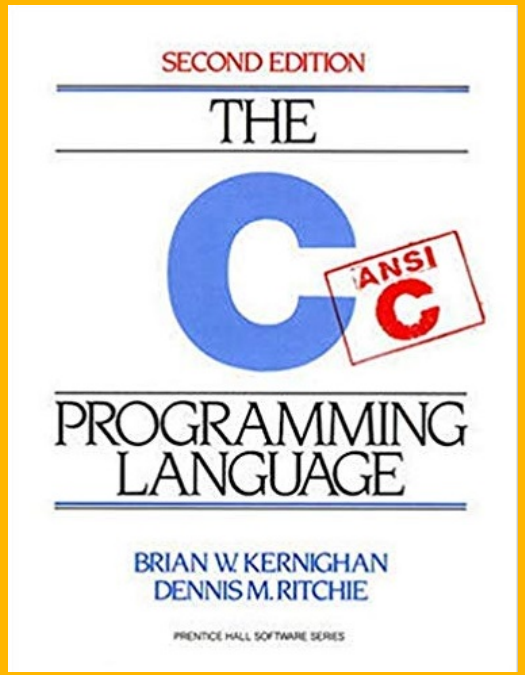


Introduction to Computer Science Fall 2022 #15

Chi-Jen Wu

```c

#include <stdio.h>

int main() {
    printf("顯示字元          = %c\n", 'A');
    printf("顯示字元編碼      = %d\n", 'A');
    printf("顯示字元編碼      = %c\n", 65);
    printf("顯示十進位整數    = %d\n", 15);
    printf("顯示十六進位整數= %X\n", 15);
    printf("顯示十六進位整數= %x\n", 15);
    printf("字串              = %s\n", "顯示字串");
    return 0;
}

```

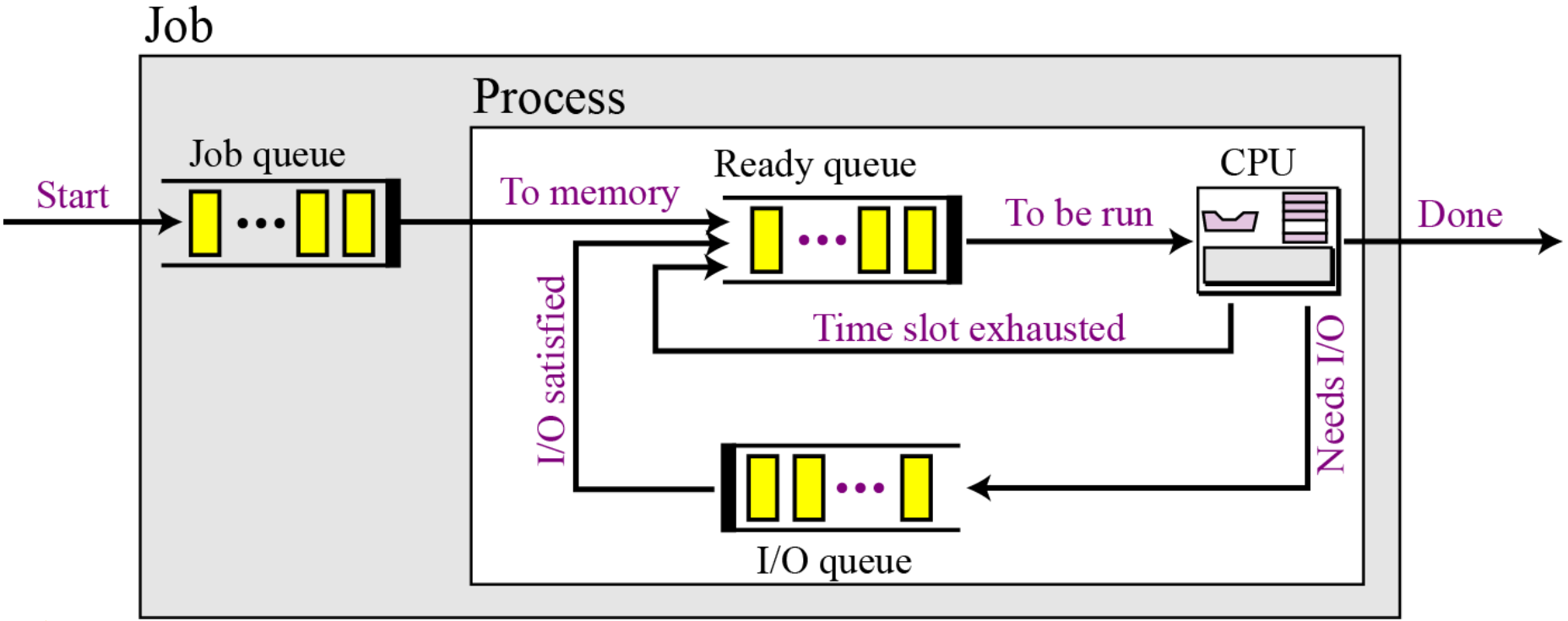
```
顯示字元          = A
顯示字元編碼      = 65
顯示字元編碼      = A
顯示十進位整數    = 15
顯示十六進位整數= F
顯示十六進位整數= f
字串              = 顯示字串


...Program finished with exit code 0
Press ENTER to exit console.
```

printf(), this function 這是一個非常重要的 function

```c
#include <stdio.h>
int main() {
    printf("\t\t %s Love %s\n", "IU", "CGU");
    return 0;
}
```

```
                IU Love CGU

...Program finished with exit code 0
Press ENTER to exit console.
```

程式編譯好後開始執行


Job
Process
Job queue
Start
To memory
Ready queue
CPU
To be run
Done
Time slot exhausted
I/O satisfied
Needs I/O
I/O queue

Topics

- Problem Solving with Programming Language
- C Programming
  - C Basics
    - Variables
    - Flow of Control
    - Function Basics
    - Programming with Arrays
    - Strings
    - Structures
    - Streams and File I/O
- Cloud Platform/Cloud Shell Editor (gcc/g++/Makefile)
- Cloud Platform/Cloud Source Repositories (git)

Variables

- Identifiers, 變數的名字
  - 開頭只能是
    - A to Z
    - a to z
    - underscore ‘_’
    - 不能有 ~, !, @, $, %, &, ^, *, (,), -, +, =, `, .
- C is a **case-sensitive** programming language
  - A123 不等於 a123

```c
#include <stdio.h>
int main(void) {
    int _0123;
    int 0123;
    return 0;
}
```

input

Compilation failed due to following error(s).

```
main.c:4:9: error: expected identifier or '(' before numeric constant
    4 |     int 0123;
      |         ^~~~
```

Keywords in C Programming			
auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

這些keywords 也不能宣告成變數

```c
#include <stdio.h>
int main(void) {
    int _0123;
    int my_name;
    int My_Name;
    int a123;
    int a123;
    return 0;
}
```

input

Compilation failed due to following error(s).

```
main.c:7:9: error: redeclaration of ‘a123’ with no linkage
    7 |     int a123;
      |         ^~~~
main.c:6:9: note: previous declaration of ‘a123’ was here
    6 |     int a123;
      |         ^~~~
```

宣告了兩個一樣的變數！

宣告的變數有特殊符號

```c
#include <stdio.h>
int main(void) {
    int a123%;
    int a123-;
    int a123=;
    return 0;
}
```

input

Compilation failed due to following error(s).

```
main.c:3:13: error: expected '=', ',', ';', 'asm' or '__attribute__' before '%' token
    3 |     int a123%;
      |             ^
main.c:3:13: error: expected expression before '%' token
main.c:4:13: error: expected '=', ',', ';', 'asm' or '__attribute__' before '-' token
    4 |     int a123-;
      |             ^
main.c:4:14: error: expected expression before ';' token
    4 |     int a123-;
      |              ^
main.c:5:14: error: expected expression before ';' token
    5 |     int a123=;
      |              ^
```

```c
#include <stdio.h>
int main(void) {
    int int;
    int else;
    int if;
    return 0;
}
```

宣告的變數有關鍵字存在

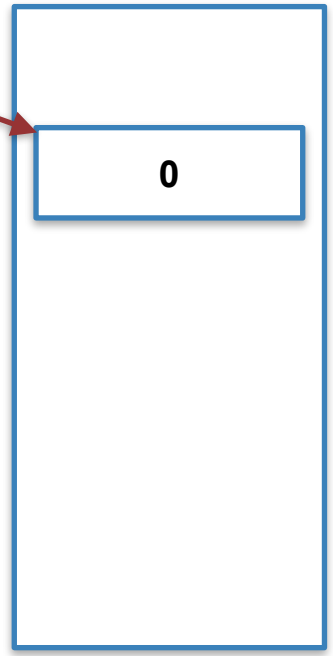
input

Compilation failed due to following error(s).

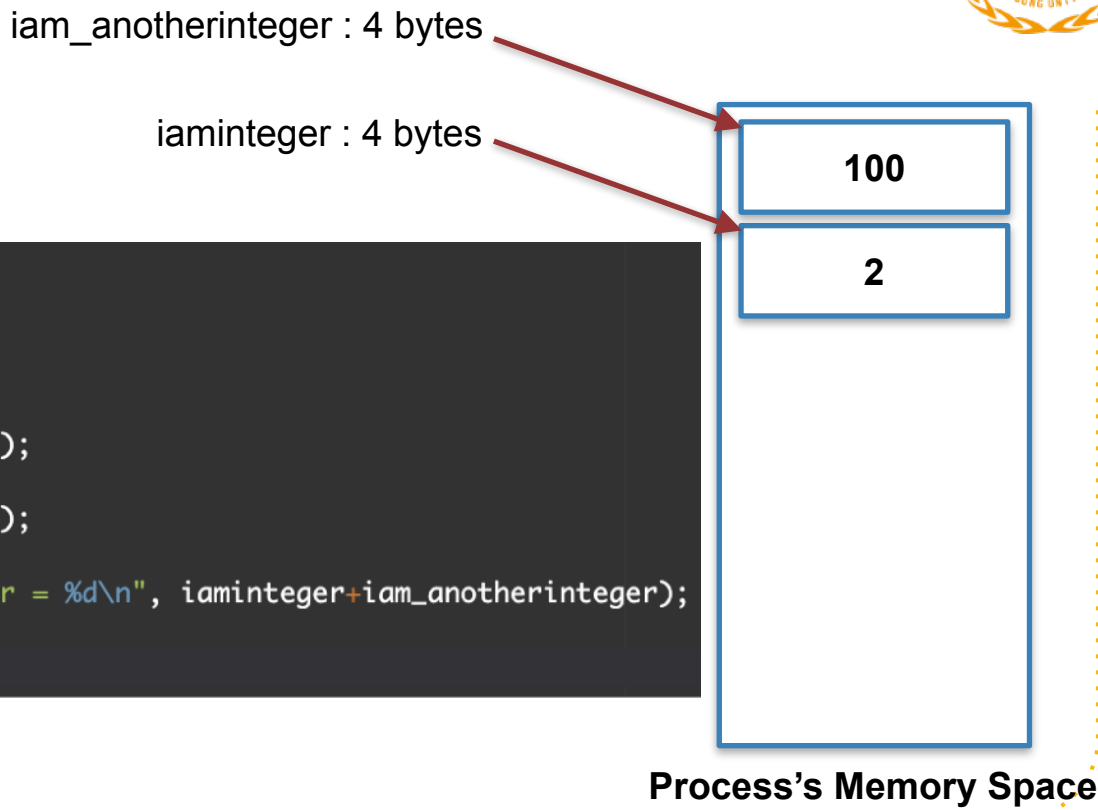
```
main.c:3:9: error: two or more data types in declaration specifiers
    3 |     int int;
      |         ^~~
main.c:3:5: warning: useless type name in empty declaration
    3 |     int int;
      |     ^~~
main.c:4:9: error: expected identifier or '(' before 'else'
    4 |     int else;
      |         ^~~~
main.c:5:9: error: expected identifier or '(' before 'if'
    5 |     int if;
      |         ^~
```

Variables：佔記憶體空間

```
3  int main() {
4      int iaminteger;
5      iaminteger = 0;
6      iaminteger = iaminteger + 1;
7      printf("iaminteger = %d\n", iaminteger);
8      iaminteger = iaminteger + 1;
9      printf("iaminteger = %d\n", iaminteger);
10     return 0;
11 }
```




int main() {
int iaminteger;
int iam_anotherinteger;
iaminteger = 0;
iaminteger = iaminteger + 1;
printf("iaminteger = %d\n", iaminteger);
iaminteger = iaminteger + 1;
printf("iaminteger = %d\n", iaminteger);
iam_anotherinteger = 100;
printf("iaminteger + iam_anotherinteger = %d\n", iaminteger+iam_anotherinteger);
return 0;
}


iam_anotherinteger : 4 bytes
iaminteger : 4 bytes
100
2
Process's Memory Space

Variables：型別

- void
  - 空，常用
- integer
  - 最多種，常用
- floating-point
  - 三種，不太常用
- 自定義變數
  - Structure type，常用，後面講

80%時間都是在操作這些變數！

void 空

- Function **arguments** as void
- Function **returns** as void，之後介紹
- **Pointers** to void，之後介紹

```c
#include <stdio.h>
int main(void) {
    printf("Hello, World!\n");
    return 0;
}
```

integer 整數

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4bytes for 32 bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

floating-point 浮點數

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

IEEE 754, 32bit, 64bit, 80bit

Variables：操作！

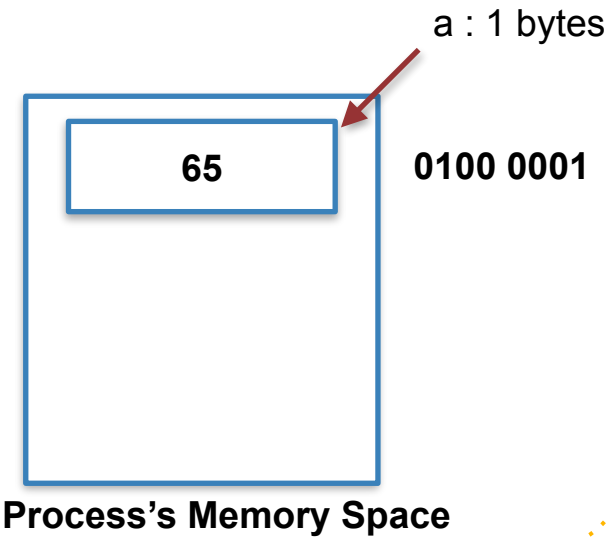
- integer
  - **char**, 8bit, ASCII
- **+**, -, *, /, **%**, **++**, — (Arithmetic operations)
- **&, |, ^, ~, <<, >>** (Bitwise Operators)
- **==, !=**, >, <, >=, <= (Relational Operators)
- Assignment Operators
  - =, +=, -=, *=, /=, %=, **<<=, >>=, &=, ^=, |=**
- Etc
  - sizeof(), ?: (三元運算子)

```c
#include <stdio.h>
int main(void) {
    char a = 65;
    printf("%d\n", a+=2);
    printf("%d\n", a-=2);
    printf("%d\n", a*=2);
    return 0;
}
```

```
67
65
-126


...Program finished with exit code 0
Press ENTER to exit console.
```

a+=2

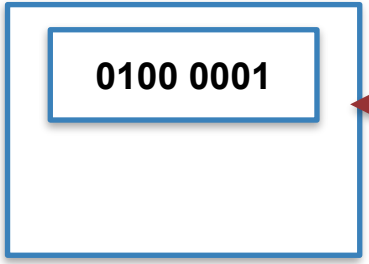
a = a + 2



```c
#include <stdio.h>
int main(void) {
    char a = 65;
    printf("%d\n", a+=2);
    printf("%d\n", a-=2);
    printf("%d\n", a*=2);
    return 0;
}
```

```
67
65
-126
...Program finished with exit code 0
Press ENTER to exit console.
```

1000 0010



Process's Memory Space

Eight-bit signed integers

Decimal value	Two's-complement representation
0	0000 0000
1	0000 0001
2	0000 0010
126	0111 1110
127	0111 1111
−128	1000 0000
−127	1000 0001
−126	1000 0010
−2	1111 1110
−1	1111 1111

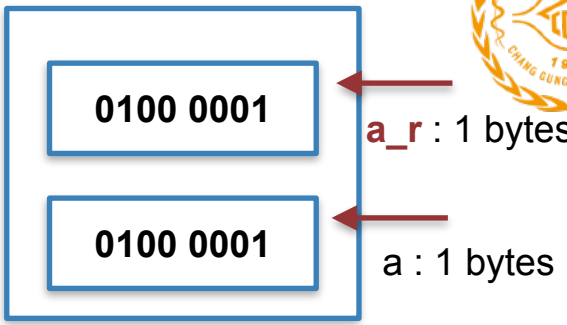
```c
#include <stdio.h>
int main(void) {
    char a = 'A';
    printf("a = %d\n", a+1);
    printf("a = %d\n", a++);
    printf("a = %d\n", a);
    printf("a = %d\n", ++a);
    return 0;
}
```

```
a = 66
a = 65
a = 66
a = 67


...Program finished with exit code 0
Press ENTER to exit console.
```

前＋＋
先加再做
後＋＋
先做再加

```c
#include <stdio.h>
int main(void) {
    char a = 'A';
    printf("a = %d\n", a+1);
    printf("a = %d\n", a++);
    printf("a = %d\n", a);
    printf("a = %d\n", ++a);
    return 0;
}
```

```
a = 66
a = 65
a = 66
a = 67


...Program finished with exit code 0
Press ENTER to exit console.
```

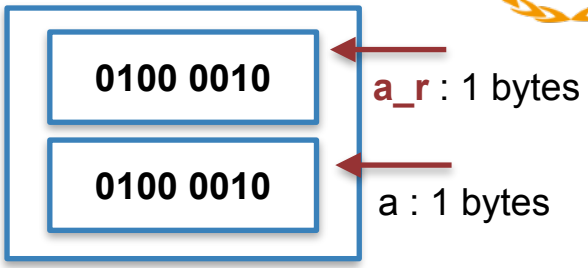


Process's Memory Space

printf("a = %d", a++);
被拆解為
a_r = a
a = a +1
printf a_r

```c
#include <stdio.h>
int main(void) {
    char a = 'A';
    printf("a = %d\n", a+1);
    printf("a = %d\n", a++);
    printf("a = %d\n", a);
    printf("a = %d\n", ++a);
    return 0;
}
```

```
a = 66
a = 65
a = 66
a = 67


...Program finished with exit code 0
Press ENTER to exit console.
```

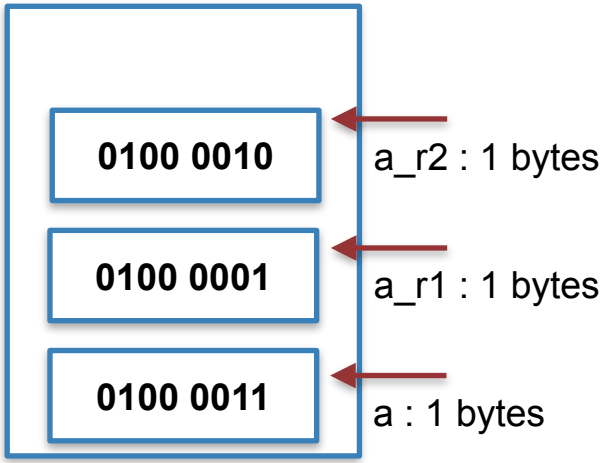


Process's Memory Space

printf("a = %d", ++a);
被拆解為
a_r = a +1
a = a +1
printf a_r

```
3 int main() {
4     char a = 'A';
5     printf("a=%d, a=%d\n", a++, a++);
6     return 0;
7 }
```

printf(“a = %d, a=%d”, a++, a++);
被拆解為
a_r1 = a
a = a + 1
a_r2 = a
a = a + 1
printf a_r2, a_r1
此時 a 是多少？



Process’s Memory Space

```c
#include <stdio.h>
int main(void) {
    char a = 65;
    printf("%d mod %d = %d\n", a, 7, a%7);
    printf("%d mod %d = %d\n", a, 7, a%2);
    printf("%d mod %d = %d\n", a, 7, a%5);
    return 0;
}
```

```
65 mod 7 = 2
65 mod 2 = 1
65 mod 5 = 0


...Program finished with exit code 0
Press ENTER to exit console.
```

取餘數%

慢

計算BMI的值

- BMI = 體重(公斤) / 身高^2 (公尺^2)

```c
#include <stdio.h>
int main() {
    float h, w, bmi;
    h = 1.55;
    w = 40;
    bmi = w/(h*h);
    printf("bmi = %.4f\n", bmi);
    return 0;
}
```

```
bmi = 16.6493


...Program finished with exit code 0
Press ENTER to exit console.
```

華氏溫度轉攝氏溫度

- Cel = (Fah-32)*(5.0f/9)

```c
#include <stdio.h>
int main() {
    int Fah;
    float Cel;
    Fah = 80;
    Cel = (Fah - 32)*(5.0f / 9);
    printf("\tFah %d = Cel = %.4f\n", Fah , Cel);
    return 0;
}
```

```
        Fah 80 = Cel = 26.6667


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include <stdio.h>
int main(void) {
    char a = 'A';
    char i;
    i = a++;
    // i = a; a = a + 1
    printf("%d\n", i);
    a = 'A';
    i = ++a;
    // a = a + 1, i = a
    printf("%d\n", i);
    return 0;
}
```

```
65
66


...Program finished with exit code 0
Press ENTER to exit console.
```

Comments
註解

```c
// inline comments

/* comment goes here */

/*
 * comment goes here
 */
```

Bitwise Operators

- &, |, ^, ~, <<, >>
- (很重要！)

```c
#include <stdio.h>
int main(void) {
    char a = 5;
    printf("\t%d\n", a&1);
    // 0101 & 0001
    printf("\t%d\n", a&2);
    // 0101 & 0010
    printf("\t%d\n", a|3);
    // 0101 | 0011
    return 0;
}
```

```
        1
        0
        7


...Program finished with exit code 0
Press ENTER to exit console.
```

Bitwise Operators

- &, |, ^, ~, <<, >>
- (很重要！)

```c
#include <stdio.h>
int main(void) {
    char a = 5;
    printf("\t%d\n", a^1);
    printf("\t%d\n", ~a);
    printf("\t%d\n", a>>1);
    printf("\t%d\n", a<<1);
    return 0;
}
```

```
        4
        -6
        2
        10


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include <stdio.h>
int main(void) {
    char a = 5;
    printf("\t%d\n", a^1);
    // 00000101 ^ 00000001 XOR
    printf("\t%d\n", ~a);
    // ~0000 0101 = 1111 1010 NOT
    printf("\t%d\n", a>>1);
    // 00000101 >> 1 = 00000010
    printf("\t%d\n", a<<1);
    // 00000101 << 1 = 00001010
    return 0;
}
```

```
4
-6
2
10


...Program finished with exit code 0
Press ENTER to exit console.
```

Bitwise Operators 可以幹嘛？

```c
#include <stdio.h>
int main() {
    // 65 - 97 = 32
    char a = 'A';
    printf("\t%c\n", a+32);
    a = 'a';
    printf("\t%c\n", a-32);
    return 0;
}
```

```
a
A


...Program finished with exit code 0
Press ENTER to exit console.
```

大小寫轉換

需要判斷現在是大寫還是小寫！

```c
#include <stdio.h>
int main() {
    char a = 'A';
    printf("\t%c\n", a^' ');
    a = 'a';
    printf("\t%c\n", a^' ');
    return 0;
}

```

```
    a
    A


...Program finished with exit code 0
Press ENTER to exit console.
```

大小寫轉換

```c
#include <stdio.h>
int main() {
    char a = 'a';
    char b = 'b';
    a ^= b;
    b ^= a;
    a ^= b;
    printf("\ta=%c, b=%c\n", a, b);
    return 0;
}
```

```
    a=b, b=a


...Program finished with exit code 0
Press ENTER to exit console.
```

交換數值

請打開online complier 試一下！十分鐘！

```c
#include <stdio.h>
int main() {
    char n = -6;
    char mask = n >> ( 8 - 1);
    printf("\twhat? %d\n", ((n ^ mask) - mask));
    return 0;
}
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

這會跑出什麼

```c
#include <stdio.h>
int main() {
    char n = -99;
    // 1111 1010
    char mask = n >> ( 8 - 1);
    // 1111 1111 (-), 0000 0000 (+)
    printf("\tabs %d\n", ((n ^ mask) - mask));
    // 1111 1010 ^ 1111 1111 = 0000 0101
    // 0000 0101 - 1111 1111 = 5 - (-1) = 6
    return 0;
}
```

8 bits

```
        abs 99


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
#include <stdio.h>
int main() {
    char n = -6;
    // 1111 1010
    char mask = n >> (sizeof(char)*8 - 1);
    // 1111 1111 (-) 0000 0000 (+)
    printf("\t abs %d\n", ((n ^ mask) - mask));
    // 1111 1010 ^ 1111 1111 = 0000 0101
    // 0000 0101 - 1111 1111 = 5 - (-1) = 6
    return 0;
}
```

計算變數型別佔多少bits

```
        abs 6


...Program finished with exit code 0
Press ENTER to exit console.
```

sizeof()

```c
#include <stdio.h>
int main() {
    printf("\tsizeof(int) %lu byte\n", sizeof(int));
    printf("\tsizeof(char) %lu byte\n", sizeof(char));
    printf("\tsizeof(float) %lu byte\n", sizeof(float));
    printf("\tsizeof(double) %lu byte\n", sizeof(double));
    printf("\tsizeof(unsigned char) %lu byte\n", sizeof(unsigned char));
    return 0;
}
```

input

```
        sizeof(int) 4 byte
        sizeof(char) 1 byte
        sizeof(float) 4 byte
        sizeof(double) 8 byte
        sizeof(unsigned char) 1 byte


...Program finished with exit code 0
Press ENTER to exit console.
```

這很重要！常用！

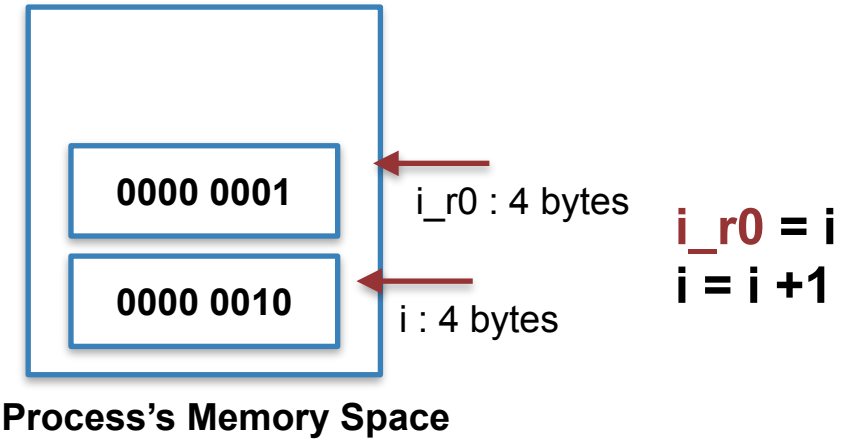
運算子的優先順序 (大部分是左到右)

```c
#include <stdio.h>
int main() {
    int i = 1;
    printf("\tWhat is: %d\n", i++ << 2 + 3 << --i);
    return 0;
}
```

```c
#include <stdio.h>
int main() {
    int i = 1;
    printf("\tWhat is: %d\n", i++ << 2 + 3 << --i);
    return 0;
}
```

優先權

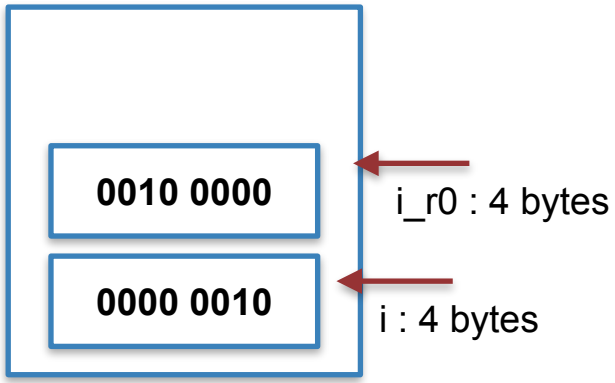
5



i_r0 = i
i = i +1

```c
#include <stdio.h>
int main() {
    int i = 1;
    printf("\tWhat is: %d\n", i++ << 2 + 3 << --i);
    return 0;
}
```

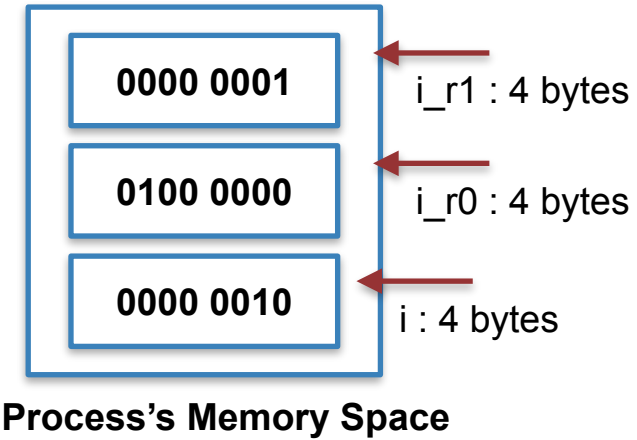
優先權



i_r0 = i
i = i +1
i_r0 << 5 = 32

```c
#include <stdio.h>
int main() {
    int i = 1;
    printf("\tWhat is: %d\n", i++ << 2 + 3 << --i);
    return 0;
}
```

優先權



i_r0 = i
i = i +1
i_r0 << 5 = 32

i_r1 = i - 1
i = i_r1
32 << i_r1
64

運算子的優先順序

```c
#include <stdio.h>
int main() {
    int i = 1;
    printf("\t%d\n", (i++)<<(2+3)<<(--i));
                    // i=i, i=i+1, << 5 << i=i-1, i=2
    return 0;
}
```

input

```
64


...Program finished with exit code 0
Press ENTER to exit console.
```

Precedence	Operator	Description	Associativity
1	++ --	Suffix/postfix increment and decrement	Left-to-right
	()	Function call	
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(*type*){*list*}	Compound literal(C99)	
2	++ --	Prefix increment and decrement[note 1]	Right-to-left
	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(*type*)	Cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof	Size-of[note 2]	
	_Alignof	Alignment requirement(C11)	
3	* / %	Multiplication, division, and remainder	Left-to-right
4	+ -	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10	|	Bitwise OR (inclusive or)	
11	&&	Logical AND	
12	||	Logical OR	
13	?:	Ternary conditional[note 3]	Right-to-left
14[note 4]	=	Simple assignment	
	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= |=	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

運算子的優先順序
能記多少算多少
多寫就會記得

https://docs.microsoft.com/zh-tw/cpp/c-language/precedence-and-order-of-evaluation?view=msvc-170

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4bytes for 32 bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615

Flow of Control

控制程式流程的方法

媽媽請你去超市買一瓶醬油

你出門時告訴你：

“**如果有蘋果，買六顆蘋果**”

所以回家應該會有兩種狀況：

超市有賣蘋果，你買了六顆蘋果

超市沒賣蘋果，你買了一瓶醬油

Flow of Control

去A7捷運站的演算法
往前100公尺
然後左轉直行50公尺
如果是紅燈，停50秒
往前300公尺
右轉直行10公尺
到達目的地

Flow of Control

- **if**
- **If else**
- **If else if else**
- **Switch case**
- While
- For
- Do while

Expression is true.

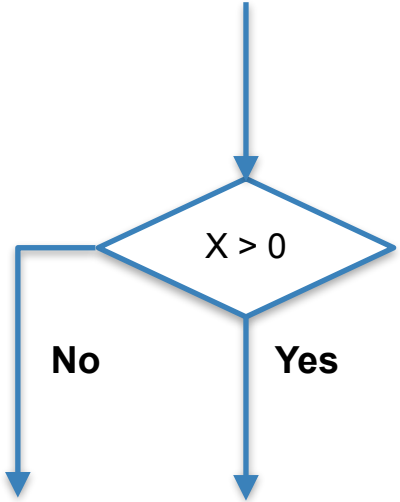
```
int test = 5;

if (test < 10)
{
  // codes
}

// codes after if
```

Expression is false.

```
int test = 5;

if (test > 10)
{
  // codes
}

// codes after if
```

if statement

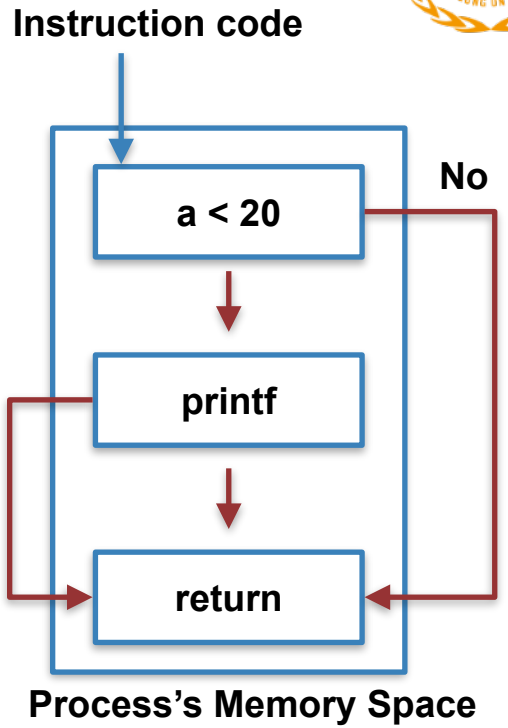
- Consists of a **boolean expression** followed by one or more statements



if statement

```c
#include <stdio.h>
int main() {
    int a = 10;

    if (a < 20) {
        printf("a is less than 20\n" );
    }

    return 0;
}
```

```
a is less than 20


...Program finished with exit code 0
Press ENTER to exit console.
```



```c
#include <stdio.h>
int main() {
    int a = 10;

    if (a < 20) {
        printf("a is less than 20\n" );
    } else {
        printf("a is not less than 20\n" );
    }

    return 0;
}
```

```
a is less than 20

...Program finished with exit code 0
Press ENTER to exit console.
```

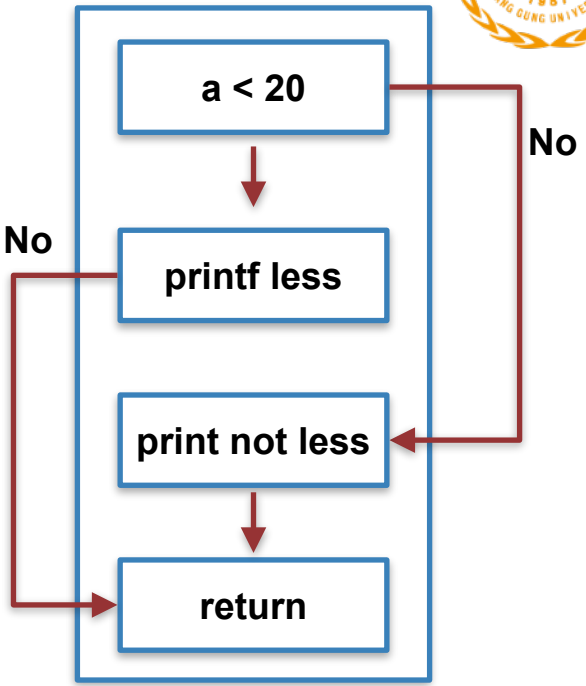
if...else statement

Code Style !!!!!!

```c
#include <stdio.h>
int main() {
    int a = 10;

    if (a < 20) {
        printf("a is less than 20\n" );
    } else {
        printf("a is not less than 20\n" );
    }

    return 0;
}
```

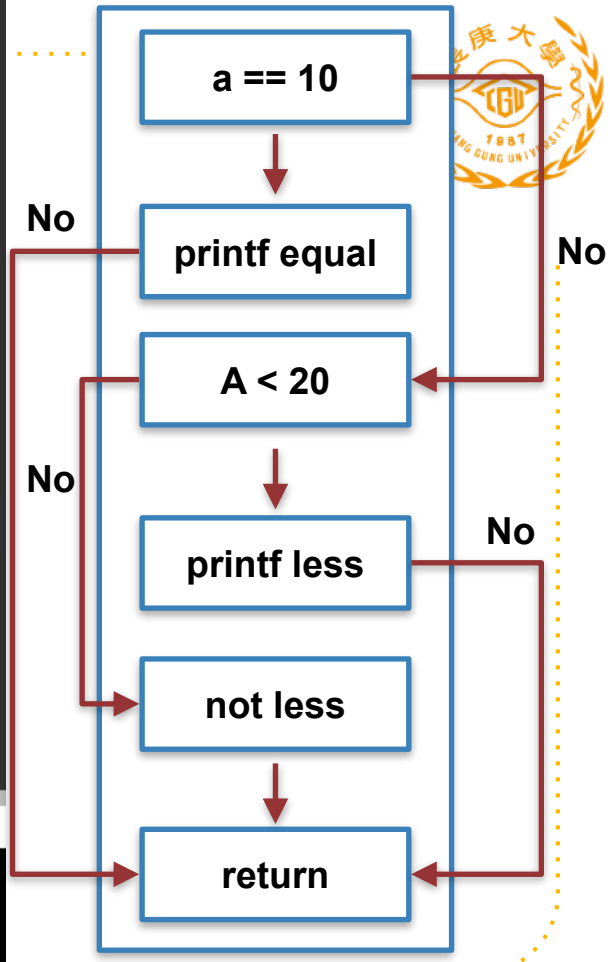
```
a is less than 20

...Program finished with exit code 0
Press ENTER to exit console.
```

if...else statement



```c
#include <stdio.h>
int main() {
    int a = 10;

    if (a == 10) {
        printf("a is equal to 10\n");
    } else if (a < 20) {
        printf("a is less than 10\n");
    } else {
        printf("a is not less than 20\n");
    }

    return 0;
}
```

```
a is equal to 10


...Program finished with exit code 0
Press ENTER to exit console.
```



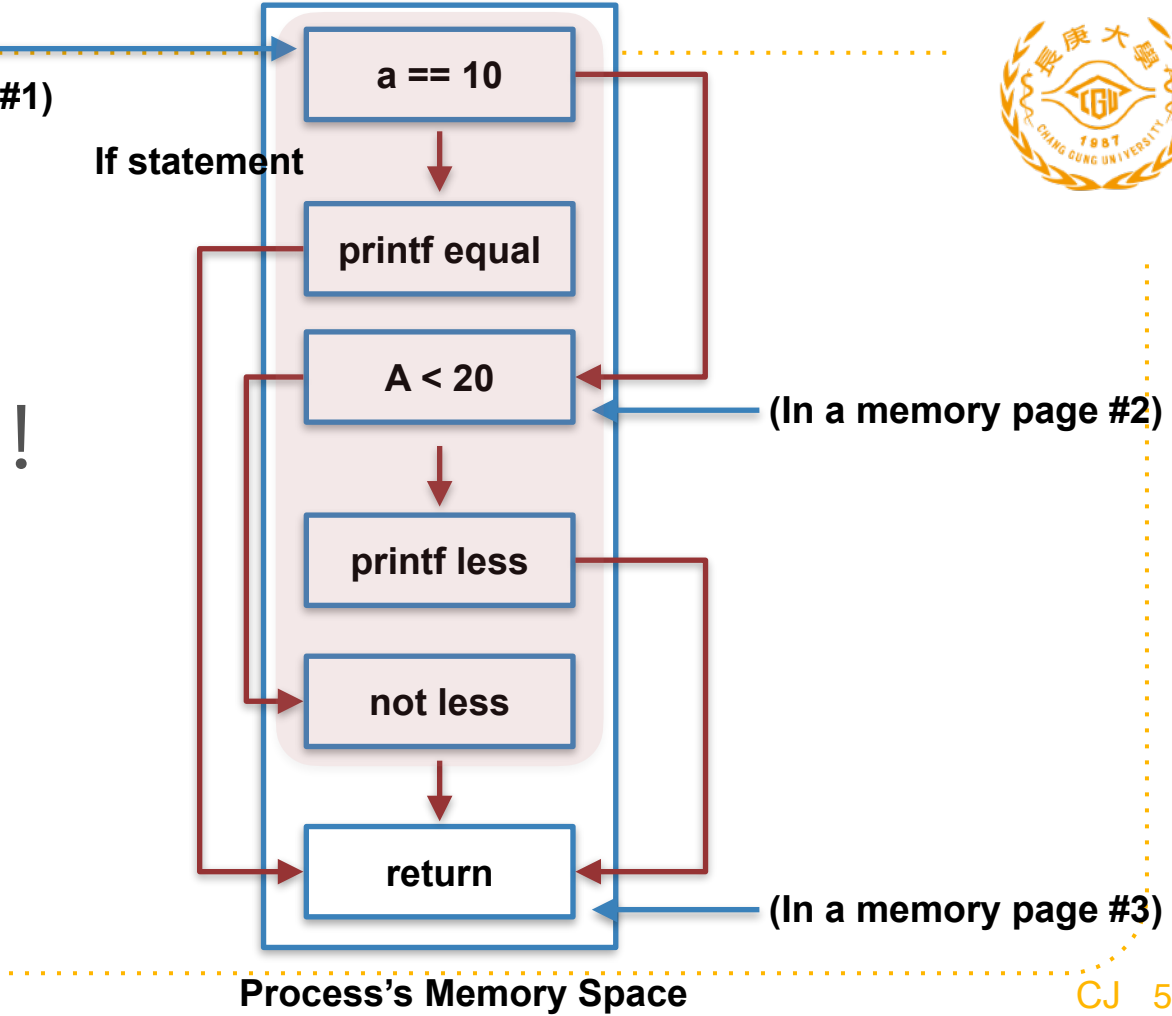
Process's Memory Space

```c
#include <stdio.h>
int main() {
    int a = 10;

    if (a == 10) {
        printf("a is equal to 10\n");
    } else if (a < 20) {
        printf("a is less than 10\n");
    } else {
        printf("a is not less than 20\n");
    }

    return 0;
}
```

```
a is equal to 10


...Program finished with exit code 0
Press ENTER to exit console.
```

判斷順序會影響你寫程式的速度！打亂你的節奏！

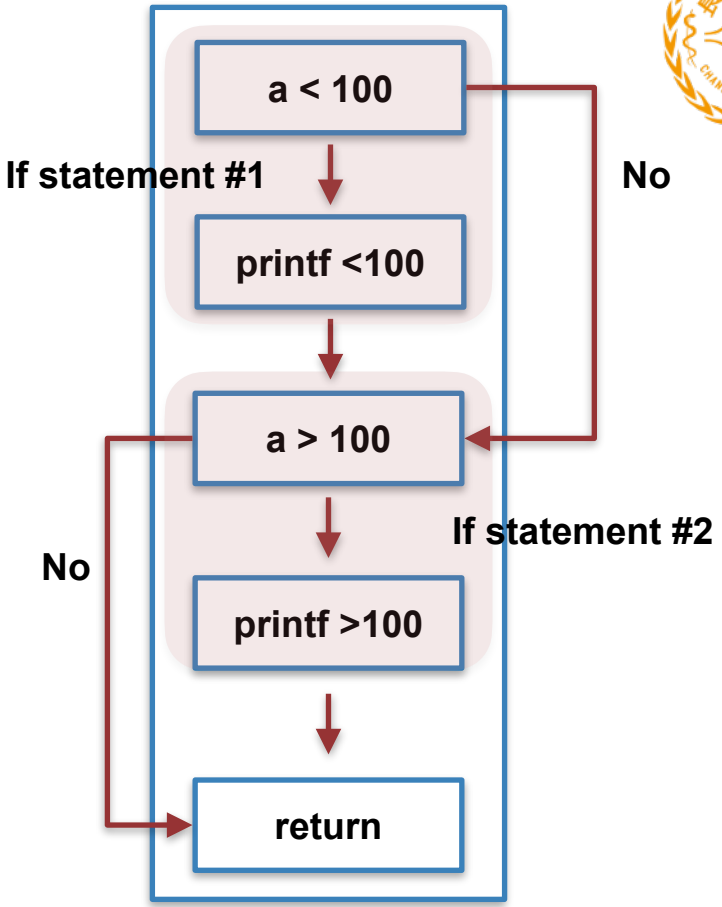
所以寫之前一定要在你腦中CPU跑過一次！



注意判斷順序！

常發生的放前面！

```c
#include <stdio.h>
int main() {
    int a = 100;

    if (a < 100) {
        printf("<100\n");
    }

    if (a > 100) {
        printf(">100\n");
    }

    return 0;
}
```



Process's Memory Space

nested if statements

```c
#include <stdio.h>
int main() {
    int a = 100;
    int b = 200;

    if (a == 100) {
        if (b == 200) {
            printf("Value of a is 100 and b is 200\n" );
        }
    }
    return 0;
}
```

input

```
Value of a is 100 and b is 200


...Program finished with exit code 0
Press ENTER to exit console.
```

盡量不要超過三層
之後會講怎麼避免！

請練習畫一下流程圖

```c
#include <stdio.h>
int main() {
    char grade = 'B';
    printf("Your grade is  %c\n", grade );
    switch (grade) {
    case 'A' :
        printf("Excellent!\n" );
        break;
    case 'B' :
    case 'C' :
        printf("Well done\n" );
        break;
    case 'D' :
    case 'F' :
        printf("Better try again\n" );
        break;
    default :
        printf("Invalid grade\n" );
    }
    return 0;
}
```

```
Your grade is  B
Well done


...Program finished with exit code 0
Press ENTER to exit console.
```

switch statement

判斷同type性的變數

有很多選項時候適合使用

和if else 一樣

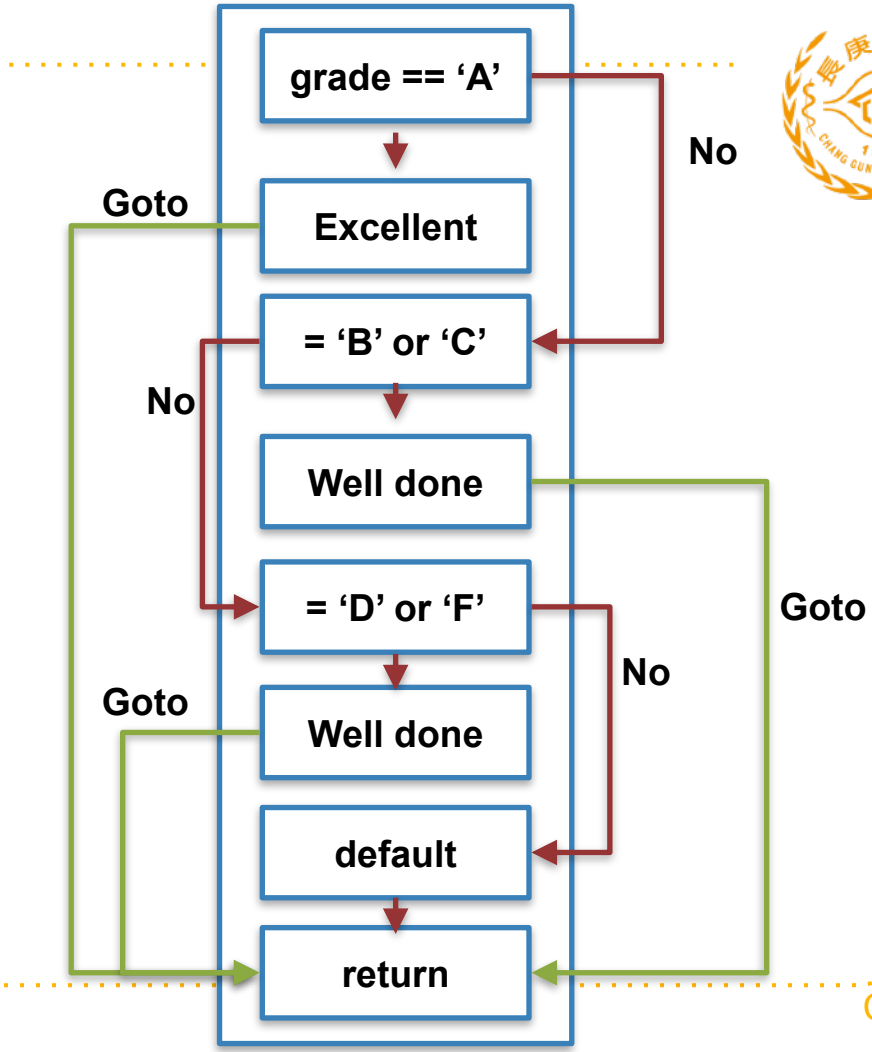
但是比較易讀

if else 不一定是判斷同type的變數

break

```c
#include <stdio.h>
int main() {
    char grade = 'B';
    printf("Your grade is  %c\n", grade );
    switch (grade) {
    case 'A' :
        printf("Excellent!\n" );
        break;
    case 'B' :
    case 'C' :
        printf("Well done\n" );
        break;
    case 'D' :
    case 'F' :
        printf("Better try again\n" );
        break;
    default :
        printf("Invalid grade\n" );
    }
    return 0;
}
```

```
Your grade is  B
Well done


...Program finished with exit code 0
Press ENTER to exit console.
```



nested switch statements

```c
#include <stdio.h>
int main() {
    int a = 100;
    int b = 200;

    switch (a) {
    case 100:
        printf("This is part of outer switch, a=%d\n", a);

        switch (b) {
        case 200:
            printf("This is part of inner switch, a=%d, b=%d\n", a, b);
        }
    }
    return 0;
}
```

```
input
This is part of outer switch, a=100
This is part of inner switch, a=100, b=200


...Program finished with exit code 0
Press ENTER to exit console.
```

兩層就已經很長了

Code Style !!!!!!

```c
#include <stdio.h>
int main(void) {
    char a = 'A';
    char i = 'A';
    printf("%d\n", i != a);
    printf("%d\n", i == a);
    printf("%s\n", i != a ? "true" : "false");
    printf("%s\n", i == a ? "true" : "false");
    return 0;
}
```

input

```
0
1
false
true
..
Press ENTER to exit console.
```

判斷相等三元運算子

If Condition is true ? then value X : otherwise value Y

```c
#include <stdio.h>
int main() {
    int n = 1;
label:
    printf("%d ",n);
    n++;
    if (n <= 10)
        goto label;
    return 0;
}
```

```
1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

用if 當loop

很少使用，不建議使用

破壞程式結構

不易讀，知道有這就好

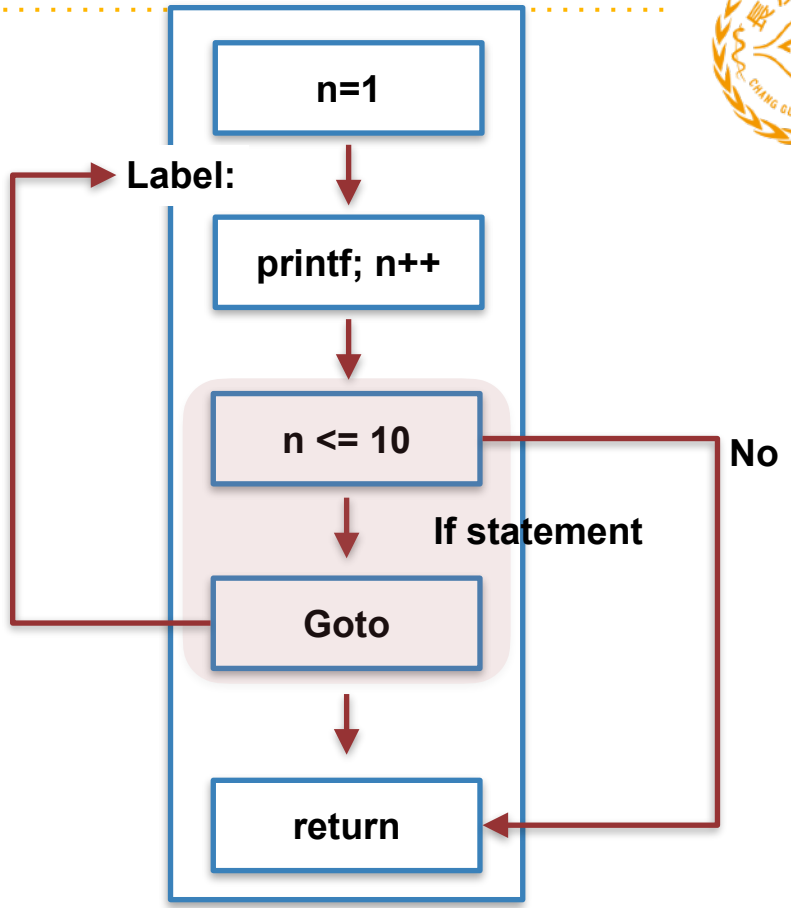
```c
#include <stdio.h>
int main() {
    int n = 1;
label:
    printf("%d ",n);
    n++;
    if (n <= 10)
        goto label;
    return 0;
}
```

```
1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

```c
if (n <= 10) {
    goto label;
}

// if (n <= 10) goto label;
```

```c
#include <stdio.h>
int main() {
    int n = 1;
label:
    printf("%d ",n);
    n++;
    if (n <= 10)
        goto label;
    return 0;
}
```

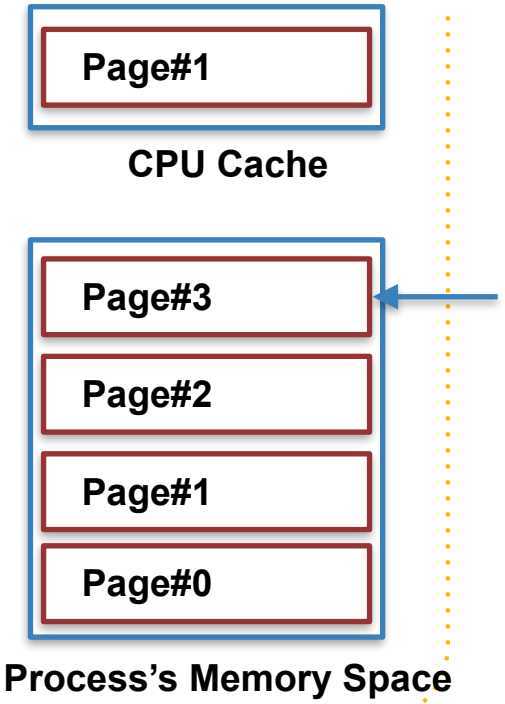
```
1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```



Process's Memory Space

If statement, control flow

- 有可能對CPU的效能造成影響
- **Caching**
  - 跳太遠了，指令不在cache裡
  - Spatial Locality, Temporal Locality
- Pipelining
  - Fetch, decode, execute, write back
  - If (a < 10) printf(“123”);



Conclusion

- C Basics
  - Variables
    - **Naming rules**
    - **Integer**, floating-point
    - Operations
      - **Bitwise operators**
      - **Priority of operators**
  - Flow of Control
    - **If else**
    - Switch

HW#6，12/21 12:00PM

Commit to your GitHub

- 給一個數字 n (> 0)
- 判斷他是不是 power of 2，請使用bitwise operators
- 1 is true
- 2 is true
- 3 is false
- 4 is true
- **main0.c**
- **flow0.pdf**

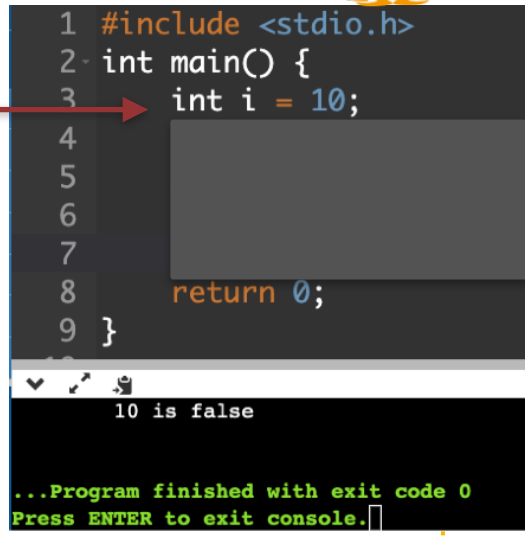
請把 i 預設為 10
但是也要考慮其他情況

```c
#include <stdio.h>
int main() {
    int i = 10;


    return 0;
}
```

```
10 is false

...Program finished with exit code 0
Press ENTER to exit console.
```

HW#6，12/21 12:00PM
Commit to your GitHub

請把 i 預設為 10
但是也要考慮其他情況



- 給一個數字 n (> 0)
- 如果 整除 3 就只能印出 “Love”
- 如果 整除 5 就只能印出 “IU”
- 如果 整除 15 就只能印出 “Love IU”
- 如果 都不整除 3, 5, 15 就只能印出 數值
- **main1.c**
- **flow1.pdf**

- **請注意！作業沒有輸入！**
- **如果程式需要輸入零分計算！**

HW#6，12/28 12:00PM
Commit to your GitHub

- 判斷一個數為偶數
- 如果是偶數請印出 even
- 如果是奇數請印出 odd
- **main2.c**
- **flow2.pdf**

請把 i 預設為 10
但是也要考慮其他情況

- **請注意！作業沒有輸入！**
- **如果程式需要輸入零分計算！**

```c
#include <stdio.h>
int main() {
    int i = 10;


    return 0;
}
```

HW#6，12/28 12:00PM
Commit to your GitHub

- 請注意！作業沒有輸入！
- 如果程式需要輸入零分計算！

- 判斷是否為閏年
- 西元年份可整除4，且不可整除100，為閏年
- 或是西元年份可整除400也為閏年
- **main3.c**
- **flow3.pdf**

請把 year 預設為 2022
但是也要考慮其他情況

```c
#include <stdio.h>
int main() {
    int year = 2022;

    return 0;
}
```

Thanks!

Open for any questions

CJ Wu

cjwu@mail.cgu.edu.tw