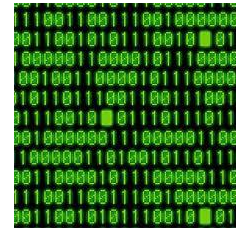


程式語言簡介

【程式語言分類】

- 機器語言 (Machine Language)：0 與 1
- 組合語言 (Assembly Language)：
用符號代替 0 與 1



▲ 機器語言

【C 語言】

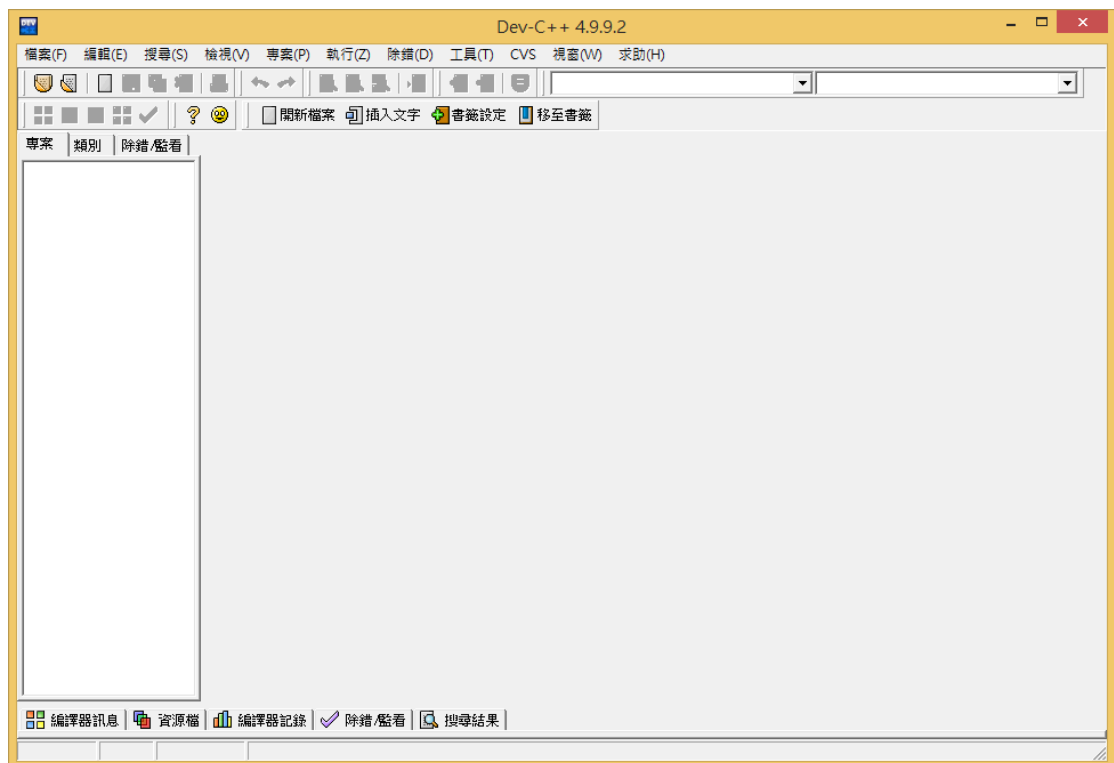
- 十分成熟的語言 (1972 A.D. 至今)
- 使用最廣泛的高階語言

【寫程式的流程】

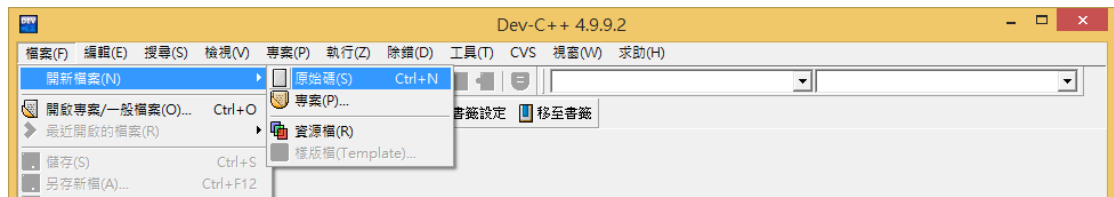
1. 打開「Dev-C++」軟體：



開啟後會看到這樣的介面：



- 點選「檔案」→「開新檔案」→「原始碼」來建立一個新的程式，
或者點選「開啟舊檔」來開啟現有的檔案。



- 打開檔案後，就會看到編輯區域。



- 在完成編寫後，點選「檔案」→「儲存」存檔。
- 若要執行程式，在存檔後點選「執行」→「編譯並執行」(快捷鍵 F9)。



Dev-C++ 會跳出一個「Compile Progress」視窗後，自動開始執行程式。

【語法解讀】

- 敘述 (Statement)：每一個動作稱為一個「敘述」。以分號 ; 結束。
- 註解 (Comment)：兩個斜線與之後的文字稱為註解。電腦不會執行註解的部份。

註解有兩種格式：

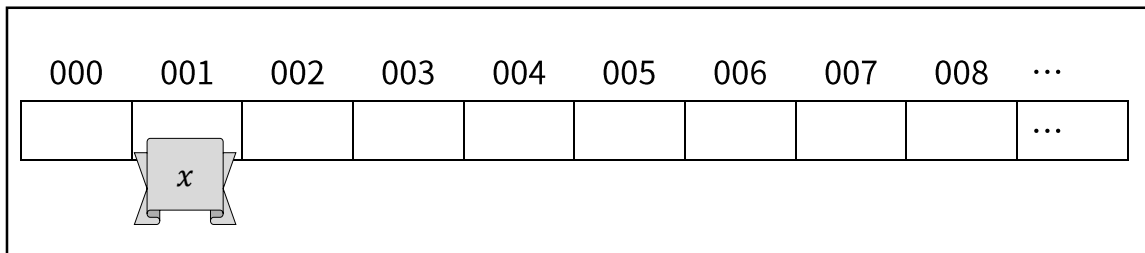
```
Statement;
/*Comment
  Comment */
Statement;
```

```
Statement; // Comment
Statement;
```

變數 Variable

變數是寫程式的時候，為了紀錄一些資料，所以在記憶體某個位子上放了一些資料，然後給他一個名字。

想像記憶體是一排櫃子，裡面有很多排列整齊的格子。格子都有編號（記憶體位址，Memory Address）。當我們創建一個變數叫做 x 的時候，就是我們在櫃子上指定了一格（確切是哪一格由電腦決定），然後把那格貼上 x 的名字：



【宣告 Declaration】

- 意思是指「創建一個新的變數」。
- 格式：變數型態 變數名稱;

Example

```
int score;
```

Explanation

`int` 是變數型態，`score` 是變數名稱。

【變數型態】

- 為了儲存不同型態的資料，C 語言定義了幾種不同的變數型態。
→ 就像不同種類的盒子裝不同的東西。
- 整數 `int`
Integer，用來存放整數。
例如：`-3`，`0`，`300`...
- 雙精度浮點數 `double`
用來存放實數。
例如：`3.14159`，`-2.1`，`10000`...

- 字元 char

Character，用來存放字元（一個字母或符號）。

例如： 'a' 、 '*' 、 'X' ...

【變數命名規則】

- 變數的名稱可以用大小寫英文字母、數字與符號組成。但不能用數字或符號當開頭。
- 大小寫敏感（Case Sensitive），亦即大寫與小寫會被視為不同的字。
- 不得使用保留字：意思是 C 語言本來會用到的一些東西不能拿來當作變數的名字，例如 int, char, printf, scanf, if, else, for, while, break, goto, continue 等。

Example

```
int ?score;          //Illegal
int score;           //Legal
int no_score;        //Legal
int Score;           //Legal, different from "score"
```

【賦值 Assignment】

- 把一個值存到指定的變數裡面。
- 格式： 目標變數 = 要存入的值或變數；

Example

```
int score;
score = 100;
int myScore = 60;
myScore = score;
```

Explanation

1. 第一行宣告了一個變數叫做 score。
2. 第二行讓 score 存入 100。
3. 第三行宣告了一個新的整數變數 myScore，同時讓他一開始等於 60。
4. 第四行把 score 的值存到 myScore 裡。這時候 myScore 等於 100。

運算 (Operation)

對變數而言，除了 Assignment 之外，我們可以做一些其他運算（如加減乘除）或判斷（比大小、相等）。

【算術運算子 Arithmetic Operator】

- 各種算術運算子：

運算子	意義	範例	範例說明	結果
+	加法	2+6	計算 2+6	8
-	減法	10-7	計算 10-7	3
*	乘法	8*4	計算 8×4	32
/	除法	60/5	計算 60÷5	12
%	模除（取餘數）	21%9	計算 21÷9 的餘數	3

- 注意資料型態：除法中，整數/整數所得的結果還是整數。

Example （假設下面的變數型態都是 int）

```
a = 3+8;  
c = 20%7;  
b = a-100;  
d = c*a;  
e = d/10;  
e = e+2;
```

Explanation

計算的結果會是：

a = 11 c = 6 b = 11 - 100 = -89 d = 6 × 11 = 66

e 在倒數第二行會等於 6。

在最後一行中，我們讓 e = e+2，意思是 e 會變成原本 e 的值再加 2，即

6 + 2 = 8

要讓自己往上增加可以使用這樣的寫法。但是如果只要讓自己往上加 1，有個更快的 Operator 可以使用：

- 連加/連減運算子

運算子	意義	範例	範例說明
++	自身加 1	a++	讓 a 等於原本的 a 再加 1
--	自身減 1	b--	讓 b 等於原本的 b 再減 1

Example (假設下面的變數型態都是 int)

```
a = 5;
a++;
b = 9;
b--;
```

Explanation

執行到最後時，a=6，b=8。

【邏輯運算子 Logical Operator】

- 各種邏輯運算子：

運算子	意義	範例	範例說明	結果
>	判斷大於	5>2	判斷 5 是否大於 2	True
<	判斷小於	3<1	判斷 3 是否小於 1	False
>=	判斷大於等於	7>=7	判斷 7 是否大於等於 7	True
<=	判斷小於等於	6<=10	判斷 6 是否小於等於 10	True
==	判斷等於	1==10	判斷 1 是否等於 10	False
!=	判斷不等於	5!=5	判斷 5 是否不等於 5	False
&&	AND (且)	a&& b	判斷 a AND b (見下方範例)	
	OR (或)	a b	判斷 a OR b (見下方範例)	

- 布林運算：AND 且

假設兩個變數或判斷式 a 與 b，當兩者均為 1 (True) 時，`a && b` 等於 1；否則等於 0 (False)。

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

Example

`(1>0) && (3>0)` : True

`(1>0) && (3<0)` : False

- 布林運算：OR 或

假設兩個變數或判斷式 a 與 b，只要其中一個為 1，`a || b` 就等於 1。

a	b	a b
0	0	0
0	1	1
1	0	1
1	1	1

Example

`(1>0) || (3>0)` : True

`(1>0) || (3<0)` : True

`(1<0) || (3<0)` : False

基本輸入 Output

為了讓程式能夠顯示或從鍵盤讀入東西，我們需要一些函數來進行標準輸入輸出（Standard Input/Output）的動作。最常使用的包含 `printf`（用來在螢幕上顯示文字或數字）以及 `scanf`（從鍵盤讀入資料）。

【 printf 】

- 印出字串

格式：`printf("字串");`

Example Demo

```
printf("NTHUEE Rocks!");
```

Result

```
NTHUEE Rocks!
```

- 雙引號

代表括起來的這個部份是一個字串。印出來的東西就會是雙引號裡面的東西，但是不會包含雙引號。

- 印出變數

格式：`printf("格式字串", 變數項目...);`

Example Demo

```
int level = 19;  
printf("We are grade %d", level);
```

Result

```
We are grade 19
```

- 格式字串

`%d` 代表這個地方會以整數的形式，把一個變數的值印出來。

在雙引號外面，以逗號分隔，依序列出要在 `%d` 位置印出的變數。

除了 `%d` 代表印出整數之外，還有其他各種不同的格式字串可以使用：

格式字串	代表意義
<code>%d</code>	十進位整數
<code>%c</code>	字元
<code>%lf</code>	浮點數

Example Demo

```
int year = 2015;
int month = 8;
int day = 20;
printf("Today is %d/%d/%d.", year, month, day);
```

Result

Today is 2015/8/20

● 印出特殊符號（跳脫字元 Escape Character）

既然 printf 裡面需要用到雙引號 `""`，那如果要顯示雙引號的話電腦不就搞混了？因此，當我們需要印出一些特殊字元的時候，我們需要使用特殊的寫法：

跳脫字元	功能
<code>\n</code>	換行
<code>\"</code>	印出雙引號
<code>\'</code>	印出單引號
<code>\\</code>	印出反斜線 <code>\</code>

Example Demo

```
printf("We are at \"NTHUEEEECamp\", ");
printf("we are having a lot of fun!\n");
printf("NTHUEE Rocks!");
```

Result

We are at "NTHUEEEECamp", we are having a lot of fun!
NTHUEE Rocks!

【 scanf 】

● 格式：`scanf("格式字串", &目標變數...);`

Example Demo

```
int level;
scanf("%d", &level);
printf("We are grade %d", level);
```

Result （斜體字為輸入的值）

5
We are grade 5

- 格式字串

%d 代表在這裡輸入一個變數，然後以整數的形式存起來。

在雙引號外面，以逗號分隔，依序列出要存進去的目標變數。

【 Group Demo 1 】

Question

寫出一個加法器，讓使用者輸入兩個數，然後顯示兩個數的和。

Code Demo

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

int main(void) {

    int augend, addend, sum;
    scanf("%d", &augend);
    scanf("%d", &addend);
    sum = augend + addend;
    printf("%d", sum);

    return 0;
}
```

這幾行打就對了！
不要問，你會怕。

Explanation

- | | |
|---------------------|----------------|
| 1. 宣告三個變數：被加數、加數、和 | → 【變數的宣告】 |
| 2. 讓使用者輸入被加數 | → 【scanf】 |
| 3. 讓使用者輸入加數 | → 【scanf】 |
| 4. 使「和」等於「被加數」+「加數」 | → 【算術運算子】、【賦值】 |
| 5. 顯示結果 | → 【printf】 |

選擇性結構 Selection

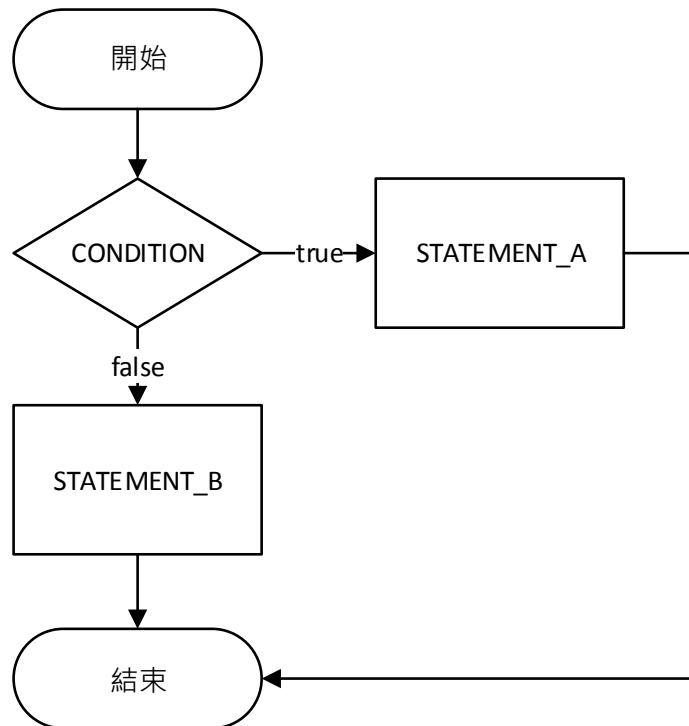
當我們需要依照不同條件做不同的事情時，我們會使用判斷式或選擇性結構來做選擇。而其中一種選擇性結構叫做 if-else 敘述。

【 if-else 】

- 格式 (else 可以不寫)：

```
if (CONDITION) {  
    STATEMENT_A;  
}  
else {  
    STATEMENT_B;  
}
```

- 流程圖：



Pseudo Example

如果今天沒有下雨，那就去打球；反之留在宿舍耍宅打 LOL。

```
if (沒有下雨) {  
    出去打球;  
}  
else {  
    留在宿舍耍宅打 LOL;  
}
```

Example Demo：正數判斷

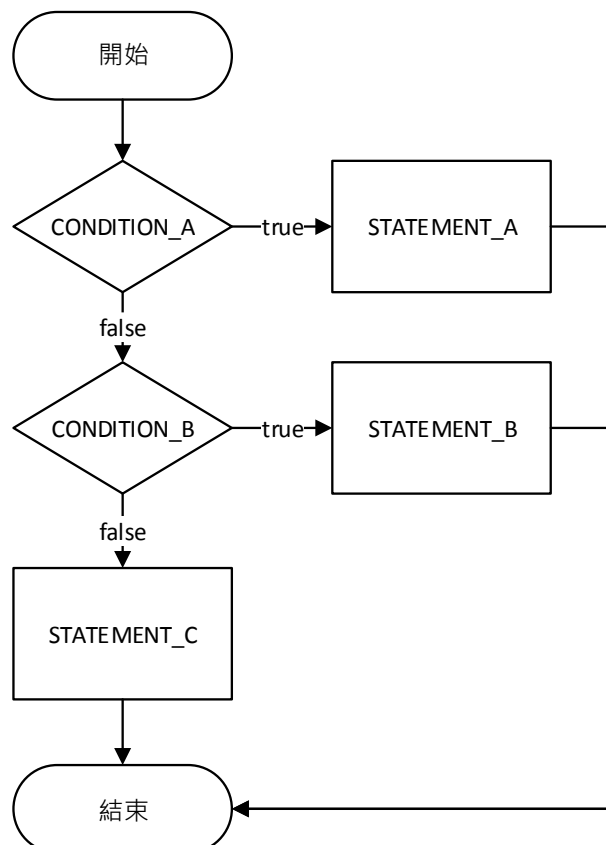
```
int a;
scanf("%d", &a);
if(a>0){
    printf("%d is bigger than 0.\n", a);
}
else{
    printf("%d is smaller or equal to 0.\n", a);
}
```

【 else if 】

- 當條件多於一個時，可以在 if 與 else 之間以 else if 多加其他條件。

```
if(CONDITION_A){
    STATEMENT_A; (Execute when CONDITION_A is TRUE)
}
else if(CONDITION_B){
    STATEMENT_B; (Execute when CONDITION_B is TRUE)
}
else{
    STATEMENT_C; (Execute when all FALSE)
}
```

- 流程圖：



- 從流程圖中可以發現在同一個 if-else if-else 的 statement 中，當寫在前面的條件符合時，後面的條件就不會判斷了。

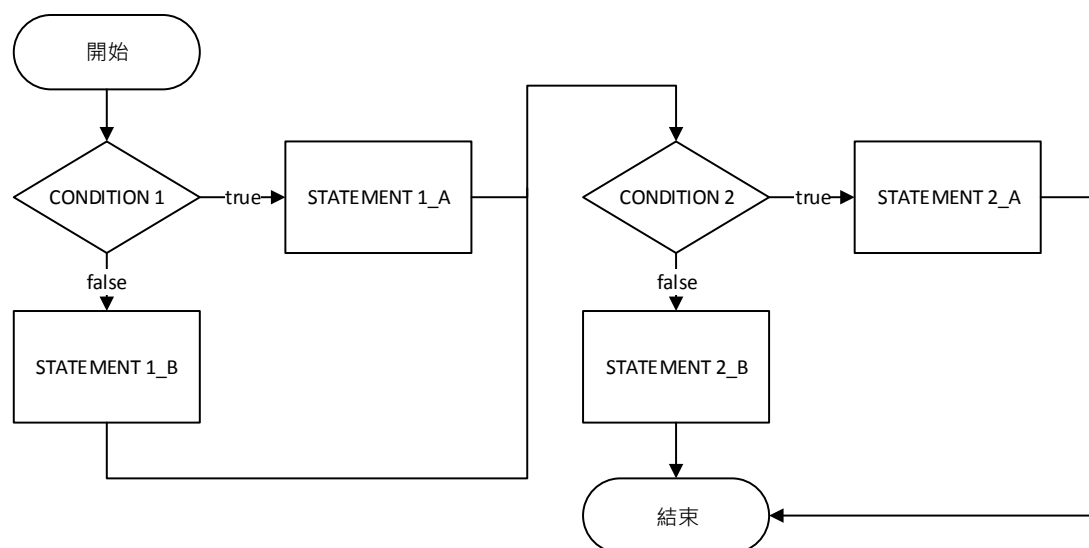
Example Demo：三一律判斷器

```
int a, b;
scanf("%d %d", &a, &b);
if(a>b){
    printf("%d is bigger than %d.\n", a, b);
}
else if(a<b){
    printf("%d is smaller than %d.\n", a, b);
}
else{
    printf("%d is equal to %d.\n", a, b);
}
```

【 各種不同的判斷組合 】

- if; else; if; else;

```
if(CONDITION 1){
    STATEMENT 1_A;
}
else{
    STATEMENT 1_B;
}
if(CONDITION 2){
    STATEMENT 2_A;
}
else{
    STATEMENT 2_B;
}
```

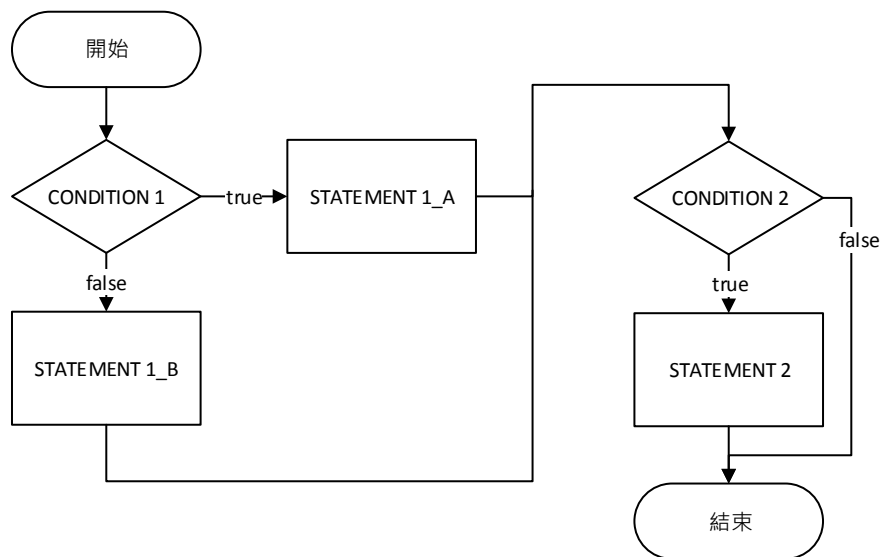


- if; else; if;

```

if (CONDITION 1) {
    STATEMENT 1_A;
}
else {
    STATEMENT 1_B;
}
if (CONDITION 2) {
    STATEMENT 2;
}

```

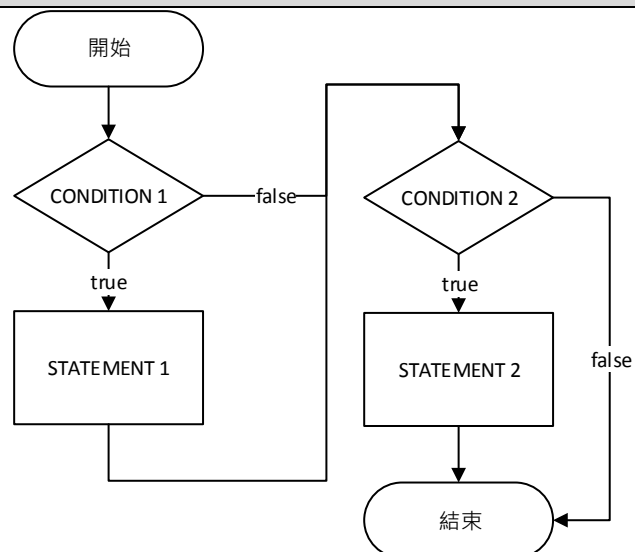


- if; if;

```

if (CONDITION 1) {
    STATEMENT 1;
}
if (CONDITION 2) {
    STATEMENT 2;
}

```



【 Group Demo 2 】

Question

寫出一個 7 的倍數判斷程式，讓使用者輸入一個數，然後顯示該數是否為 7 的倍數。

Code Demo

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

int main(void){

    int in;
    scanf("%d", &in);
    if((in%7)==0){
        printf("YES");
    }
    else{
        printf("NO");
    }

    return 0;
}
```

Steps

1. 宣告一個變數：輸入的數字 → 【變數的宣告】
2. 讓使用者輸入要判斷的數字 → 【scanf】
3. 判斷是否為 7 的倍數 → 【if-else】、【運算子】
4. 顯示結果（“YES” 或 “NO”） → 【printf】

Notice

* 7 的倍數 → 除以 7 餘數是 0 → 用取餘數運算子來判斷

重複性結構 Iteration

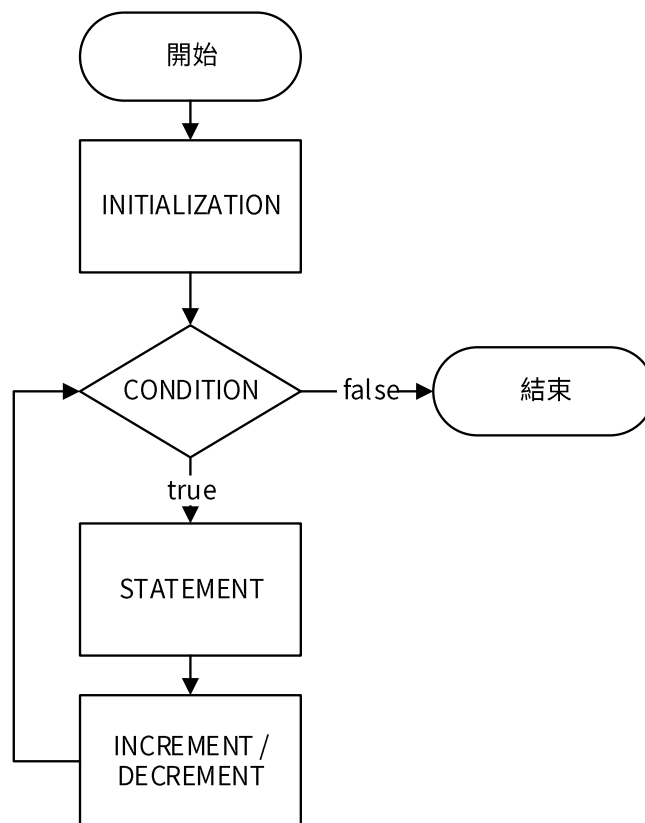
除了選擇之外，我們也會需要電腦重複做一些工作。這時候就需要重複性結構的幫忙，有時候又叫做「迴圈」(Loop)。在這裡我們將介紹 for 迴圈。

【 for loop 】

- 格式：

```
for (INITIALIZATION; CONDITION; INCREMENT/DECREMENT) {  
    STATEMENT;  
}
```

- 流程圖：



Pseudo Example

12 月時，企鵝從 1 日開始，每天的工作就是

吃飯 (Eat)、睡覺 (Sleep)、打東東 (HitDongDong)

每天會把日期加一，直到平安夜為止。因此企鵝的工作可以寫成：

```
int Date;  
for(Date = 1; Date < 25; Date++){  
    Eat;  
    Sleep;  
    HitDongDong;  
}
```


Example Demo : 1+2+...+10

```
int a, sum=0;
for(a=1; a<=10; a++){
    sum = sum + a;
    printf("%d\n", sum);
}
```

Result

55

- 注意條件要寫好：如果繼續執行的判斷條件永遠不會違反，會形成無窮迴圈（Infinite Loop）永遠跳不出來，直到程式被關閉為止。

Example Demo

```
int a, sum=0;
for(a=1; a>=1; a++){
    sum = sum + a;
    printf("%d\n", sum);
}
```

【 Group Demo 3 】

Question

列出 100~999 間所有的阿姆斯壯數（每個位數的 3 次方的總和等於該數本身）。 Ex. $153 = 1^3 + 5^3 + 3^3$

Code Demo

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

int main(void){

    int counter, units, tens, hundreds, sum;
    for(counter = 100; counter <=999; counter++){
        hundreds = counter/100;
        tens = (counter/10)%10;
        units = counter%10;
        sum = hundreds*hundreds*hundreds +
            tens*tens*tens +
            units*units*units;
        if(sum == counter){
            printf("%d\n", counter);
        }
    }

    return 0;
}
```

Steps

1. 宣告四個變數：上數數、百/十/個位數 → **【變數的宣告】**
2. 逐一計算 100~999 的數 → **【for】**
3. 計算出百位/十位/個位數與其三次方的和 → **【算術運算子】**
4. 判斷是否與原本的數字一樣 → **【if-else】**

【巢狀迴圈】

- 巢狀：意思是一個迴圈裡面出現另外一個迴圈。

Pseudo Example

假設一個大學生從禮拜一到禮拜五，每天的行程是：

起床 (GetUp)、上課、耍廢 (VegOut)、吃飯 (Eat)、睡覺 (Sleep)

而在第一堂到第八堂「上課」的時候，過程是：

打開課本 (OpenBook)、睡覺 (Sleep)、下課 (TakeBreak)

因此我們可以寫：

```
int WeekDay, class;
for(WeekDay = 1; WeekDay <= 5; WeekDay++){
    GetUp;
    for(class = 1; class <= 8; class++){
        OpenBook;
        Sleep;
        TakeBreak;
    }
    VegOut;
    Eat;
    Sleep;
}
```

Example Demo：用星號印出 20x3 的矩形

```
int w, h;
for(h=1; h<=3; h++){
    for(w=1; w<=20; w++){
        printf("*");
    }
    printf("\n");
}
```

Result

```
*****
*****
*****
```

【 Group Demo 4 】

Question

印出九九乘法表。

Code Demo

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

int main(){
    int a, b;
    for(a=1; a<=9; a=a+1){
        for(b=1; b<=9; b=b+1){
            printf("%d*%d=%d ", a, b, a*b);
        }
        printf("\n");
    }
    return 0;
}
```

Steps

1. 宣告兩個變數：乘數、被乘數 → 【變數的宣告】、【scanf】
2. 改變乘數 → 【for】、【printf】
3. 對每一個乘數改變被乘數 → 【for】、【printf】

Hint

要先從 $1*1 \rightarrow 1*2 \rightarrow \dots \rightarrow 1*9$ ，然後

$2*1 \rightarrow 2*2 \rightarrow \dots$ 一直到 $9*9$

試著找找看規則？