<=0)  return 1;  else  return F1(n-2)+F1(n-4);  }  int main()  {  printf("%d\n", F1(10)); return 0;  } | (A)11 | (B)12 | (c)13 | (D)14。 |
| ( A ) | 36. | 假設 n=5，寫出下列程式碼之執行結果的 sum(n)值為何？ | | | | |
|  |  | (A)15 (B)14 (c)13 (D)12。  unsigned int sum(unsigned int n)  { if (n==0) return 0;  else return n + sum (n-1);} | | | | |
| ( c ) | 37. | 在 c 程式中，以 y=FX(2)， 呼叫下列 FX( )函數，程式會回傳什麼數值給變數 y？ | | | | |

(A)9 (B)10 (c)11 (D)12。

int FX(int n)

{

if (n>8)

return 1; else if (n==4)

return 4+FX(n+1);

else

}

return 1+FX(n+1);

( B ) 38. 請問以下程式之執行結果為何？ (A)39 (B)45 (c)42 (D)48。

#include <stdio.h> int f2(int x)

{

static int y = 5; y++;

return (x+y);

}

int main()

{

printf("%d\n", f2(8)+f2(7)+f2(9)); return 0;

}

#### 6-36

Chapter 6 函式

( B ) 39. 請問以下程式之執行結果為何？ (A)5 (B)6 (c)7 (D)8。

#include <stdio.h> long rec(int a, int b)

{

if(a==b 廿廿 b==0) return 1;

else

return(rec(a-1,b) + rec(a-1,b-1));

}

int main()

{

printf("%ld\n",rec(4,2));

}

( B ) 40. 寫出以下程式執行後之輸出結果？ (A)504 (B)252 (c)9828 (D)5544。

#include <stdio.h> int FX(int n1,int n2)

{

while (n1 != 0 && n2 != 0)

{

if (n1 > n2)

n1 = n1 % n2; else

n2 = n2 % n1;

}

return n1+n2;

}

int main()

{

int y=FX(9828,5544);

printf("%d\n", y); return 0;

}

( c ) 41. 寫出以下程式執行後之輸出結果之 y 值為何？ (A)5 (B)8 (c)7 (D)6。

#include <iostream> #include <cstdio> using namespace std; void s2( int \*a)

{

(\*a)++;

}

int main()

{

int y = 6; s2( &y );

cout << "y=" << y ; return 0;

}

#### 6-37

Ễೣ⥑⤺૪∳

( C ) 42. ⫄ೣ₤澷(Fibonacci Number) רᮢʁӬ⳴Ⲫ〦͸⠧ḻŘ

F(0) = 0, F(1) = 1, F(i) = F(i-1)+F(i-2) for i 2: 2, ԅ F(8)᱿ϊᣅ̬ŝ (A)89 (B)55 (C)21 (D)34ȯ

( B ) 43. ࡠ⠗ʁӬ C ỄೣᦸᕀŊ⦼ۊሳⱚӛˑ㓯ŝ (A)10 (B)15 (C)5 (D)30ȯ

void f(int \*k)

{

\*k=\*k+5;

}

main()

{

int a =10; f(&a); printf("%d", a);

}

( C ) 44. ࡠ⠗ʁӬ C ỄೣᦸᕀŊ⦼ۊሳⱚӛˑ㓯ŝ (A)13 (B)32 (C)22 (D)12ȯ

int afunc(int n,int m)

{

n = n + 1; return m;

}

int main(void)

{

int m = 1, n = 2; m = afunc(m,n); printf("%d%d",m,n); return 0;

}

( D ) 45. ʁӬỄೣࡠ⠗ര᱿ⱚӛ⃌ኞᣅ̬ŝ (A)8 (B)7 (C)4 (D)5ȯ

int FX(int n)

{

if (n>7)

return 1; else if (n==3)

return 3+FX(n+1); else

return 1+FX(n+1);

}

int main(){

int y=FX(4); printf("%d",y);

}

#### 6-38

Chapter 6 函式

( D ) 46. 下列程式執行後的輸出結果為何？ (A)12 (B)15 (C)21 (D)27。

int a=15; int f(int k)

{

k=k+6; return (k+6);

}

int main()

{

int a=15; a=f(a); printf("%d", a);

}

( A ) 47. 請問下列遞迴函式在 quiz(29)時列印出幾個\*號？ (A)31 (B)30 (C)32 (D)29。

void quiz(int i) { if (i > 1) {

quiz(i / 2);

quiz(i / 2);

}

printf("\*");

}

( B ) 48. 下列 C 程式執行後顯示結果為何？ (A)X=1 (B)X=2 (C)X=3 (D)X=4。

#include <stdio.h> int main()

{ int fun(int); int x=1;

printf("X=%d\n",fun(x++));

}

int fun(int i)

{ i++; return(i); }

( D ) 49. 下列程式若執行 T(5)時，請問此函澷 T 共會被呼叫幾次？

(A)11 (B)10 (C)8 (D)9。

int T(int n)

{

if(n<3)

return 1; else

return T(n-1)+T(n-2);

}

#### 6-39

程式設計實習

( c ) 50. 執行以下之 c 程式碼後，最下面一個 f( ) 輸出的是？

(A)7 6 9 (B)7 6 8 (c)7 7 9 (D)7 7 8。

#include <stdio.h> int a=0;

f()

{

int a=6, b=7; static int c=8;

printf("%d %d %d\n", ++a, b--, c++);

}

int main(){ f();

f();

return 0;

}

( A ) 51. 寫出以下程式執行後之輸出結果為何？ (A)55 (B)54 (c)45 (D)19。

#include <iostream> using namespace std; int sum(int);

int main()

{

cout<<sum(10);

}

int sum(int n)

{

if (n == 0)

return 0; else

return n+sum(n-1);

}

#### 6-40

Chapter 6 函式

( D ) 52. 寫出以下程式執行後之輸出結果為何？ (A)12 (B)9 (c)8 (D)10。

#include <iostream> using namespace std; int FX(int);

int main()

{

cout<<FX(0);

}

int FX(int n)

{

if (n>7)

return 2; else if (n==4)

return 4+FX(n+1); else

return 1+FX(n+2);

}

( B ) 53. 下列 G( ) 為遞迴函式，G(3， 7)執行後回傳值為何？ (A)128 (B)2187 (c)6561 (D)1024。

int G (int a, int x) { if (x == 0)

return 1;

else

return (a \* G(a, x - 1));

}

( c ) 54. 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， …， a[8]=6， a[9]=7，以 f(a， 10) 呼叫執行下列函式後，回傳值為何？ (A)1 (B)2 (c)7 (D)9。

int f (int a[], int n) { int index = 0;

for (int i=1; i<=n-1; i=i+1) { if (a[i] >= a[index]) {

index = i;

}

}

return index;

}

#### 6-41

程式設計實習

( c ) 55. 下列 g(4) 函式呼叫執行後，回傳值為何？ (A)6 (B)11 (c)13 (D)14。

int f(int n) { if (n > 3) {

return 1;

}

else if (n == 2) { return (3 + f(n+1));

}

else {

return (1 + f(n+1));

}

}

int g(int n) { int j = 0;

for (int i=1; i<=n-1; i=i+1) { j = j + f(i);

}

return j;

}

( D ) 56. 下列 Mystery( ) 函式 else 部分運算式應為何，才能使得 Mystery(9) 的回傳值為 34？ (A)x + Mystery(x-1) (B)x ＊ Mystery(x-1)

(c)Mystery(x-2) + Mystery(x+2) (D)Mystery(x-2) + Mystery(x-1)

int Mystery (int x) { if (x <= 1) {

return x;

}

else {

return

}

}

( c ) 57. 若以 B(5，2) 呼叫下列 B( ) 函式，總共會印出幾次"base case"？ (A)1 (B)5 (c)10 (D)19。

int B(int n, int k) {

if (k == 0 || k == n){ printf("base case\n"); return 1;

}

return B(n-1,k-1) + B(n-1,k);

}

#### 6-42

Chapter 6 函式

( D ) 58. 若以 G(100) 呼叫下列函式後，n 的值為何？ (A)25 (B)75 (c)150 (D)250。

int n = 0;

void K (int b) { n = n + 1; if (b % 4)

K(b+1);

}

void G (int m) {

for (int i=0; i<m; i=i+1) { K(i);

}

}

( A ) 59. 小藍寫了一段複雜的程式碼想考考你是否了解函式的執行流程。請回答程式最後輸出的數值為何？ (A)70 (B)80 (c)100 (D)190。



int g1 = 30, g2 = 20; int f1(int v) {

int g1 = 10; return g1+v;

}

int f2(int v) { int c = g2; v = v+c+g1; g1 = 10;

c = 40;

return v;

}

int main() {

g2 = 0;

g2 = f1(g2);

printf("%d", f2(f2(g2))); return 0;

}

( c ) 60. 若以 F(5，2) 呼叫下列 F( ) 函式，執行完畢後回傳值為何？

(A)1 (B)3 (c)5 (D)8。

int F (int x,int y) { if (x<1)

return 1;

else

}

return F(x-y,y)+F(x-2\*y,y);

#### 6-43

程式設計實習

( A ) 61. 若已經定義一函式 mySwap 如左，則右之程式碼片段執行後，變數 x 及 y 的值分別為何？

(A)x = 3；y = 4

(B)x = 4；y = 3 (c)x = 3；y = 3 (D)x = 4；y = 4。

void mySwap(int a, int b){ int x = 3, y = 4; int temp; mySwap(x, y); temp = a;

a = b;

b = temp;

}

( D ) 62. 若自訂函式 myFunc 需要傳回一個整數陣列，則 myFunc 函式的原型宣告應為下列何者？

(A)void myFunc()，

(B)int myFunc()，

1. int [] myFunc()， (D)int ＊ myFunc()，。



( A ) 1. 一個 *RC* 串聯電路，計算電容充電時的暫態電壓公式為*Vc* = *E*(1一*e*

一  *t*

*RC* ) ，其中 *e* 為自

然對數的底數，若要使用 c 語言程式完成此算式，則下列哪一個函式最適合被使用？

* 1. exp( ) (B)log( ) (c)sqrt( ) (D)pow( )。

( B ) 2. 某大型科技公司今年業績創歷年新高，董事長郭董打算自掏腰包，提供一棟陽明山獨棟別墅，作為今年公司尾牙的第一大獎，為求公平，郭董要求資訊部張經理親自撰寫一抽籤程式，使用亂數隨機產 員工編號，該公司共有 800 名員工，員工編號由 1~800 號，請問下列哪一個才是正確的 c 語言指令？

1. rand( )％800， (B) rand( )％800+1， (c) rand( )I800， (D) rand( )I800+1，。

( c ) 3. 承上題，張經理發現每次執行程式，產 的員工編號都是 20 號，而不是隨機產 的數字，距離尾牙只剩 3 天，請問，發 此問題最有可能的原因是？

1. 沒有引用正確的函式庫<stdlib.h> (B)電腦主機沒有放一包綠色乖乖 (c)沒有使用初始化亂數函式 srand( ) (D)張經理的員工編號是 20。

#### 6-44

Chapter 6 函式

( D ) 4. 在 c 語言中，遞迴函式指的是在函式中呼叫自己，輾轉相除法、費氏數列或河內塔問題，都是使用遞迴函式的經典範例，這星期的程式設計實習作業，老師要求使用遞迴的方式，撰寫一計算 n 階層的遞迴函式，彤彤寫的函式 K( )如下，請問第 50 行應塡入哪一個指令？

1. n＊K(n) (B)K(n-1) (c)K(n)＊K(n-1) (D)n＊K(n-1)。

10 int K(int n){

20 if (n==1)

30 return 1;

40 else

50 return ? ;

60 }

題組：賭神高進與新加坡賭王陳 城在公海的一場睹局中，以高超的賭技獲勝，陳 城也被香港皇家冨察拘補，隨後睹神收陳小刀為弟子，帶小刀到美國拉斯維加斯學習賭技。陳小刀到了美國後，除了跟隨師父學習賭技外，同時也在大學進修資訊工程學位，他了解函式可以幫助程式模組化，也可以讓程式更易於維護，若能善用各種公用函式，在程式開發的過程，更能達到事半功倍的效果。

( c ) 5. 請問關於 c 語言函式的說明，下列何者不正確？

(A)要使用公用函式，需利用#include 指令引用相關的函式庫

(B)自訂函式不一定要有回傳值 (c)函式不可以呼叫自己

(D)在函式內宣告的變數，稱為區域變數。

( B ) 6. 他嘗試撰寫一支擲骰子函式 dice()如下，參數 n 表示擲出的骰子數量，例如呼叫 dice(2)，會模擬擲出 2 顆骰子，亂數產 介於 2~12 之間的整數後回傳，但編譯時卻出現 [Error] 'rand' was not declared in this scope 的錯誤訊息，請問他應該引用哪一個函式庫，才能正確編譯？

(A)math.h (B)stdlib.h (c)time.h (D)stdio.h。

int dice(int n){

return rand()%(n\*5+1)+n;

}

( A ) 7. 正所謂小賭怡情養性，大賭家破人亡，在一次賭王爭霸賽中，小刀不幸慘敗，輸光所有家產，只好跟師父高進借錢週轉，兩人約定利息採複利計算，中途不需還款，於約定日一次歸還本 和利息，本利和的計算公式如下，請問小刀若要使用 c 語言撰寫本利和計算的程式，下列哪一個公用函式最適合使用？

(A)double pow(double x， double y)

(B)double hypot(double x， double y) (c)double exp(double x)

(D)double sqrt(double x)。

FV=PV\*(1+r)n

FV：需歸還本利和 PV：借款 額 r：年利率 n：借款期數(年)

#### 6-45

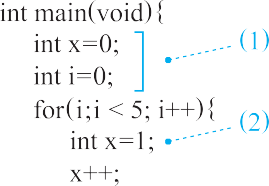
程式設計實習



打 表示有詳解

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.(A) | 2.(C) | 3.(D) | 4.(A) | 5.(D) | 6.(B) | 7.(B) | 8.(B) | 9.(C) | 10.(C) |
| 11.(D) | 12.(D) | 13.(A) | 14.(C) | 15.(D) | 16.(A) | 17.(A) | 18.(B) | 19.(B) | 20.(B) |
| 21.(D) | 22.(B) | 23.(C) | 24.(B) | 25.(A) | 26.(A) | 27.(D) | 28.(B) | 29.(A) | 30.(D) |
| 31.(A) | 32.(C) | 33.(B) | 34.(C) | 35.(C) | 36.(A) | 37.(D) | 38.(B) | 39.(B) | 40.(B) |
| 41.(C) | 42.(C) | 43.(B) | 44.(C) | 45.(D) | 46.(D) | 47.(A) | 48.(B) | 49.(D) | 50.(C) |
| 51.(A) | 52.(D) | 53.(B) | 54.(C) | 55.(C) | 56.(D) | 57.(C) | 58.(D) | 59.(A) | 60.(C) |
| 61.(A)  ·素簣題  1.(A) | 62.(D)  2.(B) | 3.(C) | 4.(D) | 5.(C) | 6.(B) | 7.(A) |  |  |  |

I解斬I



1. 程式的執行步驟如下，
   1. 宣告區域變數 x 和 i，
   2. 在 for 迴圈內宣告區域變數 x，因為 main()的區域變數 x 與 for 迴圈的區域變數 x 存取範圍不同，故允許同名，當 for 迴圈結束後，迴圈內的區域變數 x 即消失。
   3. 離開迴圈後 i=5，x=x+i=0+5=5（x 為 main()的區域變數），程式碼 printf("x = %d\n", --x); 會先將x

減 1 後再輸出，故輸出 4。

1. 使用遞迴函式會花費較長執行時間，且需較多記憶體 間。

8. floor(-8.7)回傳-9。

1. 回傳字串長度，包含結束字元’\0’。
2. 參考單元 6-2 立即練習第 12 題。

|  |  |
| --- | --- |
| 條件式 | 回傳值 |
| s1==s2 | 0 |
| s1>s2 | 1 |
| s1<s2 | -1 |

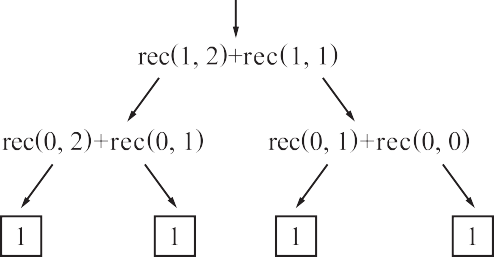
1. 公用函式 strcmp(s1, s2)會比較兩個字串（逐一比較每個字元的 ASCII 碼），函式的回傳值如右，

程式中 s1 和 s2 的前 7 個字元皆相同，第 8 個字元分別為’M’和’K’，’M’的 ASCII 碼為 0x4D，’K’的 ASCII 碼為 0x4B，因為’M’的 ASCII 碼大於’K’的 ASCII 碼，故回傳值為 1。

#### 6-46

Chapter 6 函式

1. 呼叫 G(5)的過程如下， G(5)



5+G(4)

5+4+G(3)

5+4+3+G(2)

5+4+3+2+G(1)

5+4+3+2+0

1. 呼叫 K(2)後會輸出 2 4 6，執行過程如下， K(2)

K(4)

K(6)

1. 呼叫 rec(2， 2)的過程如右圖。
2. 呼叫 A(0， 1)的過程如下，
3. 函式 add(int ＊d){ . }的參數為指標變數 d，呼叫函式 add(pData)時，傳入的引數 pData 為指標變數（指向變數 data），呼叫後，參數 d 也會指向變數 data，屬於傳址呼叫。
4. 參考第 21 題解析。
5. 選項(A)，陣列名稱 a 為第 1 個元素的位址，「＊j為取值運算子，因此＊a 等於 a[0]，

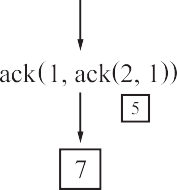
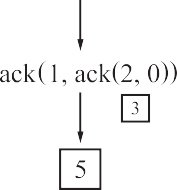
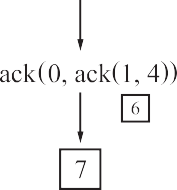
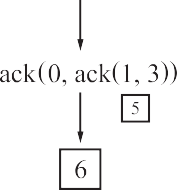
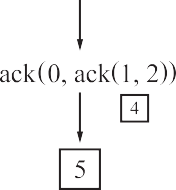
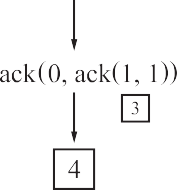
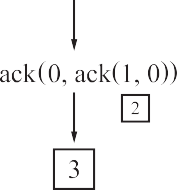
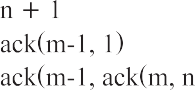
選項(B)(c)(D)都是陣列第 1 個元素的位址，且 a、b、c 都指向相同位址（執行函式 f(a)後，陣列 a[]和 b[]是位於同樣記憶體位置的相同陣列）

1. 參考單元 6-2 的範例 4，
   1. 執行 Fib(5)的回傳值為 5，即 a=5，
   2. 執行 Fib(2)，函式 Fib()被呼叫 3 次，即 b=3，
   3. a+b=5+3=8

#### 6-47

程式設計實習

1. 呼叫 Answering(4)的過程如下， Answering(4)



5＊Answering(3)

5＊4＊Answering(2)

5＊4＊3＊ Answering(1)

5＊4＊3＊2＊Answering(0)

 5＊4＊3＊2＊4

1. 呼叫 fun1(13597， 80)的過程如下， fun1(13597， 80)

fun1(13595， 40)

fun1(13593， 20)，終止條件(a％3==0)成立，回傳 20。

1. 此程式以遞迴方式實作數學上有名的阿克曼函數（Ackermann Function），定義如下，

為能順利求解，我們從較小的參數開始，計算函式的回傳值，經過耐心且辛苦的推導之後，最後可得到

ack(2， 2)的回傳值為 7。

(1) ack(1， 0)=2 (2) ack(1， 1)=3 (3) ack(1， 2)=4

(4) ack(1， 3)=5 (5) ack(1， 4)=6 (6) ack(1， 5)=7

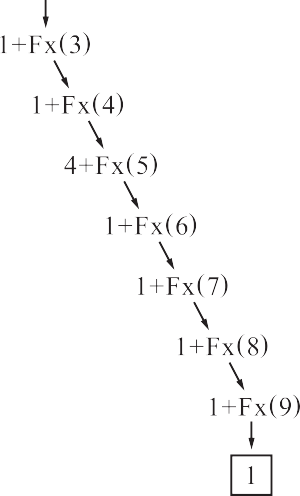
(7) ack(2， 0)=3 (8) ack(2， 1)=5 (9) ack(2， 2)=7

1. 在函式 call()中以傳值呼叫的方式呼叫 f(a， b)，不會改變原始變數值，call()的回傳值 b 仍為 7。
2. c 語言不可直接回傳陣列。
3. 傳值呼叫，在函式內改變參數值，來源變數的值不會改變。
4. 參考單元 6-2 立即練習第 12 題。

#### 6-48

Chapter 6 函式

1. 此程式以遞迴方式實作 n+(n-1)+(n-2)+ . +1+0，故 sum(5) = 5+4+3+2+1+0 = 15。



1. 呼叫 FX(2)的執行過程如下，
2. 變數宣告時加上 static，該變數稱為靜態變數，與區域變數不同，靜態變數的 命周期會維持到程式結束，故變數 y 不會因函式 f2()結束而消失，程式執行過程如下，

|  |  |  |  |
| --- | --- | --- | --- |
|  | x | y |  |
| 呼叫 f2(8) | 8 | 5 | static int y=5， IIy 為靜態變數 |
|  |  | 6 | y++， |
|  |  |  | retrun (x+y)， II回傳 x+y=8+6=14 |
| 呼叫 f2(7) | 7 | 6 | static int y=5， IIy 為靜態變數，其值仍為 6 |
|  |  | 7 | y++， |
|  |  |  | retrun (x+y)， II回傳 x+y=8+6=14 |
| 呼叫 f2(8) | 9 | 7 | static int y=5， IIy 為靜態變數，其值仍為 7 |
|  |  | 8 | y++， |
|  |  |  | retrun (x+y)， II回傳 x+y=9+8=17 |
|  |  |  | 輸出 f2(8)+f2(7)+f2(8)=14+14+17=45 |

1. 參考第 21 題解析。
2. 同學不要怕，此題目看似複雜，但其實不難，只是虛張聲勢而已，不過程式中有用到運算式、條件判斷、迴圈、函式呼叫等，是很棒的整合式題型。
3. 函式 s2(int ＊a){ . }的參數為指標變數，呼叫 s2(＆y)時的引數為變數 y 的位址，s2()被呼叫後，參數 a

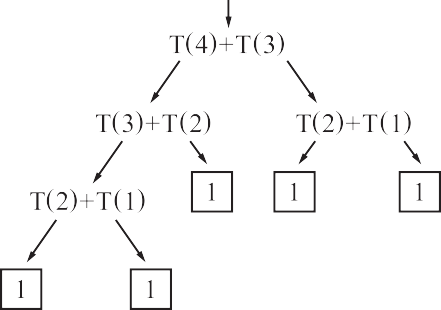
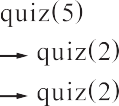
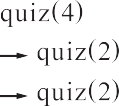
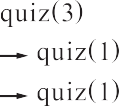
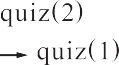
會指向變數 y，此為傳址呼叫，在函式內的操作，會改變原始變數 y 的值。

1. 參考第 41 題解析。
2. 呼叫 afunc()傳入引數為 m 和 n，但 afunc()的接收參數是 n 和 m。
3. 參考第 37 題解析。
4. 參考 6-2 立即練習第 11 題。

#### 6-49

程式設計實習

1. 為能順利求解，我們從較小的參數開始，推導在不同的參數下，函式 quiz()會輸出幾個"＊"， quiz(1)：輸出 1 個"＊"



ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ

quiz(2)：輸出 3 個"＊"

ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ

quiz(3)：輸出 3 個"＊"

ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ

quiz(4)：輸出 7 個"＊"

ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ

quiz(5)：輸出 7 個"＊"

ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ

quiz(6)：輸出 7 個"＊" quiz(7)：輸出 7 個"＊" quiz(8)：輸出 15 個"＊" quiz(9)：輸出 15 個"＊"

|  |  |
| --- | --- |
| n | 輸出"＊"數量 |
| 1 | 1 |
| 2~3 | 3 |
| 4~7 | 7 |
| 8~15 | 15 |
| 16~31 | 31 |

.

quiz(16)：輸出 31 個"＊"

.

quiz(29)：輸出 31 個"＊"

.

程式之路一向是 獨而漫長的，在經過一段艱辛的推導過程後，我們終於可以找出程式執行的規律性，得知參數 n 與輸出"＊"數量的關係如右上表，由表可知，quiz(29)會輸出 31 個"＊"。

1. 指令 fun(x++)會先呼叫 fun(x)，等 fun()執行完畢返回後，才執行 x++，故 fun(x++)傳入的引數是 1，回傳值是 2。
2. 請注意題目問的是函式 T()被呼叫幾次，而不是 T(5)的執行結果，執行 T(5)的過程如右。
3. (1) 在函式 f()內宣告的變數 a 和 b 為區域變數，函式結束即消失，而變數 c 在宣告時加上 static，稱為靜態變數，與一般區域變數不同，靜態變數的 命周期會維持到程式結束，故變數 c 不會因函式結束而消失。

#### 6-50

Chapter 6 函式

* 1. 指令 printf("％d ％d ％d＼n"， ++a， b--， c++)，的結果與下列程式碼相同 ： a=a+1，

printf("％d ％d ％d＼n"， a， b， c)， b=b-1，

c=c+1，

* 1. 呼叫 f()時的步驟與變數變化如下，

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | c |  |
| 第 1 次呼叫 f() | 6  7  X | 7  6  X | 8  9 | int a=6， b=7，  static int c=8， II第一次宣告變數 c，其值為 8 a=a+1，  printf("％d ％d ％d＼n"， a， b， c)， II輸出 7 7 8  b=b-1， c=c+1，  IIf()結束，變數 a 和 b 消失，但變數 c 仍然存在 |
| 第 2 次呼叫 f() | 6  7  X | 7  6  X | 9  10 | int a=6， b=7，  static int c=8， IIc 為靜態變數，其值仍為 9 a=a+1，  printf("％d ％d ％d＼n"， a， b， c)， II輸出 7 7 9  b=b-1， c=c+1，  IIf()結束，變數 a 和 b 消失，但變數 c 仍然存在 |

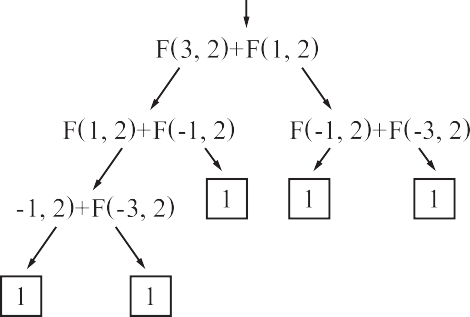
1. 此程式以遞迴方式實作 n+(n-1)+(n-2)+ . +1+0，故 sum(10) = 10+9+8+ . +1+0 = 55。
2. 參考第 37 題解析。
3. G(a， x)=a＊G(a， x-1)=a＊a＊G(a， x-2)=a＊a＊a＊G(a， x-3) = a＊a＊a＊ . ＊G(a， 0)，可知函式 G(a， x)的回傳值為 ax，故 G(3， 7)會回傳 37=2187。
4. 函式 f(int a[]， int n){ . }以陣列方式接收陣列資料，呼叫 f(a， 10)時，傳入的引數分別為陣列名稱和陣列大小，在函式內使用陣列索引的方式走訪陣列，當函式結束後，回傳值 index 為陣列中最大元素值的索引。
5. 參考第 37 題解析。
6. 遞迴函式 Mystery(int n)會回傳費式數列的第 n 個數，參考單元 6-2 範例 4。
7. 參考第 21 是解析。
8. 執行 G(100)，會進入 for 迴圈，變數 i 從 0 到 99，依序呼叫函式 K(i)，程式執行過程如下，可以發現 i從 1 開始，每經過 4 次迴圈，n 就會加 10，將 i 為 1~4 視為一組，5~7 視為一組，97、98、99 和 0 視為一組，故共有 100I4=25 組，每一組都將 n 加 10，所以 n=25＊10=250。

|  |  |  |
| --- | --- | --- |
| i | n＝0 | 呼叫 K（i）的過程 |
| 0 | +1 | K(0) |
| 1    2    3    4 | +4 | K(1)K(2)K(3)K(4)    K(2)K(3)K(4) K(3)K(4)  K(4) |
| +3 |
| +2 |
| +1 |
| 5    6    7    8 | +4 | K(5)K(6)K(7)K(8)    K(6)K(7)K(8) K(7)K(8)  K(8) |
| +3 |
| +2 |
| +1 |
| . |  | . |
| 97    98    99 | +4 | K(97)K(98)K(99)K(100)  K(98)K(99)K(100) K(99)K(100) |
| +3 |
| +2 |

#### 6-51

程式設計實習

1. 此程式執行過程如下，需注意在函式 f1()內有宣告一區域變數 g1，不要和全域變數 g1 混淆，



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 全域變數 | | 區域變數 | | |  |
| g1＝30 | g2＝20 | g1＝30 | g2＝20 | g1＝30 |
|  | 0  10 | X | 0  X | 10  X | g2=0， g2=f1(g2)，  II執行函式 f1(0)  int g1=10，  return g1+v， II回傳值 10，指定給 g2  II函式 f1()執行結束，區域變數消失 |
|  |  |  |  |  | printf("％d"， f2(f2(g2)))， II先執行函式 f2(10) int c=g2，  v=v+c+g1， g1=10， c=40，  return v， II回傳值 50  II函式 f2()執行結束，區域變數消失 II再執行函式 f2(50)  int c=g2，  v=v+c+g1， g1=10， c=40，  return v， II回傳值 70  II函式 f2()執行結束，區域變數消失 |
| 10 |  | 10  40  X | 10  50  X |  |
| 10 |  | 10  40  X | 50  70  X |  |
|  |  |  |  |  | II輸出 70 |

1. 執行 F(5， 2)的過程如下，
2. 參考 6-3 立即練習第 4 題。
3. c 語言的函式無法直接回傳陣列，變通的作法是回傳陣列的位址，依題意，若要傳回一個整數陣列，可

宣告函式原型的回傳值型態為整數指標(位址)，函式實作只需回傳陣列名稱即可(陣列名稱為該陣列第 1

個元素的位址)，函式原型與實作的範例如下，

int＊ myFunc()， II函式原型

int＊ myFunc(){ II函式實作 int array[] = {1，2，3}，

return array， II回傳陣列第 1 個元素的位址

}

#### 6-52

Chapter 6 函式



( D ) 1. 下列是一個用 c 的語法寫成的程式，請問程式執行結束時，變數 value 之值

=一一一一一一一一一。

(A)1 (B)4 (c)6 (D)8。 【102 年工科技藝競賽】

void main(){

int value=1, list[5]={0,2,4,6,8};

swap(&list[2],&list[3]); swap(&value,&list[3]); swap(&value,&list[value]);

}

void swap(int \*a, int \*b){ int t; t=\*a;\*a=\*b;\*b=t;

}

( c ) 2. 下列哪一種演算法採用重複呼叫自己，直至達到設定條件滿足後才逐層跳脫。

(A)搜尋 (B)反覆

(c)遞迴 (D)排序。 【102 年工科技藝競賽】

( B ) 3. 程式設計時，區域變數(local variable)是指 (A)宣告在某個程式檔中的所有變數

(B)宣告在副程式內的變數

(c)副程式呼叫時，呼叫者傳遞給受呼叫者之變數

(D)副程式呼叫時，受呼叫者回傳給受呼叫者的值。 【102 年工科技藝競賽】

( c ) 4. 在下列程式片段中，其執行之輸出結果為何？

(A)EDcBA (B)EABcBcc

(c)DcBAE (D)EABcD。 【103 年工科技藝競賽】

char s1[10] = "DCBA";

char s2[] = "E"; puts(strcat(s1, s2));

( B ) 5. 下列何者不是 ANSI c 之標準副程式函數？

(A)printf (B)main

(c)scanf (D)pow。 【103 年工科技藝競賽】

( B ) 6. 假如 a = 7.0， b = 7.0， c = 6.0， 執行 printf( "％.2f"， sqrt( a + b ＊ c ) )，？

(A)49 (B)7.00

1. 7 (D)49.00。 【104 年工科技藝競賽】

( B ) 7. 如果呼叫下面這個函式 mystery，並傳入參數 4 給 number，請問這個函式的回傳值為何？

* 1. 1 (B)24 (c)0 (D)4。 【104 年工科技藝競賽】

int mystery(int number) { if (number <= 1)

return 1; else

return number\*mystery(number-1);

【110 年工科技藝競賽】

}

#### 6-53

程式設計實習

( D ) 8. 下列何者為傳址呼叫(call by address)的特點？ (A)以額外記憶體存放副程式的參數

* 1. 主程式參數值不因副程式改變而變

(c)主程式參數名稱取代副程式相對應形式參數名稱

(D)主副兩程式參數佔用相同記憶體位址。 【106 年工科技藝競賽】

( c ) 9. 執行下列 c 程式片段，請問輸出的第 9 個數值是？

(A)362880 (B)720 (c)40320 (D)5040。 【106 年工科技藝競賽】

#include <stdio.h>

unsigned long int test(unsigned int n){ unsigned long int result;

if (n==0) result=1;

else result=n\*test(n-1); return result;

}

void main(){

unsigned int j,N=15;

for(j=0;j<N;++j) printf("%lu\n", test(j));

}

( B ) 10. 假如 a=9.0， b=8.0 以及 c=9.0， 請問執行下列程式後，會顯示什麼？

(A)8.00 (B)9.00 (c)10.00 (D)81.00。 【107 年工科技藝競賽】

printf("%.2f", sqrt(a+b\*c));

( c ) 11. 考慮下列 c 函式，請問以 func(15) 呼叫後回傳值為何？

(A)30 (B)32 (c)34 (D)36。 【107 年工科技藝競賽】

int func(int n){

if (n<=2)

return n; else

return func(n-1)/2 + func(n-2);

}

( A ) 12. 若一仿 c 程式如下，若輸入 n 值為 3，當程式執行結束結果值為何？

(A)12 (B) 3 (c) 6 (D)程式無法停止。 【107 年工科技藝競賽】

int X(int n){

if (n<1)

return 3; else

return 3+X(n-1);

}

( B ) 13. 針對以下 c++程式，f(8，3)輸出是哪一個值？

(A)3 (B)5 (c)8 (D)11。 【107 年工科技藝競賽】

int f(int x, int y){ if (x==y) return 0;

else return f(x-1, y)+1;

}

#### 6-54

Chapter 6 函式

( A ) 14. 呼叫下列的函式，fibonacci(2)答案為何？

(A)1 (B)2 (c)3 (D)4。 【107 年工科技藝競賽】

int fibonacci(int n)

{

if (n==0||n==1) return n;

else

return fibonacci(n-1) + fibonacci(n-2);

}

( c ) 15. 呼叫下列的函式，fibonacci(6)答案為何？

(A)3 (B)5 (c)8 (D)13。 【107 年工科技藝競賽】

int fibonacci(int n)

{

if (n==0||n==1) return n;

else

return fibonacci(n-1) + fibonacci(n-2);

}

( B ) 16. 給予一個整數陣列 int n[10]={3，5，7，9，11，14，17，33，44，50}， 呼叫下圖的函式 bsearch(n， 44， 0， 9) 來搜尋 44，請問 while 迴圈內的指令 middle = (low+high)I2，會被執行幾次？ (A)2 (B)3 (c)4 (D)5。 【107 年工科技藝競賽】

int bsearch(int b[], int Key, int low, int high){ int middle;

while (low <= high){ printf("%d", 1); middle=(low+high)/2; if (Key == b[middle])

return middle;

else if (Key < b[middle]) high = middle -1;

else

low = middle+1;

}

return -1;

}

( B ) 17. 同上題，呼叫函式 bsearch(n， 13， 0， 9) 來搜尋 key 13，請問 while 迴圈內的指令 middle

= (low+high)I2，會被執行幾次？

(A)2 (B)3 (c)4 (D)5。 【107 年工科技藝競賽】

( c ) 18. 同上題，呼叫函式 bsearch(n， 55， 0， 9) 來搜尋 key 55，請問 while 迴圈內的指令 middle

= (low+high)I2，會被執行幾次？

(A)2 (B)3 (c)4 (D)5。 【107 年工科技藝競賽】

#### 6-55

程式設計實習

( A ) 19. 執行下列 c 程式後，請問最後 z 輸出是？

(A)1 (B)2 (c)13 (D)26。 【107 年工科技藝競賽】

#include <stdio.h> int abc(int x, int y){

int tmp;

while (x%y != 0){ tmp=y; y=x%y; x=tmp;

}

return y;

}

int main(){

int z=abc(159, 52); printf("%i", z); return 0;

}

( c ) 20. 執行下列 c 程式後，請問最後 z 輸出是？

(A)5 (B)8 (c)13 (D)21。 【107 年工科技藝競賽】

#include <stdio.h> int def(int x){

if (x==1) return 1; else{

int i, B=0, A=1, O, S=0;

for (i=2;i<=x;i++){ S=A+B; O=S; B=A; A=S;

}

return O;

}

}

int main(){

int z=def(7); printf("%i", z); return 0;

}

( c ) 21. 下列何者為 c 語言函式，傳回字串長度？

(A)strcpy (B)lencat (c)strlen (D)strcmp。

#### 6-56

Chapter 6 函式

( c ) 22. 以下為 c 語言之主程式呼叫副程式，請呈現 printf 敘述所顯示 x y z 的結果？

(A) x=12 y=8 z=4 (B)x=16 y=11 z=5 (c)x=12 y=11 z=5 (D)x=16 y=11 z=4。

【108 年工科技藝競賽】

main(){

int x=12,y=8,z=4; SubProgram(x, &y, &z);

printf("x=%d, y=%d, z=%d\n", x, y, z);

}

SubProgram(int x, int \*y, int \*z){ x=x+\*y-\*z;

\*y=x-\*z-1;

\*z=x-\*y;

}

( c ) 23. 執行下列程式碼後，請問輸出結果為何？

(A)24 (B)180 (c)360 (D)720。 【108 年工科技藝競賽】

#include <stdio.h> int test(int a){

if (a==1) return 1;

else return a\*test(a-1);

}

int main(){

int x=5, y=2;

int z=test(x)/test(y)\*test(x-y); printf("%d", z);

return 0;

}

( B ) 24. 執行下列程式碼後，請問輸出結果為何？

(A)110 (B)68 (c)38 (D)166。 【108 年工科技藝競賽】

#include <stdio.h> int TestYou(int a){

if (a==1) return 0;

else return TestYou(a-1)+a\*(a+1);

}

int main(){

int a=5;

int b=TestYou(a); printf("%i", b);

}

#### 6-57

程式設計實習

( D ) 25. 執行下列程式碼後，請問輸出結果為何？

(A)153 (B)306 (c)68 (D)34。 【108 年工科技藝競賽】

#include <stdio.h>

int TestYou(int a, int b){ if (b==0) return a;

else return TestYou(b, a%b);

}

int main(){

int c=TestYou(306, 68); printf("%i", c);

}

( A ) 26. 執行下列程式碼後，請問輸出結果為何？

(A)128 (B)256 (c)512 (D)511。 【108 年工科技藝競賽】

#include <stdio.h>

int TestYou(int a, int b){ if (b==0) return 1;

if (b==1) return a;

else return (a\*TestYou(a, b-1));

}

int main(){

int c=TestYou(2,7); printf("%i", c);

}

( c ) 27. 有一 c 程式片段如下，其中 round( )是四捨五入的函數，執行的結果下列何者正確？ (A)-1 (B)0 (c)1 (D)2。 【108 年工科技藝競賽】 #include <stdio.h>

#include <string.h> #include <math.h> main(){

float dataB[4]={2.2, 2.8, -2.4, -1.8};

int j, result=0; for(j=0;j<4;j++){

result = result + round(dataB[j]);

}

printf("%d", result);

}

( B ) 28. 在下列程式片段中，呼叫 printDigit(n)，輸入 n 為 1234，請問輸出為何？

(A). 10 (B). 4321 (c). 1234 (D). 0。 【109 年工科技藝競賽】

void printDigit(int n){ printf("%d", n%10); if (n>10){

printDigit(n/10);

【110 年工科技藝競賽】

}

}

#### 6-58

Chapter 6 函式

( A ) 29. 在下列程式片段中，執行結果為何？ 【109 年工科技藝競賽】

(A). a = 10， b = 20 (B). a = 10， b = 10 (c). a = 20， b = 20 (D). a = 20， b = 10。

#include <stdio.h>

void swap(int a, int b){ int temp;

temp = a;

a = b;

b = temp;

}

int main(){

int a=10, b=20;

swap(a, b);

printf("a = %d, b = %d\n", a, b);

return 0;

}

( c ) 30. 下列程式片段中，f(4)的輸出應該為何？

(A). 2 (B). 4 (c). 7 (D). 11。 【109 年工科技藝競賽】

int f(int n){

if (n==1)

return 1; else

return f(n-1) + n - 1;

}

( c ) 31. 呼叫下列的函式，fibonacci(6)答案為何？

(A). 3 (B). 5 (c). 8 (D). 13。 【109 年工科技藝競賽】

int fibonacci(int n){

if (n==0 || n==1) return n;

else

return fibonacci(n-1)+fibonacci(n-2);

}

( A ) 32. 請問下列程式片段執行後，會印出甚麼？

(A). 729 (B). 3 (c). 243 (D). 128。 【109 年工科技藝競賽】

#include <stdio.h>

int FunctionA(int x, int y){ if (y==0) return 1;

if (y==1) return x;

else return (x \* FunctionA(x,y-1));

}

int main(){

int c=FunctionA(3,6); printf("%i\n", c);

}

#### 6-59

程式設計實習

( c ) 33. 有一程式如下所示，執行後，顯示值為何？ 【110 年工科技藝競賽】

(A)0 (B)1 (c)6 (D)9。

#include <stdio.h> int f(int a[], int n){

int index = 0;

for (int i=1; i<=n-1; i=i+1){ if (a[i] >= a[index]){

index = i;

}

}

return index;

}

int main(void) {

int a[9]={1,2,3,4,7,5,9,6,8};

int ret=0; ret=f(a, 9);

printf("%d", ret);

}

( D ) 34. 下列程式執行結果為何？ 【110 年工科技藝競賽】

(A)10 20 10 (B)10 10 10 (c)10 10 20 (D)10 11 10。

#include <stdio.h> int s=1;

void add(int); int main() {

int s=10; printf("%d ", s); add(s); printf("%d ", s); return 0;

}

void add(int a){ s = s+a;

printf("%d ", s);

}

#### 6-60

Chapter 6 函式

( C ) 35. 下列程式執行結果為何？ 【110 年工科技藝競賽】

(A)10 20 (B)11 21 (C)11 11 (D)21 21。

#include <stdio.h> int s=1;

void add(int); int main() {

int s=10; add(s);

add(s); return 0;

}

void add(int a){ int s=1;

s = s+a; printf("%d ", s);

}

( C ) 36. 請問下列程式執行後，輸出的數值是？ 【110 年工科技藝競賽】

(A)2 3 (B)40 3 (C)2 40 (D)40 2。

#include <stdio.h>

void funC(int\* o, int q){ int p=5;

q=3+p;

\*o=p\*q;

}

int main() {

int f=2, s=3; funC(&s, f);

printf("%4d%4d", f, s);

}

( C ) 37. 下列 C 語言程式碼片段執行後，變數 result 的值為何？

(A)0.0 (B)7.0 (C)7.5 (D)15.0。 【111 年統測】



( D ) 38. 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);。

【111 年統測】

#### 6-61

程式設計實習

( B ) 39. 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？ 【112 年統測】

1

1. #include <stdio.h>
2. //void sub(int i, char \*s);
3. int main(int argc, char \*argv[]) {
4. sub(argc, argv[2]);
5. return 0;

7 }

8

1. void sub(int i, char \*s){
2. printf("total %d arguments, and the 2nd one is %s\n", i, s);

11 }

* 1. 將行號 4 中 main(int argc, char \*argv[] )改為 main()
  2. 去掉行號 3 最前面的註解標記//
  3. 將行號 1 的 白行刪除
  4. 在行號 1 新增#include <stdlib.h>。

A閱讀下文，回答第 40~42 題

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 | } |  |

#### 6-62

Chapter 6 函式

( D ) 40. 下列何者為程式執行結果？

(A)11, 11 (B)13, 10 (C)12, 11 (D)12, 10。 【112 年統測】

( A ) 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？ 【112 年統測】

1. 行號 11 的 x 改為&x，並將函式 inc( )中所有的 xin 全部改為\*xin
2. 行號 11 的 x 改為\*x，並將函式 inc( )中所有的 xin 全部改為&xin
3. 行號 11 的 x 改為&x，並將函式 inc( )中所有的 xin 全部改為&xin
4. 行號 11 的 x 改為\*x，並將函式 inc( )中所有的 xin 全部改為\*xin。

( A ) 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？

1. 行號 2 的 sum 是全域變數，行號 4 的 sum 是區域變數
2. 行號 2 的 sum 是區域變數，行號 4 的 sum 是全域變數
3. 行號 2 的 sum 和行號 10 的 sum 都是區域變數
4. 行號 2 的 sum 和行號 10 的 sum 都是全域變數。 【112 年統測】

( C ) 43. 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 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？ 【113 年統測】

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 }

1. 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
2. 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
3. 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

1. 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。

#### 6-63

程式設計實習

( A ) 44. 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 }

1. int main(){
2. float x;
3. unsigned char found=0;
4. scanf("%f",&a); scanf("%f",&b);
5. scanf("%f",&m); scanf("%f",&n);
6. if( **??** <=0 )
7. found = 1;
8. 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*)。【113 年統測】 A閱讀下文，回答第 45~46 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'};

1. int tmp, i, min;
2. //min=0;
3. for(min=0; min<N; min++)

14 for(i=0; i<N; i++){

1. if(numbers[i]<numbers[min]){
2. //swap(numbers+i, numbers+min);
3. tmp=numbers[min];
4. numbers[min]=numbers[i];
5. numbers[i]=tmp;

20 }

21 }

22 for(i=0; i<N; i++){

23 printf("%d ", numbers[i]);

24 }

25 }

#### 6-64

Chapter 6 函式

( C ) 45. 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 。 【113 年統測】

( A ) 46. 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 的註解拿掉以便啓用函式呼叫 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。 【113 年統測】

打 表示有詳解



1.(D) 2.(C) 3.(B) 4.(C) 5.(B) 6.(B) 7.(B) 8.(D) 9.(C) 10.(B)

11.(C) 12.(A) 13.(B) 14.(A) 15.(C) 16.(B) 17.(B) 18.(C) 19.(A) 20.(C)

21.(C) 22.(C) 23.(C) 24.(B) 25.(D) 26.(A) 27.(C) 28.(B) 29.(A) 30.(C)

31.(C) 32.(A) 33.(C) 34.(D) 35.(C) 36.(C) 37.(C) 38.(D) 39.(B) 40.(D)

41.(A) 42.(A) 43.(C) 44.(A) 45.(C) 46.(A)

I解斬I

1. 採傳址呼叫的方式呼叫 swap()，main()中的原始資料會同時被改變，程式執行過程如下，

|  |  |  |
| --- | --- | --- |
| value=1 | list[5]={0, 2, 4, 6, 8} |  |
| 1 | {0, 2, 6, 4, 8} | swap(&list[2], &list[3]); //list[2]和 list[3]交換 |
| 4 | {0, 2, 6, 1, 8} | swap(&value, &list[3]); //value 和 list[3]交換 |
| 8 | {0, 2, 6, 1, 4} | swap(&value, &list[value]); //value 和 list[4]交換 |

4. puts( )是 C++的函式，用途是將字串寫到標準輸出(stdout)。

1. 控制字元 "%.2f" 表示只顯示小數點後兩位數。
2. 呼叫 mystery(n)，回傳 n!（n 階層），故 mystery(4)=4x3x2x1=24。

9. 呼叫 test(j)會回傳 j!（j 階層），此程式會依序計算 0~14 的階層後輸出，第 9 個數值為 8!=40320。

11. 為能順利求解，我們從較小的參數開始，計算函式的回傳值，

fucn(1)=1 func(2)=2

func(3)=fucn(2)/2+func(1)=2/2+1=3 func(4)=func(3)/2+func(2)=2/2+2=3 func(5)=func(4)/2+func(3)=3/2+3=4 fucn(6)=func(5)/2+func(4)=4/2+3=5

:

func(15)=func(14)/2+func(13)=26/2+21=34

14. 遞迴函式 fibonacci(int n)會回傳費式數列的第 n 個數，參考單元 6-2 範例 4。

15. 參考第 14 題解析。

#### 6-65

程式設計實習

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. 呼叫 bsearch(n, 44, 0, 9)會回傳元素 44 在陣列 n 的索引值。 Step 1：   Step 2：  Step 3：   1. 參考第 16 題。 2. 參考第 16 題。 3. 函數 abc(int x, int y)會回傳 x 和 y 的最大公因數，此程式是輾轉相除法的非遞迴版本，而單元 6-2 的範例 5，是輾轉相除法的遞迴程式版本，同學可以在夜深人靜時，比較兩者的差異，必能深刻體會程式的美。 4. 呼叫 def(7)，因參數 x 為 7，if 條件(x==1)不成立，因此進入 else 區塊執行，過程如下， | | | | | | | |
|  | i | B=0 | A=1 | O | S=0 |  |  |
| 2 | 1 | 1 | 1 | 1 | S=A+B; O=S; B=A;  A=S; |
| 3 | 1 | 2 | 2 | 2 | S=A+B; O=S; B=A;  A=S; |
| 4 | 2 | 3 | 3 | 3 | S=A+B; O=S; B=A;  A=S; |
| 5 | 3 |  | 5 | 5 | S=A+B; O=S; B=A; |

#### 6-66

Chapter 6 函式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 5 |  |  | A=S; |  |
| 6 | 5 | 8 | 8 | 8 | S=A+B; O=S; B=A;  A=S; |
| 7 | 8 | 13 | 13 | 13 | S=A+B; O=S; B=A;  A=S; |
|  |  |  |  |  | return O;//函式 def()結束，回傳值為 13 |
| 1. round(2.2) = 2, round(2.8) = 3, round(-2.4) = -2, round(-1.8) = -2。  1. 函式 SubProgram(int x, int \*y, int \*z){ … }的參數 x 可視為原始變數的複製品，屬於函式內部的區域變數，而 y 和 z 則是指向原始變數的指標變數，函式執行後，不會改變 main()內的變數 x 的值，但會改變變數 y和 z 的值。 2. 函式 test(int a)會回傳 a!（a 階層），程式執行後，z = test(x)/test(y)\*test(x-y) = test(5)/test(2)\*test(3) = 5!/2!\*3! = 360。 3. 呼叫函式 TestYou(5)的過程如下，可知 TestYou(5)=0+6+12+20+30=68。 4. 遞迴函式 TestYou(int a, intb)回傳變數 a 和 b 的最大公因數，請參考單元 6-2 範例 5。 5. 參考精選試題第 53 題，TestYou(2, 7) = 27 = 128。 6. round(2.2) = 2, round(2.8) = 3, round(-2.4) = -2, round(-1.8) = -2。 7. 函式 printDigit(int n)將輸入的參數 n，從最低位元開始輸出，輸出結果剛好左右顛倒。 8. 呼叫 swap( )採傳值呼叫，在 swap( )內將 a 和 b 交換，不會改變 main( )內 a 和 b 的值。 9. 呼叫函式 f(4)的過程如下，可知 f(4)=1+1+2+3=7。 10. 參考第 14 題解析。 11. 參考精選試題第 53 題，FunctionA(3, 6) = 36 = 729 。 12. 函式 f(int a[], int n)會回傳陣列 a[]中的最大值元素的索引，請參考單元 6-3 範例 7。 | | | | | | | |

#### 6-67

程式設計實習

1. 程式執行過程與說明如下，

(l) #include <stdio.h>

丶、--. int s = 1;

｀｀

|  |  |  |
| --- | --- | --- |
| void add(int); |  | |
| int main(){ |  | (2) |
| int s=l O;  printf("¾d", s);  -·'· add(s); | ]; | (5) |

-- ·

(3 ---- p

)" rintf("¾d", s); •

、\return O;

、`�oid add(int a){ \_ (4)

s = s+a; ]./

printf("¾d", s);

1. 宣告全域變數 s=1。
2. 宣告區域變數 s=10，若存在相同名稱的區域變數與全域變數，則優先存取區域變數，故 輸出 10。
3. 呼叫函式 add(s)，傳入的引數為區域變數 s 的值，故參數 a 會被指定為 10。
4. 指定全域變數 s 的值為 11(s=s+a=1+10=11)，輸出 11。
5. 輸出 10(區域變數 s)。
6. 程式執行過程與變數在記憶體內的變化如下，

#include <stdio.h>

* 1. ~---• int s = 1;

void add(int);

int main(){

* int s=lO;
* add(s);
* add(s); return O;

(5) }

� void add(int a){

(6)

｛［門］］

printf("¾d", s);

1. 宣告全域變數 s=1

口

全域變數

1. 宣告區域變數 s=10，若存在相同名稱的區域變數與全域變數，則優先存取區域變數。

二

,, II

全域變數 ＇11＇

＇＇

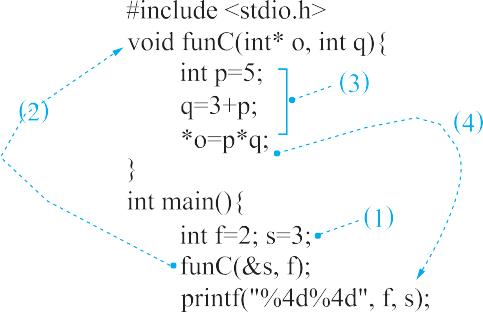
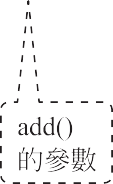
＇I ＇I

main()的區域變數

#### 6-68

Chapter 6 函式

1. 呼叫函式 add(s)，傳入的引數為區域變數 s 的值，故參數 a 會被指定為 10。



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

1. 宣告區域變數 s=1，指定區域變數 s 的值為 11(s=s+a=1+10=11)，輸出 11，當函式結束時，add()的區域變數 s 和a 即消失。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

1. 同步驟(3)。
2. 同步驟(4)。
3. 程式執行過程與變數在記憶體內的變化如下，
   1. 宣告整數變數 f 和 s。

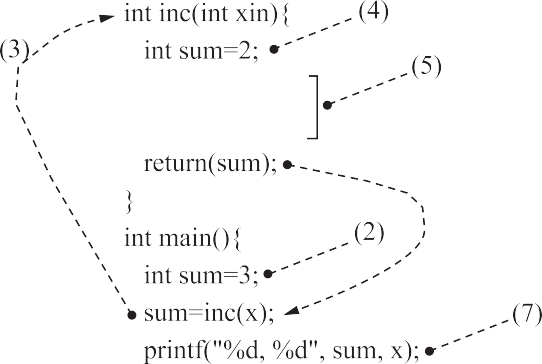
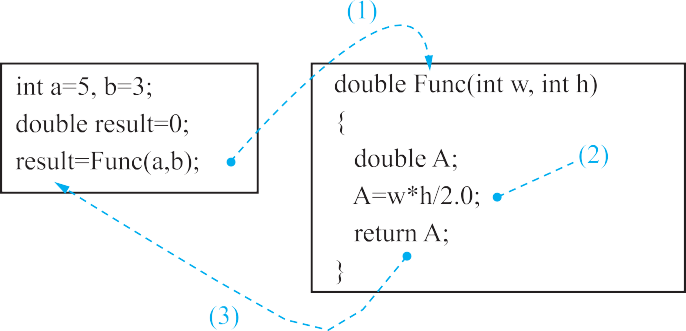
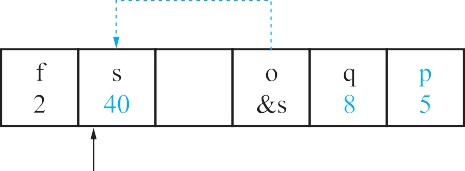
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

* 1. 採傳址方式呼叫函式 funC(&s, f)，引數為變數 s 的位址和變數 f 的值，分別由參數 o(指標變數)和 q(整數變數)接收，呼叫後，o 會指向 s。

#### 6-69

程式設計實習

* 1. 宣告區域變數 p=5，q=3+p=8，執行\*o=p\*q 後，變數s 的值將被指定為 40(p\*q=5\*8=40)。



* 1. 函式執行結束後，區域變數 o、p、q 將消失，並返回原來的程式碼位置繼續執行，輸出變數 f 和 s。

1. 程式執行過程如下，
   1. 呼叫 Func(a, b)，傳入引數 5 和 3，分別指定給 Func()的參數 w 和 h。
   2. w 和 h 為整數型態，故 w\*h 為整數相乘，結果為整數值 15，之後進行「/j運算時，會先自動轉型為浮點數 15.0 後再除 2.0，運算後 A 的值為 7.5。
   3. 傳回 A 的值，指定給 result。
2. (A)回傳值為 double 型態，但沒有參數，不符合題目要求

(B)void 表示沒有回傳值，不符合題目要求

(C)回傳值為 int 型態，不符合題目要求

1. (A) 程式執行時要讀取使用者輸入的參數，需在 main()函式中使用 argc 和 argv，修改後反而不符題意。

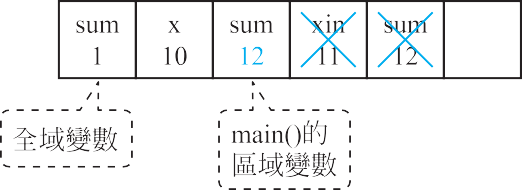
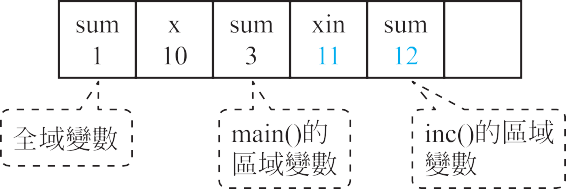
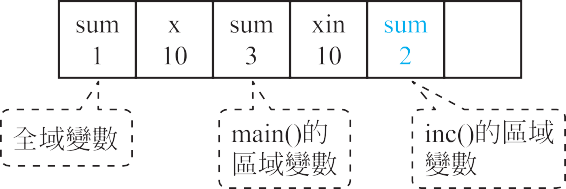
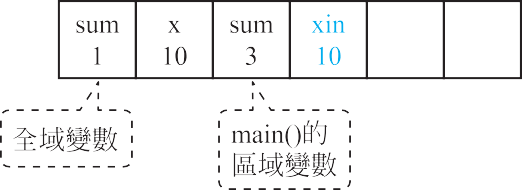
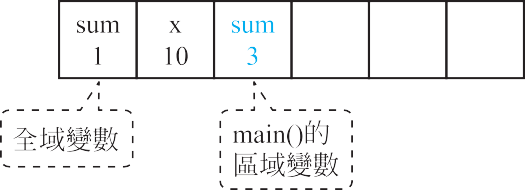
(B) 編譯錯誤的原因是沒有宣告 sub()的函式原型，將第 3 行的註解拿掉即可正常編譯。

1. 程式執行過程與變數在記憶體內的變化如下，

#### 6-70

Chapter 6 函式

* 1. 宣告全域變數 sum 和 x



* 1. 宣告 main()的區域變數 sum
  2. 呼叫 inc(x)，引數為全域變數 x 的值，函式被呼叫後，參數 xin 等於 10
  3. 宣告 inc()的區域變數 sum
  4. 運算後，sum=12、xin=11
  5. 函式 inc()結束，區域變數 xin 和 sum 消失，返回原程式位置，回傳 12，main()的區域變數 sum 被指定為 12
  6. 輸出 main()的區域變數 sum 與全域變數 x 的值

1. 函式呼叫時，使用傳址呼叫的方式傳遞資料，即可同步改變來源變數的值。
2. 全域變數才能在程式的任何地方被存取，修正後的程式碼如下，

#include <stdio.h>

float a=1, b=0, c=-1; //宣告 a, b, c 為全域變數 float f(float x){

return(a\*x\*x+b\*x+c);

}

int main(){

#### 6-71

程式設計實習

float x; //宣告 x 為區域變數 for(x= –10; x<=10; x=x+0.1)



printf("f(%.1f)=%.1f\n", x, f(x));

}

1. (1) 當 *f* (*m*)與 *f* (*n*)為一正一負時，兩點間的連線必經過 x 軸，則存在 *f* (*x*)=*ax*+*b*=0，且 *m*  *x*  *n* 。
   1. 如下圖所示，假設 *a*=2，*b*=3，*m*= –4，*n*=2，則 *f* (*m*)= –5，*f* (*n*)=7，可得到 *f* (–1.5)=0。
   2. 程式第 11 行，若判斷式 *f* (*m*)\**f* (*n*)<=0 成立，表示 *f* (*m*)與 *f* (*n*)為一正一負，即存在 *f* (*x*)=0。
2. 參考單元 5-1 範例 4，其中字元'a'的 ASCII 碼為 97。
3. 第 16 行呼叫函式 swap()時，傳入的引數為指標(記憶體位址)，則函式 swap()的參數必須是指標變數，修正後的程式碼如下，

#include <stdio.h> #define N 11

void swap(int \*a, int \*b){ int tmp;

tmp=\*a;

\*a=\*b;

\*b=tmp;

}

void main(void){

int numbers[N]={1,3,5,7,9,2,4,6,8,0,'a'};

int tmp, i, min;

//min=0;

for(min=0; min<N; min++) for(i=0; i<N; i++){

if(numbers[i]<numbers[min]){ swap(numbers+i, numbers+min);

//tmp=numbers[min];

#### 6-72

Chapter 6 函式

//numbers[min]=numbers[i];

//numbers[i]=tmp;

}

}

for(i=0; i<N; i++){

printf("%d ", numbers[i]);

}

}