

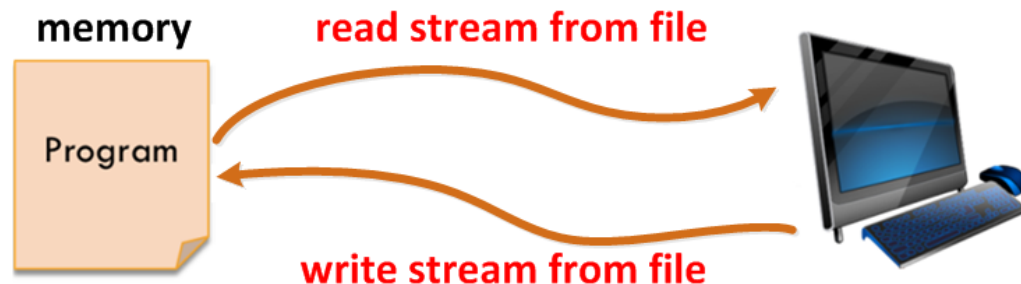
CHAPTER 7

File Processing (ch11)

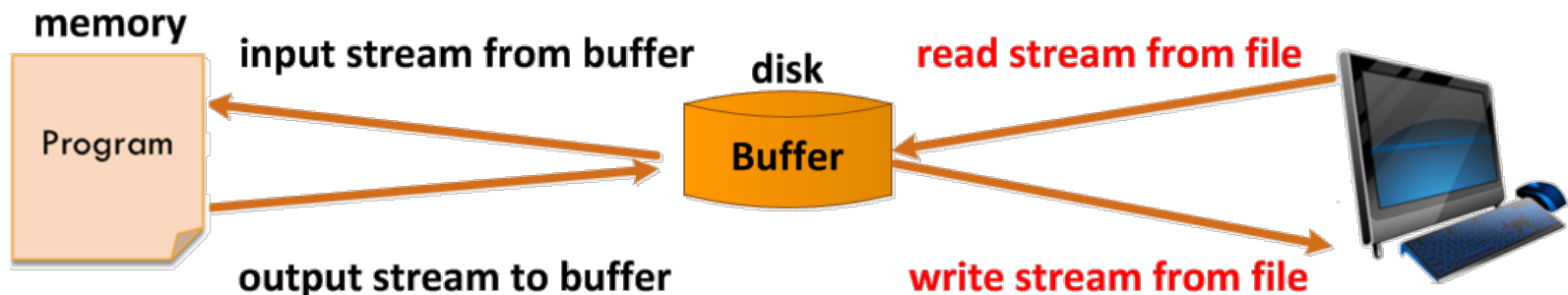


- **File Processing**主要是將資料檔案放在存放在磁碟中，不同於先前的資料放存放在記憶體中。
 - 將資料檔案放在存放在磁碟中，如果電腦重新開機，仍舊可以保留這些資料，資料不會因此而消失。

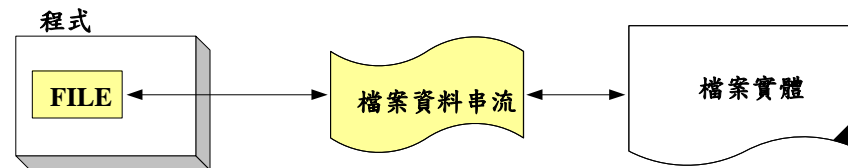
1. 資料放在記憶體中



2. 資料檔案放在磁碟中



- File Processing的資料檔案(Files)分為文字檔(Text file)與二進位檔(Binary file)兩種資料串流(stream)，函式庫stdio.h負責檔案資料串流(stream)處理



1. 文字檔(Text file)

- 方便閱讀，但較無保密性。
- 可以透過純文字編輯器開啟並成功閱讀，例如利用 Windows 的記事本就是文字檔，內容可以是英文字母、數字、中文等。通常文字檔(Text file)的副檔名會用 .txt，但這並非必要。

2. 二進位檔(Binary file)

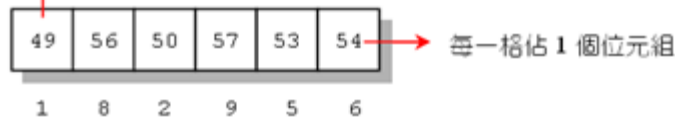
- I/O處理速度較快並具有保密性，但檔案內容需透過程式轉譯才能閱讀。
- 二進位檔的資料是由一連串的位元組(Byte)所組成，通常使用在某些特殊用途，例如圖片檔、音樂檔、Word 檔、資料庫檔、執行檔等。

檔案格式(File Types)，以數值182956為例

1. 文字檔(Text file)

將數值 182956
以文字檔儲存

以 ASCII 碼儲存



2. 二進位檔(Binary file)

將數值 182956 以二
進位的格式儲存

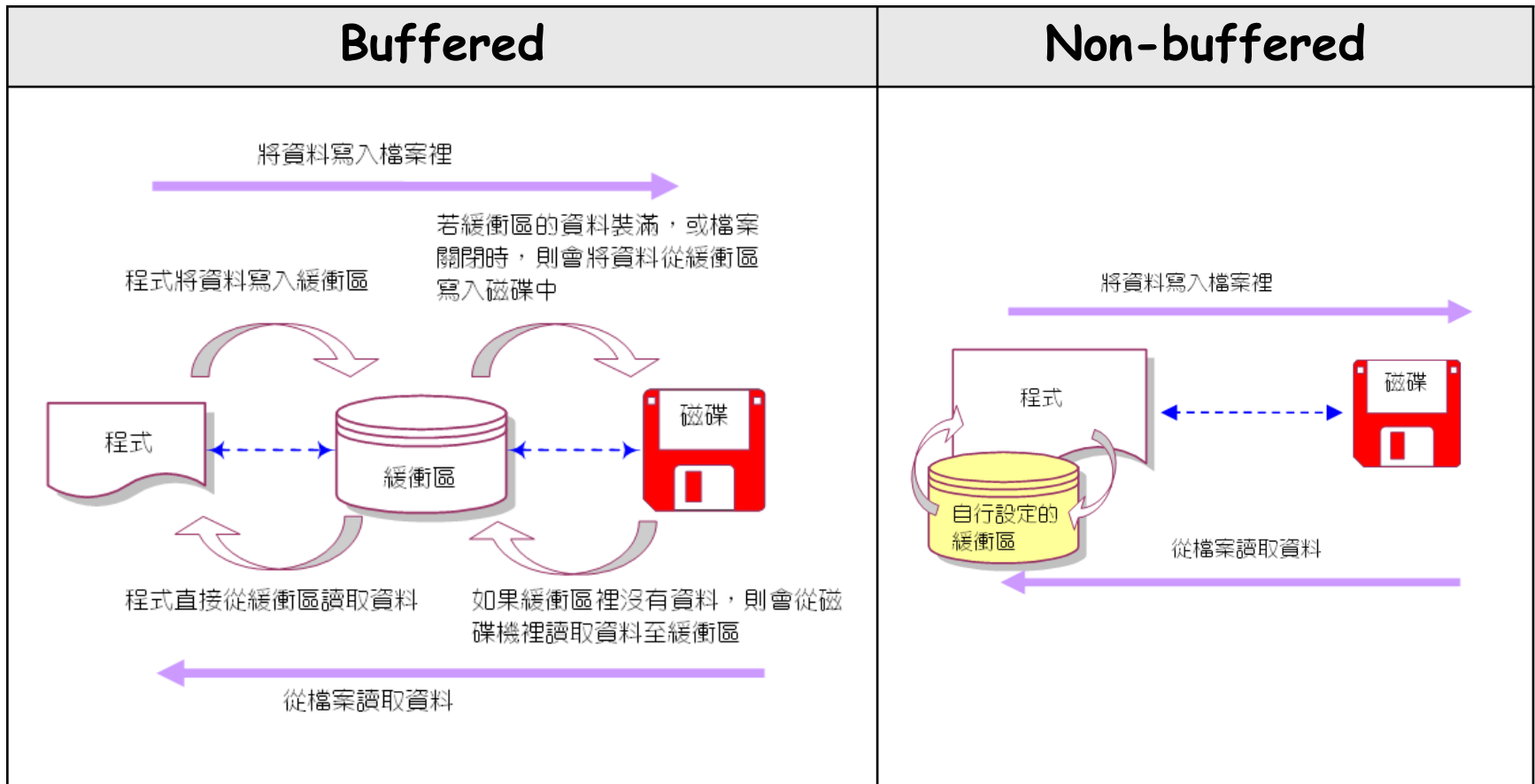
以二進位碼儲存



182956

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	Space	64	40	100	64	@	96	60	140	96	`	
1	1	001	SOH (start of heading)	33	21	041	!	65	41	101	65	A	97	61	141	97	a	
2	2	002	STX (start of text)	34	22	042	"	66	42	102	66	B	98	62	142	98	b	
3	3	003	ETX (end of text)	35	23	043	#	67	43	103	67	C	99	63	143	99	c	
4	4	004	EOT (end of transmission)	36	24	044	\$	68	44	104	68	D	100	64	144	100	d	
5	5	005	ENQ (enquiry)	37	25	045	%	69	45	105	69	E	101	65	145	101	e	
6	6	006	ACK (acknowledge)	38	26	046	&	70	46	106	70	F	102	66	146	102	f	
7	7	007	BEL (bell)	39	27	047	'	71	47	107	71	G	103	67	147	103	g	
8	8	010	BS (backspace)	40	28	050	(72	48	110	72	H	104	68	150	104	h	
9	9	011	TAB (horizontal tab)	41	29	051)	73	49	111	73	I	105	69	151	105	i	
10	A	012	LF (NL line feed, new line)	42	2A	052	*	74	4A	112	74	J	106	6A	152	106	j	
11	B	013	VT (vertical tab)	43	2B	053	+	75	4B	113	75	K	107	6B	153	107	k	
12	C	014	FF (NP form feed, new page)	44	2C	054	,	76	4C	114	76	L	108	6C	154	108	l	
13	D	015	CR (carriage return)	45	2D	055	-	77	4D	115	77	M	109	6D	155	109	m	
14	E	016	SO (shift out)	46	2E	056	.	78	4E	116	78	N	110	6E	156	110	n	
15	F	017	SI (shift in)	47	2F	057	/	79	4F	117	79	O	111	6F	157	111	o	
16	10	020	DLE (data link escape)	48	30	060	0	80	50	120	80	P	112	70	160	112	p	
17	11	021	DC1 (device control 1)	49	31	061	1	81	51	121	81	Q	113	71	161	113	q	
18	12	022	DC2 (device control 2)	50	32	062	2	82	52	122	82	R	114	72	162	114	r	
19	13	023	DC3 (device control 3)	51	33	063	3	83	53	123	83	S	115	73	163	115	s	
20	14	024	DC4 (device control 4)	52	34	064	4	84	54	124	84	T	116	74	164	116	t	
21	15	025	NAK (negative acknowledge)	53	35	065	5	85	55	125	85	U	117	75	165	117	u	
22	16	026	SYN (synchronous idle)	54	36	066	6	86	56	126	86	V	118	76	166	118	v	
23	17	027	ETB (end of trans. block)	55	37	067	7	87	57	127	87	W	119	77	167	119	w	
24	18	030	CAN (cancel)	56	38	070	8	88	58	130	88	X	120	78	170	120	x	
25	19	031	EM (end of medium)	57	39	071	9	89	59	131	89	Y	121	79	171	121	y	
26	1A	032	SUB (substitute)	58	3A	072	:	90	5A	132	90	Z	122	7A	172	122	z	
27	1B	033	ESC (escape)	59	3B	073	;	91	5B	133	91	[123	7B	173	123	{	
28	1C	034	FS (file separator)	60	3C	074	<	92	5C	134	92	\	124	7C	174	124		
29	1D	035	GS (group separator)	61	3D	075	=	93	5D	135	93]	125	7D	175	125	}	
30	1E	036	RS (record separator)	62	3E	076	>	94	5E	136	94	^	126	7E	176	126	~	
31	1F	037	US (unit separator)	63	3F	077	?	95	5F	137	95	_	127	7F	177	127	DEL	

- 檔案處理(File Handling)分為有緩衝區的檔案處理(Buffered)和無緩衝區的檔案處理(Non-buffered)



- 檔案讀寫中，在 `stdio.h` 裡有一個以 `typedef` 定義的 `FILE` 結構體型別，用來建立代表資料流的變數，讓程式據以存取檔案內容。
- 處理檔案需要經過以下 3 個步驟

(Step1) 開啟檔案(Open Files)

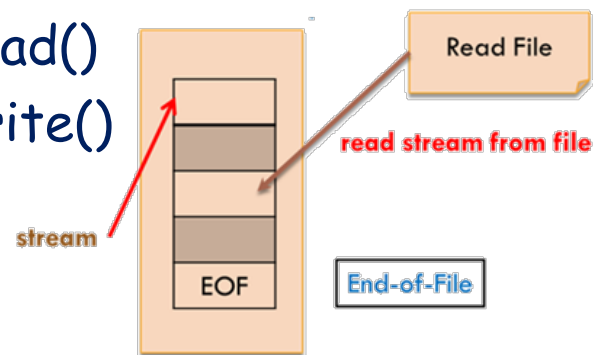
- 指令: `FILE *stream;`
`stream = fopen("file name", "open mode");`
- `fopen()` returns a `FILE` pointer to the beginning of the file data. If open fails, `NULL` returned

(Step2) 讀寫檔案(Read/Write Files)

- R指令: `fgetc()`、`fgets()`、`fprintf()`、`fread()`
- W指令: `fputc()`、`fputs()`、`fscanf()`、`fwrite()`

(Step3) 關閉檔案(Close files)

- 指令: `fclose(stream);`



- 指令: `FILE *stream;`
`stream = fopen("file name", "open mode");`
- 開啟模式(open mode)
 - 讀取資料: `r`
 - 寫入資料(會覆蓋): `w`
 - 附加在檔案後(不會覆蓋): `a`
 - 可讀可寫: `+`

mode	開檔模式說明
"r"	開啟檔案，並準備讀取檔案內容。檔案若不存在，則發生錯誤。
"w"	開啟檔案，先清除檔案內容，再重新準備寫入資料。若檔案尚未建立（檔案不存在），則建立新檔。
"a"	開啟檔案，並將寫入新資料，新的資料會加在檔案原始資料的後面。
"r+"	開啟檔案，檔案可讀／寫。若檔案不存在，則發生錯誤。
"w+"	開啟檔案，檔案可讀／寫。若檔案已存在，則先清除檔案內容。
"a+"	開啟檔案，檔案可讀／寫，寫入的資料會加在檔案原始資料的後面。

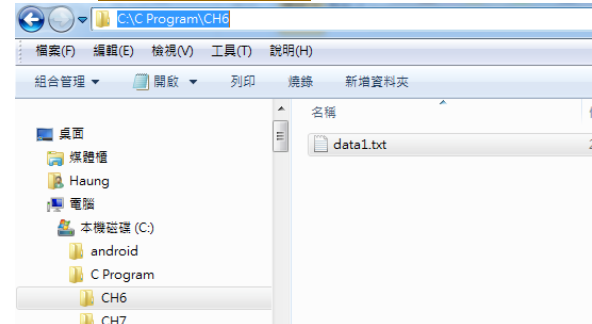
- 指令: `FILE *stream;`
`stream = fopen("file name", "open mode");`
- 開啟模式(open mode)
 - 二進位: b
 - 可讀可寫: +

mode	開檔模式說明
"rb"	開啟二進位檔案，並準備讀取檔案。檔案若不存在，則發生錯誤。
"wb"	開啟二進位檔案，先清除檔案內容，再重新準備寫入資料。若檔案尚未建立（檔案不存在），則建立新檔。
"ab"	開啟二進位檔案，並將寫入新資料，新的資料會加在檔案原始資料的後面。
"r+b"	開啟二進位檔案，檔案可讀／寫。若檔案不存在，則發生錯誤。
"w+b"	開啟二進位檔案，檔案可讀／寫。若檔案已存在，則先清除檔案內容。
"a+b"	開啟二進位檔案，檔案可讀／寫，寫入的資料會加在檔案原始資料的後面。

- 指令: `FILE *stream;`
`stream = fopen("file name", "open mode");`
- 檔案名稱(**file name**)可以包含檔案路徑(path), 分為以下三種方式
 - 相同資料夾
 - EX: 讀取相同資料夾的 data1.txt 檔案
指令 `stream = fopen("data1.txt", "r");`
 - 絕對路徑(absolute path)
 - EX: 開啟 C:\C Program\CH7 的 data1.txt, 使用寫入
指令 `stream = fopen("C:\C Program\CH7\ data1.txt", "w");`
 - 相對路徑(relative path)
 - EX: 目前在 C:\C Program\CH7, 開啟 C:\C Program\CH6 的 data1.txt, 使用寫入
指令 `stream = fopen("../CH6\data1.txt", "w");`
» ..: 表示上一層資料的意思, .: 表示相同資料夾的意思



目前在 C:\C Program\CH7



開啟 C:\C Program\CH6 的 data1.txt

- Read 指令：fgetc()、fgets()、fprintf()、fread()
- Write指令：fputc()、fputs()、fscanf()、fwrite()

函式	說明
fgetc()	從已開啓的檔案中讀取一個字元
fgets()	從已開啓的檔案中讀取一個字串
fputc()	將一個字元寫入已開啓的檔案中
fputs()	將一個字串寫入已開啓的檔案中

函式	說明
fprintf()	與printf()類似,只是輸出的對象是一個已開啓的檔案,而非標準輸出
fscanf()	與scanf()類似,只是輸入的資料來源是一個已開啓的檔案,而非從標準輸入輸入資料

- 指令: `fclose(stream);` //FILE *stream; stream為自己命名檔案指標變數
 - 回傳值: -1 或 0
 - 當回傳值為-1時，代表關閉檔案時發生錯誤
 - 當回傳值等於0，則表示成功關閉檔案
 - stream是一個已開啟的檔案指標變數，使用C語言開檔並處理資料完畢之後，記得要將檔案關閉，否則可能產生無法預期的結果。

```
#include <stdio.h>          /* (Step1) 開啟檔案的函式庫 */
FILE *fptr;                 /* (Step1) 宣告指向檔案的指標fptr */
fptr=fopen("abc.txt","r");  /* (Step1) 開啟檔案abc.txt以供讀取 */
if(fptr!=NULL)              /* 判別檔案是否開啟成功 */
{
    /* (Step2) 開啟檔案檔案開啟成功時，執行讀寫檔案的動作 */
}
else
{
    /* 檔案開啟失敗時，所要執行的程式碼 */
}
fclose(fptr);               /* (Step3) 關閉檔案 */
```

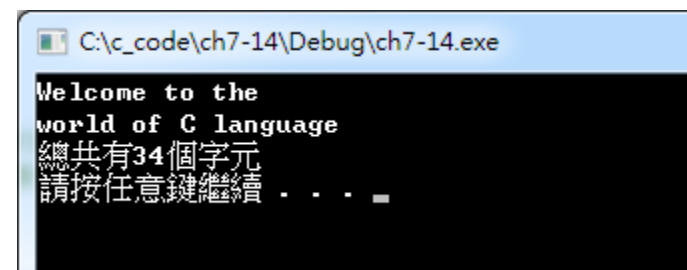
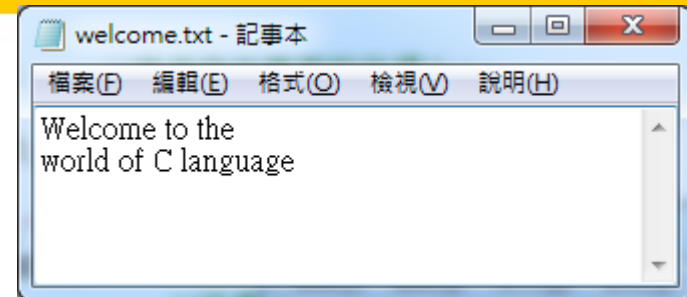
函數功能	格式及說明
開啟檔案	<pre>FILE *fopen(const char *filename, const char *mode);</pre> <p>開啟指定的檔案，並指定存取模式。fopen() 的第一個引數為檔案名稱字串，第二個引數為存取模式的代表。fopen() 的傳回值為檔案指標，開檔失敗傳回 NULL</p>
關閉檔案	<pre>int fclose(FILE *fptr);</pre> <p>關閉由 fptr 所指向的檔案，關檔成功傳回 0</p>
讀取字元	<pre>int getc(FILE *fptr);</pre> <p>由 fptr 所指向的檔案讀取一個字元，傳回值為被讀取的字元</p>
寫入字元	<pre>int putc(int ch, FILE *fptr);</pre> <p>將字元 ch 寫入由 fptr 所指向的檔案</p>

函數功能	格式及說明
讀取字串	<pre>char *fgets(char *str,int maxchar,FILE *fptr);</pre> <p>從 fptr 所指向的檔案裡讀取最多 maxchar 個字元，然後將它寫入字元陣列 str 中。若讀取失敗，或已讀到檔尾，則傳回 NULL</p>
寫入字串	<pre>int fputs(const char *str,FILE *fptr);</pre> <p>將字串 str 寫入 fptr 所指向的檔案</p>
檢查檔案是否結束	<pre>int feof(FILE *fptr);</pre> <p>檢查 fptr 所指向的檔案是否已讀取到檔案結束的位置。若尚未到達檔尾，則傳回 0；若已到檔尾，則傳回非 0 的值</p>
區塊輸入	<pre>size_t fread(void *p,size_t s,size_t cnt,FILE *fptr);</pre> <p>由檔案讀取 cnt 個資料項目，存放到指標 p 所指向的位址中，每一個資料項目的大小為 s 個位元組，傳回值為讀取資料的個數</p>
區塊輸出	<pre>size_t fwrite(const void *p,size_t s,size_t cnt,FILE *fptr);</pre> <p>將 cnt 個大小為 s 個位元組的資料，寫入指標 p 所指向的位址中，傳回值為成功寫入資料的個數</p>

```

1  /*顯示檔案內容，並計算字元數*/
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main(void)
6  {
7      FILE *fptr;      //宣告指向檔案的指標fptr
8      char ch;          //宣告字元變數ch，用來接收讀取的字元
9      int count=0;      //宣告整數count，用來計算檔案的字數
10
11     fptr=fopen("welcome.txt","r");      //開啟檔案
12     if (fptr!=NULL)      //如果fopen()的傳回值不為NULL，
13     {                    //代表檔案開啟成功
14         while ((ch=getc(fptr)) != EOF) //判斷是否到達檔尾
15         {
16             printf("%c",ch);          //一次印出一個字元
17             count++;
18         }
19         fclose(fptr);                //關閉所開啟的檔案
20         printf("\n總共有%d個字元\n",count);
21     }
22     else
23         printf("檔案開啟失敗!!\n");
24     system("pause");
25     return 0;
26 }

```

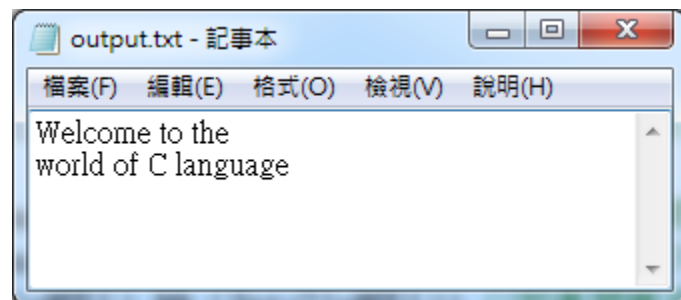
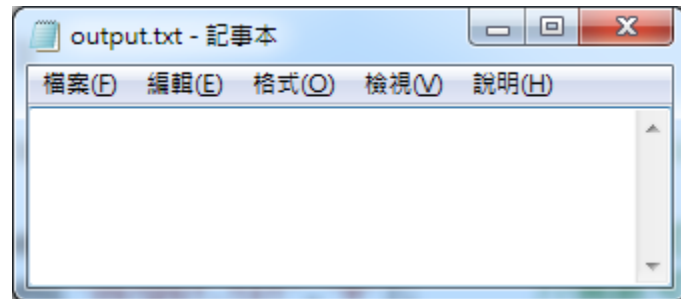
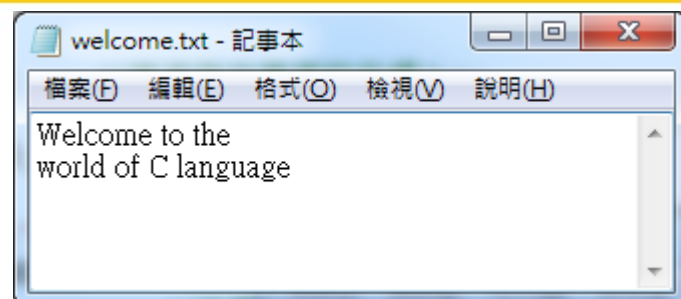


空白與換行字元也列入字元數的計算，因此總字元數為34：

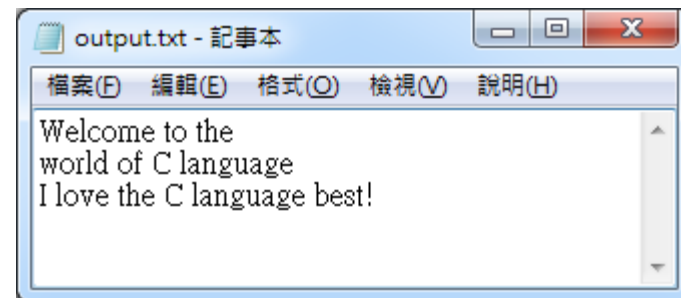
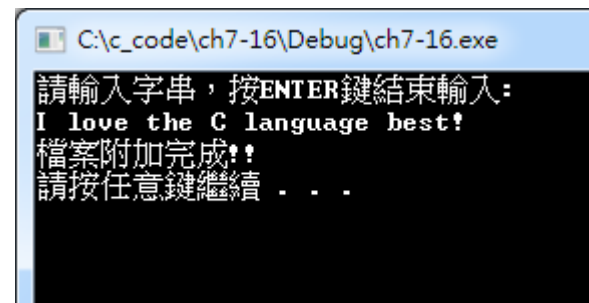
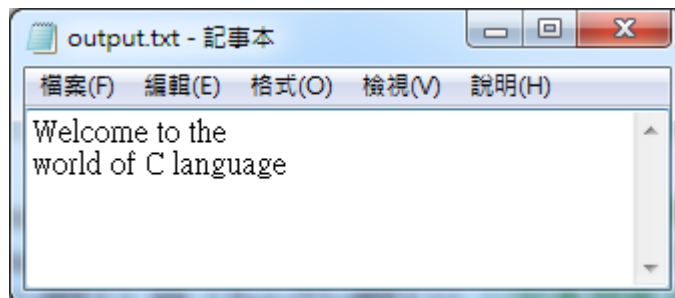
W	e	l	c	o	m	e		t	o		t	h	e	\n
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

w	o	r	l	d		o	f		C		l	a	n	g	u	a	g	e
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34

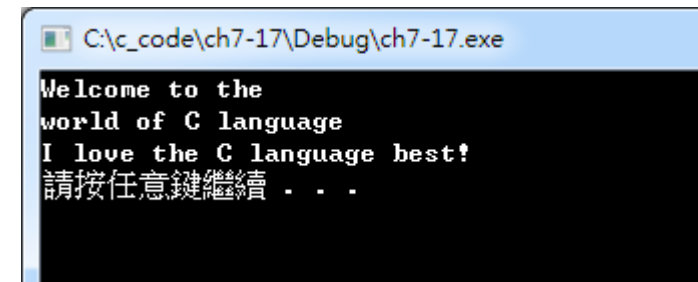
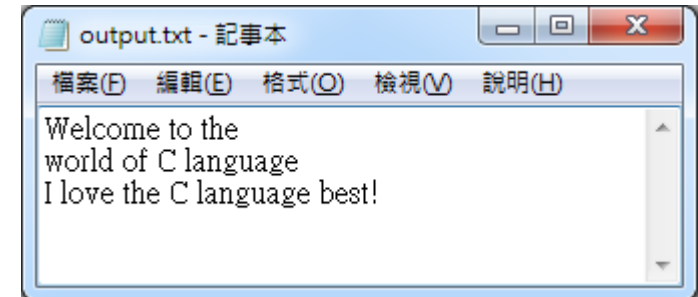
```
1  /*拷貝檔案內容到其他的檔案*/
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main(void)
6  {
7      FILE *fptr1,*fptr2; //宣告指向檔案的指標 fptr1與 fptr2
8      char ch;
9      fptr1=fopen("welcome.txt","r");    //開啟可讀取的檔案
10     fptr2=fopen("output.txt","w");      //開啟可寫入的檔案
11     if ((fptr1!=NULL) && (fptr2!=NULL)) //如果開檔成功
12     {
13         while ((ch=getc(fptr1)) != EOF) //判斷是否到達檔尾
14             putc(ch,fptr2); //將字元ch寫到 fptr2所指向的檔案
15         fclose(fptr1);      //關閉 fptr1所指向的檔案
16         fclose(fptr2);      //關閉 fptr2所指向的檔案
17         printf("檔案拷貝完成!!\n");
18     }
19     else
20         printf("檔案開啟失敗!!\n");
21     system("pause");
22     return 0;
23 }
```



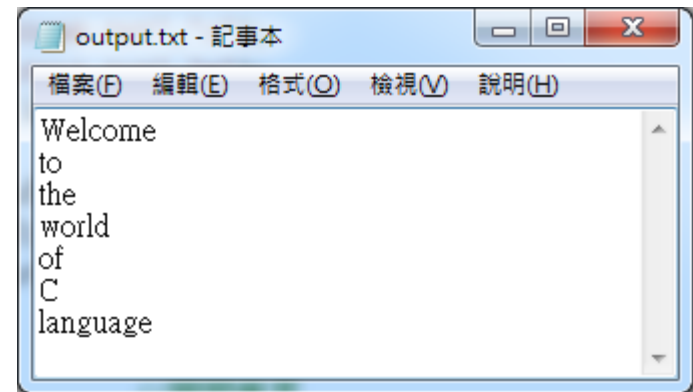
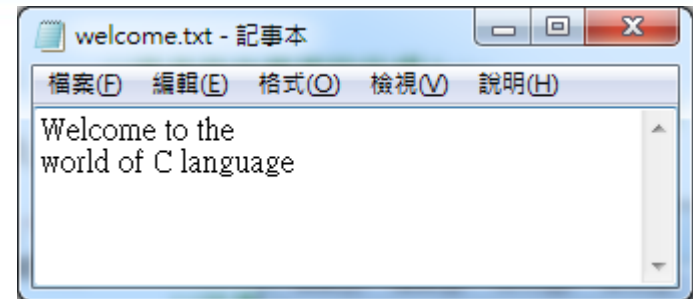
```
1  /*由鍵盤輸入字串，並附加到檔案output.txt*/
2  #include <stdio.h>
3  #include <conio.h>
4  #include <stdlib.h>
5  #define ENTER 13
6  #define MAX 80
7
8  int main(void)
9  {
10     FILE *fptr;
11     char str[MAX],ch; //宣告字元陣列str，用來儲存由鍵盤輸入的字串
12     int i=0;
13     fptr=fopen("output.txt","a");
14     printf("請輸入字串，按ENTER鍵結束輸入:\n");
15     while ((ch=getche()) != ENTER && i<MAX) //按下的鍵不是ENTER且i<MAX
16         str[i++]=ch; //一次增加一個字元到字元陣列str中
17     putc('\n',fptr); //寫入換行字元
18     fwrite(str,sizeof(char),i,fptr);
19     fclose(fptr);
20     printf("\n檔案附加完成!!\n");
21     system("pause");
22     return 0;
23 }
```




```
1  /*使用fread()函數讀取檔案內容*/
2  #include <stdio.h>
3  #include <stdlib.h>
4  #define MAX 80
5
6  int main(void)
7  {
8      FILE *fptr;
9      char str[MAX];
10     int bytes;      //存放fread()成功讀取的字元數
11     fptr=fopen("output.txt","r");
12     while (!feof(fptr)) //如果還沒讀到檔尾
13     {
14         bytes=fread(str,sizeof(char),MAX-1,fptr);
15         str[bytes]='\0';
16         printf("%s\n",str); //印出檔案內容
17     }
18     fclose(fptr);      //關閉檔案
19     system("pause");
20     return 0;
21 }
```



```
4 int main(void)
5 {
6     FILE *in,*out;
7     int cnt;
8     char str[100];
9     in=fopen("welcome.txt","r");
10    out=fopen("output.txt","w");
11    while (!feof(in))
12    {
13        cnt=fscanf(in,"%s",str); //讀取檔案
14        if (cnt>0)
15            fprintf(out,"%s\n",str); //寫入檔案
16    }
17    fclose(in); //關閉檔案
18    fclose(out); //關閉檔案
19    system("pause");
20    return 0;
21 }
```



- fscanf() to read from file (讀取檔案), fprintf() to write to file (寫入檔案)
 - fscanf(資料流stream的指標變數, 讀取的資料格式, 讀取資料存放的變數位置)
 - fprintf(資料流stream的指標變數, 寫入的資料格式, 要寫入資料存放的變數位置)
 - skipping spaces ('\n', '\t', ' ')
 - fscanf () 函式讀取資料成功時, 會傳回成功讀取的資料數, 如果讀取失敗則會傳回 EOF
 - EX1: fscanf(stream, "%s %s", str1, str2); //讀取成功傳回2
 - EX2: fscanf(stream, "%s %s %d", str1, str2, number); //讀取成功傳回3

- `fscanf ()` 的優點在於可以自由的設定讀取格式,例如:檔案資料中有姓名(char [])、電話(phone [])與年齡(age), 必須同時使用不同的格式,我們可以利用 `fscanf ()` 來完成使用不同的格式

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int age;
7     char name[8], phone[12];
8     FILE *stream;
9     stream=fopen("addbook.txt", "r");
10    if (stream==NULL)
11    {
12        printf("檔案開啟失敗!!\n");
13    }
14    else
15    {
16        printf("  姓名    電話    年齡\n");
17        printf("-----\n");
18        while ((fscanf(stream, "%s %s %d", name, phone, &age)) != EOF)
19            printf("%6s %-11s %4d\n", name, phone, age);
20        fclose(stream);
21    }
22    system("pause");
23    return 0;
24 }

```

```

addbook.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
張三 02-22331144 18
李四 07-1112222 32
王瓜 08-2223333 25
趙曉華 02-33444556 18
陳大明 08-1234567 30
蔡光頭 06-4564794 21
歐森 02-88877777 22
丹尼斯 04-3355544 32

```

```

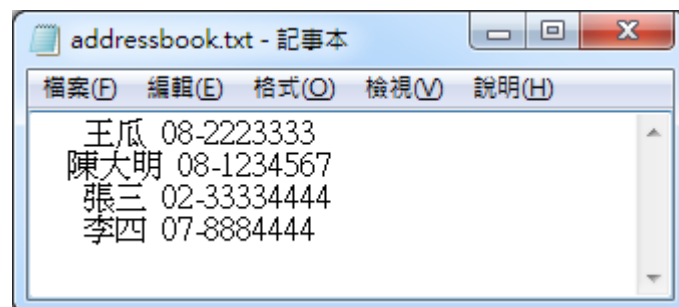
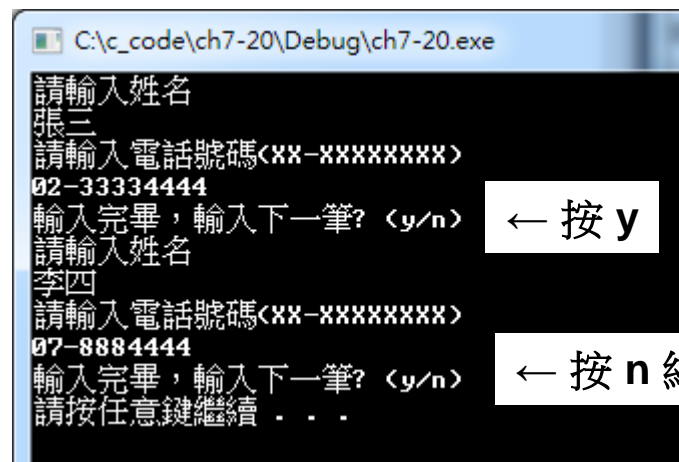
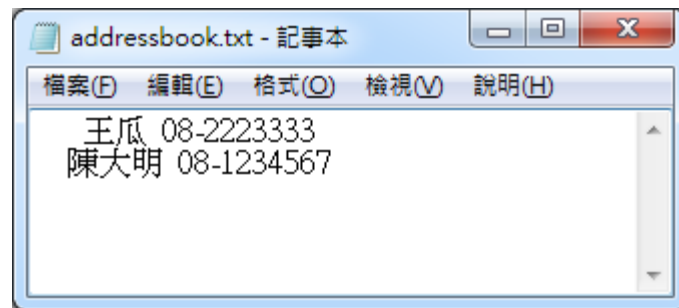
C:\c_code\ch7-19\Debug\ch7-19.exe
姓名    電話    年齡
-----
  張三    02-22331144    18
  李四    07-1112222    32
  王瓜    08-2223333    25
  趙曉華  02-33444556    18
  陳大明  08-1234567    30
  蔡光頭  06-4564794    21
  歐森    02-88877777    22
  丹尼斯  04-3355544    32
請按任意鍵繼續 . . .

```

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4
5 int main(void)
6 {
7     char name[10],phone[12],choice;
8     FILE *stream;
9     stream=fopen("addressbook.txt","a");
10    if (stream==NULL)
11    {
12        printf("檔案開啟失敗!!\n");
13    }
14    else
15    {
16        do
17        {
18            printf("請輸入姓名\n");
19            scanf("%s",name);
20            printf("請輸入電話號碼(XX-XXXXXXX)\n");
21            scanf("%s",phone);
22            fprintf(stream,"%10s %-12s\n",name,phone);
23            printf("輸入完畢，輸入下一筆? (y/n)\n");
24            choice=getch();
25        }while (choice=='y');
26        fclose(stream);
27    }
28    system("pause");
29    return 0;
30 }

```



```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int i;
7     char lf;
8     FILE *fp;
9     typedef struct _student
10    {
11        char num[7];    //學號
12        char clas[9];   //班級
13        char name[12];  //姓名
14        int score[3];   //各科成績
15    } student;
16    student stu[10];
17    fp=fopen("stu.txt","r");
18    printf("\n          *****\n");
19    printf("          * 學生成績 * \n");
20    printf("          *****\n\n");

```

```

21    printf("===== \n");
22    printf(" 學號      班級      姓名      國文  英文  數學 \n");
23    printf("===== \n");
24    for (i=0; i<=9; i++)
25    {
26        fgets(stu[i].num, 7, fp);
27        fgets(stu[i].clas, 9, fp);
28        fgets(stu[i].name, 12, fp);
29        fscanf(fp, "%3d%3d%3d%c", &stu[i].score[0],
30                &stu[i].score[1], &stu[i].score[2], &lf);
31        printf("%-8s%-8s%-10s  %4d  %4d  %4d\n",
32                stu[i].num, stu[i].clas, stu[i].name,
33                stu[i].score[0], stu[i].score[1], stu[i].score[2]);
34    }
35    fclose(fp);
36    printf("\n\n");
37    system("pause");
38    return 0;
39 }

```

本機磁碟 (C:) > c_code > ch7-9 > ch7-9 > Debug

欲 燒錄 新增資料夾

名稱	修改日期	類型
stu.txt	2012/12/31 下午 ...	文字文

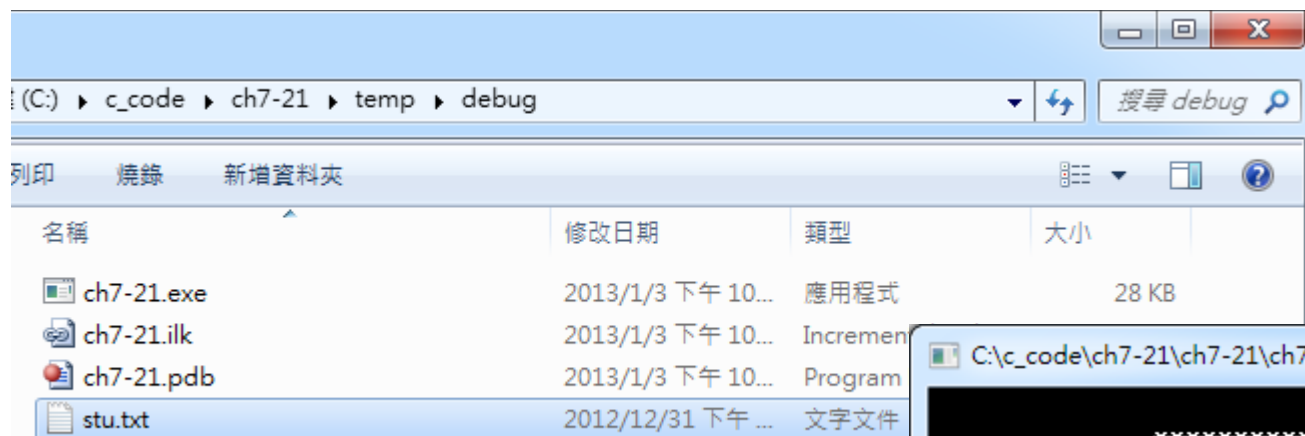
stu.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```

970101四子一乙 王小明 61 71 81
970102四子一乙 李中雄 100 82 72
970103四子一乙 張大成 73 100 83
970104四子一乙 Tom Johns 64 74 54
970105四子一乙 陳小冬 65 65 65
970106四子一乙 Mary Lin 76 76 56
970107四子一乙 錢中春 57 87 77
970108四子一乙 Lily 68 98 88
970109四子一乙 Cary Hu 89 59 49
970110四子一乙 趙大秋 70 60 100

```



```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct _student
5 {
6     char name[5];    //學生姓名
7     int chinese;     //國文成績
8     int english;     //英文成績
9     int math;        //數學成績
10    int sum;          //成績總和
11    float avg;        //成績平均
12 } student;
13
14 int main(void)
15 {
16     student data[5];
17     FILE *stream1;
18     FILE *stream2;
19     int i;
20
21     stream1=fopen("gradedata.txt","r");
22     stream2=fopen("gradedoutput.txt","w");
23
24     if (stream1 == NULL || stream2 == NULL)
25     {
26         printf("檔案開啟失敗\n");
27     }
28     else
29     {
30         fprintf(stream2,"姓名\t國文\t英文\t數學\t總合\t平均\n");
31         for (i=0;i<5;i++)
32         {
33             fscanf(stream1,"%s",data[i].name);
34             fscanf(stream1,"%d",&data[i].chinese);
35             fscanf(stream1,"%d",&data[i].english);

```

```

36             fscanf(stream1,"%d",&data[i].math);
37
38             data[i].sum=data[i].chinese+data[i].english+data[i].math;
39             data[i].avg=data[i].sum/3.0;
40             fprintf(stream2,"%s\t%d\t%d\t%d\t%d\t%.2f\n",
41                 data[i].name,data[i].chinese,data[i].english,
42                 data[i].math,data[i].sum,data[i].avg);
43         }
44         fclose(stream1);
45         fclose(stream2);
46     }
47     system("pause");
48     return 0;
49 }

```

gradedata.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```

Marry 30 20 54
Joe 58 62 75
Kile 65 98 78
Frank 18 43 69
Gray 61 58 78

```

gradedoutput.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

姓名	國文	英文	數學	總合	平均
Marry	30	20	54	104	34.67
Joe	58	62	75	195	65.00
Kile	65	98	78	241	80.33
Frank	18	43	69	130	43.33
Gray	61	58	78	197	65.67

- 設計程式可以讓信貸部經理得到下列三種客戶的清單
 - (選項1) 欠款餘額為零的客戶 (`balance==0`)
 - (選項2) 存款客戶 (`balance>0`)
 - (選項3) 借款客戶 (`balance<0`)
- `rewind(cfPtr);`
 - `cfPtr` 所指向的檔案位置指標 (`file position pointer`)，也就是檔案下一個要讀取或寫入的位元組，重新指回到檔案的開頭


```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int request;    //request number
7     int account;    //account number
8     double balance; //account balance
9     char name[30];  //account name
10    FILE *cfPtr;     //clients.txt file pointer
11
12    if ((cfPtr=fopen("clients.txt","r")) == NULL)
13    {
14        printf("File could not be opened\n");
15    }
16    else
17    {
18        printf("Enter request\n"
19              "1 - List accounts with zero balances\n"
20              "2 - List accounts with credit balances\n"
21              "3 - List accounts with debit balances\n"
22              "4 - End of run\n?");
23        scanf("%d",&request);
24
25        while (request != 4)
26        {
27            fscanf(cfPtr,"%d%s%lf", &account, name, &balance);
28            switch (request)
29            {

```

```

30
31            case 1:
32                printf("\nAccounts with zero balances:\n");
33                while (!feof(cfPtr))
34                {
35                    if (balance == 0)
36                    {
37                        printf("%-10d%-13s%7.2f\n",
38                              account, name, balance);
39                    }
40                    fscanf(cfPtr,"%d%s%lf",
41                          &account, name, &balance);
42                }
43                break;

```

```

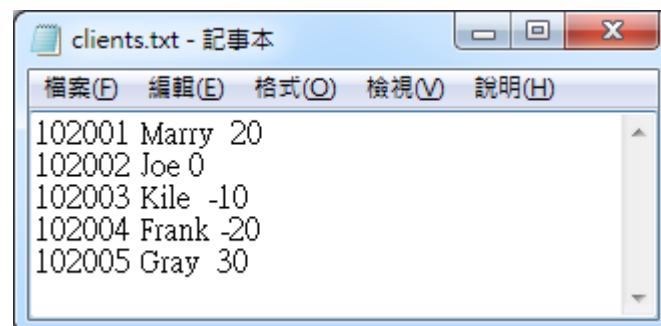
clients.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
102001 Marry 20
102002 Joe 0
102003 Kile -10
102004 Frank -20
102005 Gray 30

```

```

43 case 2:
44     printf("\nAccounts with credit balances:\n");
45     while (!feof(cfPtr))
46     {
47         if (balance < 0)
48         {
49             printf("%-10d%-13s%7.2f\n",
50                 account, name, balance);
51         }
52         fscanf(cfPtr, "%d%s%lf",
53             &account, name, &balance);
54     }
55     break;
56 case 3:
57     printf("\nAccounts with debit balances:\n");
58     while (!feof(cfPtr))
59     {
60         if (balance > 0)
61         {
62             printf("%-10d%-13s%7.2f\n",
63                 account, name, balance);
64         }
65         fscanf(cfPtr, "%d%s%lf",
66             &account, name, &balance);
67     }
68     break;
69 }
70 rewind(cfPtr); //return cfPtr to beginning of file
71 printf("\n?");
72 scanf("%d", &request);
73 }
74 printf("End of run.\n");
75 fclose(cfPtr);
76 }
77 system("pause");
78 return 0;
79 }

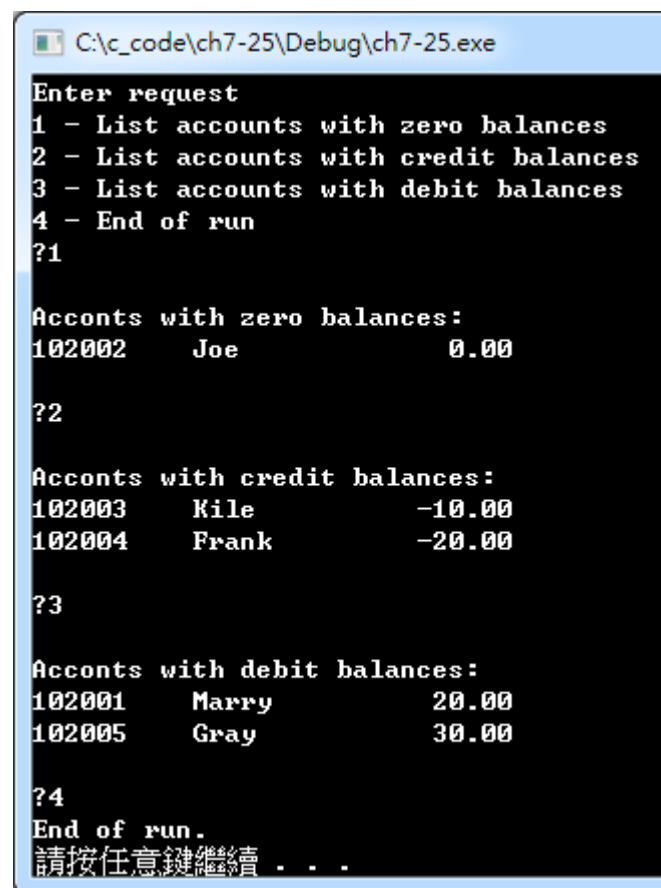
```



```

clients.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
102001 Marry 20
102002 Joe 0
102003 Kile -10
102004 Frank -20
102005 Gray 30

```



```

C:\c_code\ch7-25\Debug\ch7-25.exe
Enter request
1 - List accounts with zero balances
2 - List accounts with credit balances
3 - List accounts with debit balances
4 - End of run
?1

Accounts with zero balances:
102002      Joe              0.00

?2

Accounts with credit balances:
102003      Kile             -10.00
102004      Frank            -20.00

?3

Accounts with debit balances:
102001      Marry            20.00
102005      Gray             30.00

?4
End of run.
請按任意鍵繼續 . . .

```

函式	用途
fopen()	開啟檔案
fclose()	關閉檔案
fgetc()	從檔案串流中，讀出一個字元
fputc()	寫入一個字元到檔案串流內
fgets()	從檔案串流中，讀出一段文字
fputs()	寫入字串到檔案串流內
fscanf()	從檔案串流中，讀出一段固定格式的文字
fprintf()	寫入一段格式化的文字到檔案串流內
fseek()	移動檔案指標
fread()	讀出檔案串流資料
fwrite()	寫入資料到檔案串流