

Var and IO

製作者:吳忻珉

成功大學資工系醒獅團上課講義

Outline

- ▶ Review
- ▶ Variable and data type
- ▶ Operator
- ▶ I/O

Review

```
#include<stdio.h>
```

```
int main()  
{  
    /* First program*/  
  
    printf("Hello world");// printf function|  
  
    return 0;  
}
```

預處理指令:#開頭，編譯的時候由前處理器處理，會作簡單的代換，主要會包含要使用的library，這裡使用stdio.h

Review

```
#include<stdio.h>
```

```
int main()
{
    /* First program*/

    printf("Hello world");// printf function|

    return 0;
}
```

敘述：main function 是程式執行的部分，return 0是程
是結束的地方

Review

```
#include<stdio.h>

int main()
{
    /* This is
       first program
    */

    printf("Hello world");// printf function

    return 0;
}
```

註解：編譯器編譯時會忽視的部分，可以用來幫助了解程式碼

// 後接單行註解

/* 之後的文字會是註解直到 */ 前

Review

- ▶ 編譯指令：`gcc -o 執行黨名 原始碼`
- ▶ `gcc 原始檔名 (執行黨名為a.out)`
- ▶ 執行：`./執行黨名`
- ▶ Ex：`gcc -o test test.c`
`./test`
`gcc test.c`
`./a.out`

Variable and data type

- ▶ 程式中我們會宣告變數來儲存資料
- ▶ 變數是儲存在電腦的記憶體中
- ▶ 不同變數在記憶體中佔有不同大小的空間，佔用的空間會影響變數大小

Variable and data type

- ▶ 資料在電腦中以2進位計算
 - ▶ Ex : 8 bit 整數
 - ▶ 00001100 (2進位) => 12 (10進位)
- ▶ 電腦中的儲存空間是以byte來計算，8bit=1byte

Variable and data type

► 整數(以64位元的linux為例)

Type	Byte	boundary
short int	2	-32768~32767
unsigned short int	2	0~65535
int	4	-2147483648~2147483647
unsigned int	4	0~4294967295
long int	8	-9223372036854775808~ 9223372036854775807
unsigned long int	8	0~18446744073709551615
long long int	8	-9223372036854775808~ 9223372036854775807
unsigned long long int	8	0~18446744073709551615

Variable and data type

► 浮點數

Type	Byte	最小正數	最大數值
float	4	1.17549×10^{-38}	3.40280×10^{38}
double	8	2.22507×10^{-308}	1.79769×10^{308}

Variable and data type

► Char

Type	Byte	boundary
char	1	-128~127
unsigned char	1	0~255

► Char 顯示在螢幕上是依照ASCII 編碼

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

Source: www.LookupTables.com

Variable and data type

- ▶ 變數宣告
 - ▶ 變數型態 變數名稱
- ▶ Example:
 int a;
 int b,c,d;
 float e,f;
 char h;

Variable and data type

- ▶ 變數的名稱可以包含英文字母，數字和底線，但是不能數字開頭，ex: 2a (X)，a2 (O)
- ▶ 變數的名稱也不能是關鍵字
- ▶ 變數的名稱不可以重複
 - ▶ Ex: `int a; float a;` (X)
- ▶ 在取變數名稱時，請取有意義的名稱，減少程式碼維護的問題

Variable and data type

► 一些常用關鍵字：

auto	enum	restrict	unsigned
break	extern	return	void
case	float	short	volatile
char	for	signed	while
const	goto	sizeof	_Bool
continue	if	static	_Complex
default	inline	struct	_Imaginary
do	int	switch	else
double	long	typedef	union
register			

Variable and data type

- ▶ 初始化，變數宣告時就給他一個值，使用沒有初始化的變數，會有預期外的結果

- ▶ Example:

```
int a=2;
```

```
int b=3,c=7,d=9;
```

```
int e,f,g=10; //e,f 沒有初始化，e初始化為10
```

```
char test='a'; //test代表字元a，但是test的值是97
```

```
char test2='1'; //test2代表字元1，但是test2的值是49
```


Variable and data type

- ▶ Assignment (賦值)
- ▶ 等號 = 是用來將等號右邊的數值放到左邊變數的記憶體空間，不是數學符號等於的意思
- ▶ Ex:

```
int a,b;
```

```
a=2;
```

```
b=3;
```

Operator

- ▶ Arithmetic operators
- ▶ Unary: +, -
- ▶ Binary: + , - , * , / , %(取餘數)
- ▶ Example:
 - ▶ `int i=0,j=0;`
 - ▶ `i+1; j+1;` //這行不會改變i和j的值，僅得到運算結果
 - ▶ `i=+1; j=-1;`
 - ▶ `i=2+j ;`
 - ▶ `j=5%2;`

Operator

- ▶ = 左邊要放 lvalue，lvalue表示一個有存在記憶體的可變數，不能是常數或是運算結果
 - ▶ `12 = i ; // wrong`
 - ▶ `i+j=0; //wrong`
 - ▶ `-i=j; // wrong`

Operator

- ▶ Compound assignment

- ▶ `+=` , `-=` , `/=` , `%=`

- ▶ Ex:

- ▶ `i+=2; // i=i+2;`

- ▶ `i-=2; //i=i-2;`

- ▶ `i*=2; //i=i*2;`

- ▶ `i%=2; //i=i%2;`

Operator

- ▶ ++ , -- (Increment and decrement operators)
- ▶ Example :
 - ▶ `int a=5;`
 - ▶ `printf(" a is %d\n",a++);`
 - ▶ `printf(" a is %d\n",a);`
 - ▶ `printf ("a is %d\n",++a);`
 - ▶ `printf("a is %d\n",a);`

```
a is 5
a is 6
a is 7
a is 7
```

Operator

- ▶ operator的優先度，()中的最優先

precedence	operators
1	++ 、 -- (postfix)
2	++ 、 -- (prefix) 、 + 、 - (unary)
3	* 、 / 、 %
4	+ 、 -
5	= 、 *= 、 /= 、 %= 、 += 、 -=

- ▶ 附註一般數學運算式我們習慣寫的[]和{}，在C語言中一律用()，[]和{}在C語言有其他意思

Operator

► Ex:

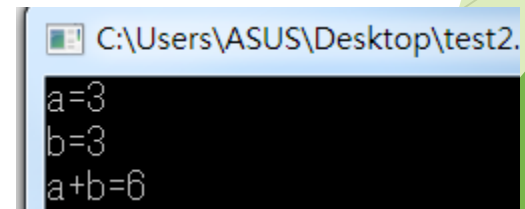
- float a;
- a=3/5; // 答案會是0，因為把3跟5作整數運算
- a=3.0/5.0 // 答案是3/5的浮點數
- a=(float)3/(float)5 //可以在資料前面加上(type)使資料強制轉換成其他type

I/O

- ▶ `printf("string");`
- ▶ `printf` 是基本用來輸出的函式，`()`中要放函式要的參數
- ▶ `printf` 需要字串作為參數，`"string"`代表一個字串
- ▶ 要用`printf`，一定要`#include<stdio.h>`，`printf`是被寫在`stdio.h`中的函式
- ▶ `stdio.h`是 Standard Input and Output Library，內有許多基本輸出輸入的函式

I/O

- ▶ 輸出資料
- ▶ `printf(string , expr1,expr2,...)`
- ▶ `string` 裡包含了要輸出的內容及輸出資料的格式
- ▶ 後面的`expr1,expr2,...` 是要輸出的資料
- ▶ Ex:
 - ▶ `int a=3,b=3;`
 - ▶ `printf("a=%d\nb=%d\na+b=%d",a,b,a+b);`



```
C:\Users\ASUS\Desktop\test2.  
a=3  
b=3  
a+b=6
```

I/O

- ▶ Conversion specifications
- ▶ %開頭後面接一個字母
- ▶ 在輸出的時候，會被printf後面接的參數代換掉

Type	Conversion specifications
char	%c
int	%d
unsigned int	%u
short int	%hd
unsigned short int	%hu
long int	%ld
unsigned long int	%lu
long long int	%lld
unsigned long long int	%llu
float	%f or %e 輸出科學記號
double	%f or %e 輸出科學記號
字串	%s

escape character

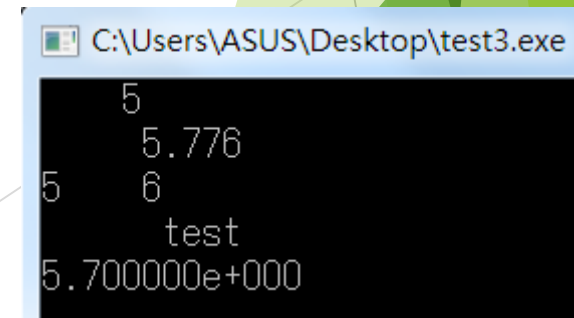
- ▶ `\n` : 換行
 - ▶ `\t` : tab
 - ▶ `\\` : 印出\
Example : `printf("I love \n\"ncku\" \t\\csie\\");`
 - ▶ `\"` : 印出"
- I love
"ncku" \csie\

I/O

▶ 可以在%和字母間加入數字來表示輸出要用幾格

▶ Ex:

- ▶ `printf("%5d",5);` /*代表輸出的數字佔五格，多的從前面補齊*/
- ▶ `printf("%10.2f",5.775757);` /*代表輸出10格，小數點後的精準度是2*/
- ▶ `printf("%-5d%d",5,6);` /*小於0代表多的格子從後面補*/
- ▶ `printf("%10s", "test");`
- ▶ `printf("%e",5.7);`



C:\Users\ASUS\Desktop\test3.exe

```
5
 5.776
5  6
      test
5.700000e+000
```

I/O

- ▶ `scanf(string, pointer1, pointer2.....);`
- ▶ `string` 中要打你準備要輸入的內容，`pointer1, pointer2..` 是你要儲存變數的位址
- ▶ `scanf`遇到空格或是換行符號會跳過不讀取，直到下一個輸入才會讀取
- ▶ **Ex:**
 - ▶ `int a,b;`
 - ▶ `scanf(“%d%d” ,&a,&b);` //記得在變數前加上`&`，表示你要儲存到的變數位址

I/O

- ▶ scanf 的字串中的Conversion specifications和前面printf的是一樣的
- ▶ 不要在scanf 的字串中加\n
- ▶ Ex:
 - ▶ int a,b;
 - ▶ scanf(“%d/%d” ,&a,&b); /*這樣輸入的時候，可以輸入
2/3 */

```
#include<stdio.h>
```

```
int main()  
{  
    int a,b,c,d;  
    scanf("%d%d",&a,&b);  
    scanf("%d/%d",&c,&d);  
    printf("a:%d\nb:%d\nc:%d\nd:%d\n",a,b,c,d);  
}
```

```
1      2  
3/4  
a:1  
b:2  
c:3  
d:4
```


練習一

- ▶ 練習輸入兩個數字後輸出兩數字相加相減相乘相除後的結果(要用排版，數字要對齊)

```
input a and b:25 4
a:      25
b:       4
a+b:    29
a-b:    21
a*b:   100
a/b:     6
```

練習二

- ▶ 華氏溫度轉攝氏溫度
- ▶ 由鍵盤輸入華氏溫度,輸出攝氏溫度
- ▶ $C = 5/9 * (F - 32)$

練習三

- ▶ 輸入使用者的身高和體重，計算出bmi

補充(電腦中表示變數的方式)

- ▶ 整數
- ▶ 10進位表示: $125 \Rightarrow 1 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$
- ▶ 2進位表示:
- ▶ $010001 \Rightarrow 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 17$ (10進位數值)

- ▶ 這裡以8bit為例

1	2	3	4	5	6	7	8
正負	2進位表示數值						

- ▶ 第一位表示正負，0為正，1為負，後面七位表示數值
- ▶ 最大值為 $2^7-1=127$
- ▶ Ex: 25 => 00011001
75 => 01001011

▶ 負數並不是直接將第一位設為1，後直接接正數數值

▶ 電腦中表示負整數是用2的補數

▶ 先找出一個正數，然後將1變成0，0變成1，再加1

▶ Ex :找-2

▶ $2 \Rightarrow 00000010$ ，將1變成0，0變成1， 111111101

▶ 再加1， $11111110 \Rightarrow -2$

▶ $37 \Rightarrow 00100101$ ， $-37 \Rightarrow 11011011$

▶ $124 \Rightarrow 01111100$ ， $-124 \Rightarrow 10000100$

▶ 也可用負數加正數=0來檢驗

▶ 8bit的最小值是 $10000000 \Rightarrow -128 = -2^7$

補充(浮點數)

- ▶ 以float為例，共32bit



- ▶ **sign**: 第31個bit表示正負，1是負，0是正
- ▶ **exponent**: 23~30bit 共8bit表示，為指數加上exponent bias的整數，exponent bias是 $2^7-1=127$
- ▶ **fraction** : 0~22bit 共23bit表示，表示小數部分

▶ 計算方式: $4.25 \Rightarrow 100.01 = 1 \cdot 2^2 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2}$

▶ $100.01 = 1.0001 \cdot 2^2$

▶ 指數是 2, $2 + 127 = 129 \Rightarrow \text{exponent} = 10000001$

▶ 小數部分是, 001

$\text{fraction} = 000100000000000000000000$

▶ 所以

▶ $4.75 \Rightarrow 01000000100010000000000000000000$

▶ $-4.75 \Rightarrow 11000000100010000000000000000000$

- ▶ 以double為例，共64bit



- ▶ **sign**: 第63個bit表示正負，1是負，0是正
- ▶ **exponent**: 52~62bit 共11bit表示，為指數加上exponent bias的整數，exponent bias是 $2^{10}-1=1023$
- ▶ **fraction** : 0~51bit 共52bit表示，表示小數部分