C/C++ 流程控制語法

if (*expression*) *statement* [else *statement*]

while (*expression*) *statement*

do *statement* while (*expression*);

for ([*initialization*]; *expression*; [*increment*]) *statement*

switch (*expression*) {

{case *label*:

statements

[break;]}

[default:

statements]

}

[…] 可省略

{…} 可重複

statement 可以是：

expression;

或

{{*statement*}}

C/C++ 流程控制實例

```
int cnt = 0;
while (cnt < 100) {
    if (! cnt) cout << ", ";
    cnt ++;
    cout << cnt;
}
cout << "\n";

int r = 1333;
for (int wc = 0; wc < 100; wc ++) if (wc > r) {
    cout << wc – 1 << "^2 <= 1333 < "
        << wc << "^2\n";
    break;
}
```

```
int resp;
cin >> resp;
switch (resp) {
case 1:
case 2:
    cout << "that's less than 3\n";
    break;
case 3:
    cout << "good choice!\n";
default:
    cout << "you entered " << resp << "\n";
}
```

變數範圍

```
#include <iostream>
using namespace std;
int ss = 123;
void set_ss(int t) {ss = t;}
int main()
{
    cout << "ss=" << ss << '\n';
    int i = 70, s = 0, ss = 0;
    for (int i = 0; i < 10; i ++) {
        int t = i;
        static int tt = i;
        s += t;
        ss += tt;
    }
    cout << "i=" << i << '\n';
    cout << "ss=" << ss << '\n';
    set_ss(321);
    cout << "::ss=" << ::ss << '\n';
    while (int s = i -= 35) cout << s << '\n';
    cout << "s=" << s << '\n';
    return 0;
}
```

global variable

local variables

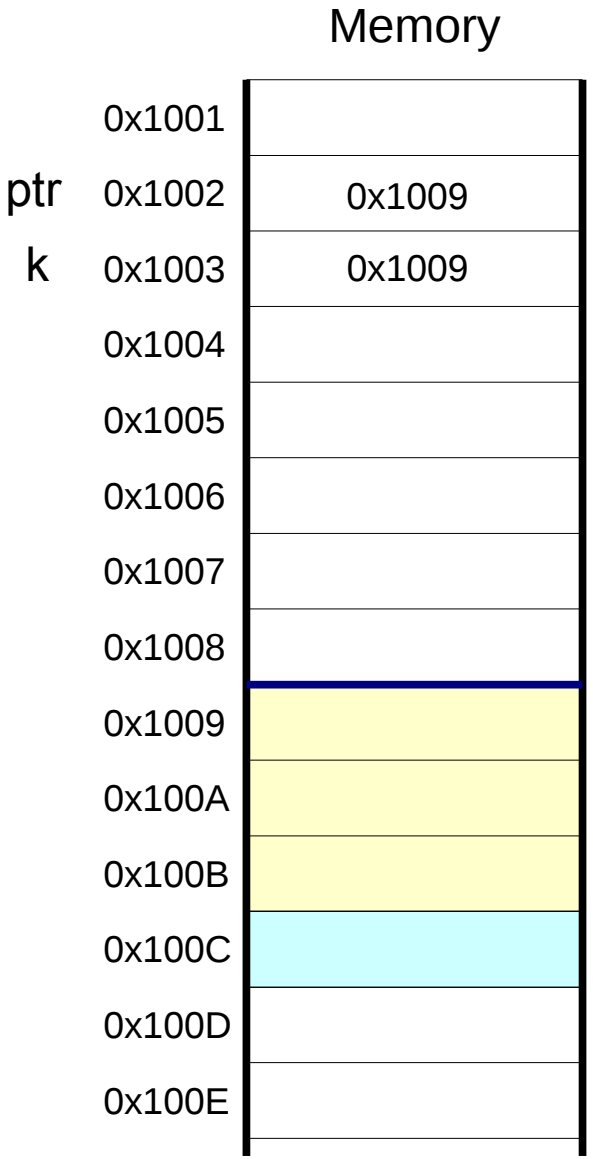
static variable

declaration as condition

Output:

```
ss=123
i=70
ss=0
::ss=321
35
s=45
```

記憶空間管理




```
#include <iostream>
using namespace std;
int main()
{
        int * ptr = new int [3];
        int * k = ptr;
        ptr[1] = 30;
        * ptr = * ++ptr - 40;
        ptr--[1] = 20;
        k[0] = 21;
        cout << '{' << ptr[0] << ',' << ptr[1]
             << ',' << ptr[2] << "}\n";
        ptr = new int;
        * ptr = 7;
        delete [] k;
        delete ptr;
        return 0;
}
```

```
Output:
{21,-10,20}
```

複合變數 (結構)

```
#include <iostream>
using namespace std;

enum PetType
{
    PT_DOG = 1,
    PT_CAT,
    PT_BIRD,
    PT_FISH
};

struct Pet
{
    string name;
    PetType type;
    unsigned age;
    double cost;
};

void call_pet(Pet & pet)
{
    cout << "Come here, ";
    if (pet.age > 5) cout << "old ";
    if (pet.age <= 2) cout << "little ";
    cout << pet.name << "!\n";
}
int main()
{
    Pet fred;
    fred.name = "Fred";
    fred.type = PT_DOG;
    fred.age = 7;
    fred.cost = 1230.0;

    call_pet(fred);
    return 0;
}
```

類別 (class)

```
class Pet
{
    string name;
    PetType type;
    unsigned age;
    double cost;
public:
    Pet(string n, PetType t, unsigned a, double c) :
        name(n), type(t), age(a), cost(c)
    {}
    void call()
    {
        cout << "Come here, ";
        if (age > 5) cout << "old ";
        if (age < 2) cout << "little ";
        cout << name << "!\n";
    }
};
```

```
int main()
{
    Pet fred("Fred", PT_DOG, 7, 1230.0);
    fred.call();
    return 0;
}
```

字元排列

```
#include <iostream>
using namespace std;
int main()
{
	int size = 4;
	char c[] = "abcd\n";
	int n[4];

	int idx = size;
	char t;
lab2:
	for (int i = 0; i < idx; i ++) n[i] = i + 1;
	cout << c << '\n';
lab3:
	idx = 0;
	t = c[0];
lab1:
	if (n[idx]) goto lab4;
	idx ++;
	if (idx == size) goto lab9;
	c[0] = c[idx];
	c[idx] = t;
	t = c[0];
	goto lab1;
lab4:
	n[idx] --;
	if (n[idx] == 0) goto lab3;
	goto lab2;
lab9:
	return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
	int size = 4;
	char c[] = "abcd\n";
	int n[size + 1];
	n[size + 1] = 1;
	int idx = 4;
	do {
		if (n[idx]) {
			for (int i = 0; i < idx; i ++) n [i] = i + 1;
			cout << c;
		}
		idx = 0;
		char t = c[0];
		while (n[idx] == 0) {
			idx ++;
			c[0] = c[idx];
			c[idx] = t;
			t = c[0];
		}
		n[idx] --;
	} while (idx < size);
	return 0;
}
```

結構化程式

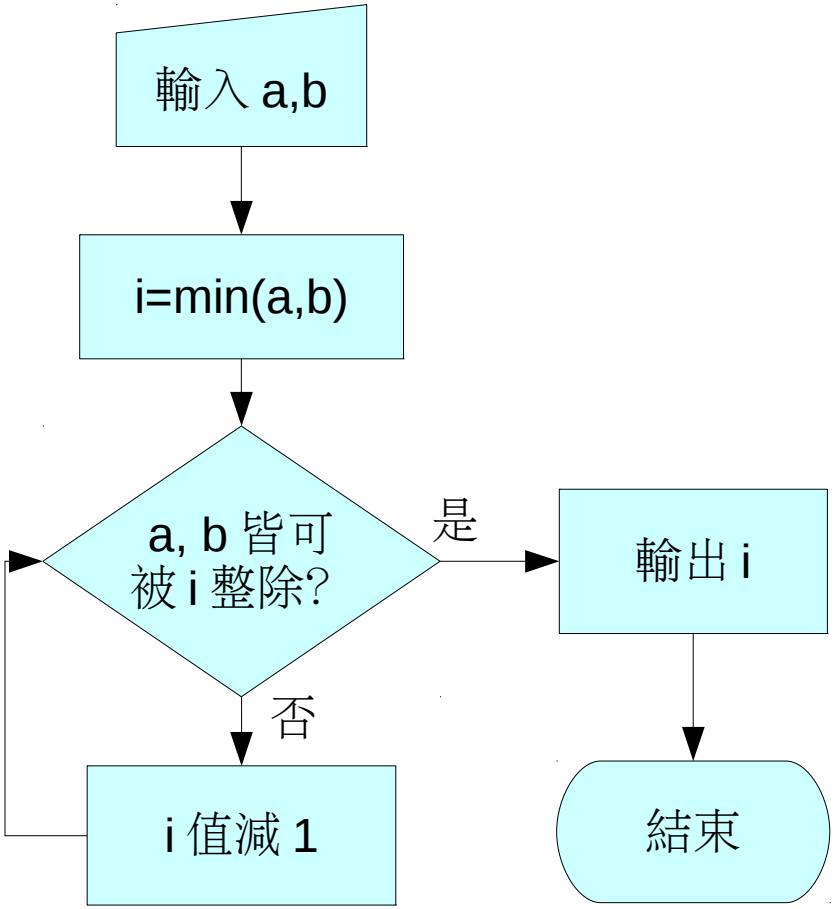
```
#include <iostream>
using namespace std;
char c[] = "abcd\n";
void rotate(char * p, int l)
{
	char ch = p[l - 1];
	for (int i = l - 1; i; i --) p[i] = p[i - 1];
	p[0] = ch;
}
void perm(char * p, int l)
{
	if (l == 1) cout << c;
	else for (int i = 0; i < l; i ++) {
		perm(p, l - 1);
		rotate(p, l);
	}
}
int main()
{
	perm(c, 4);
	return 0;
}
```

Output

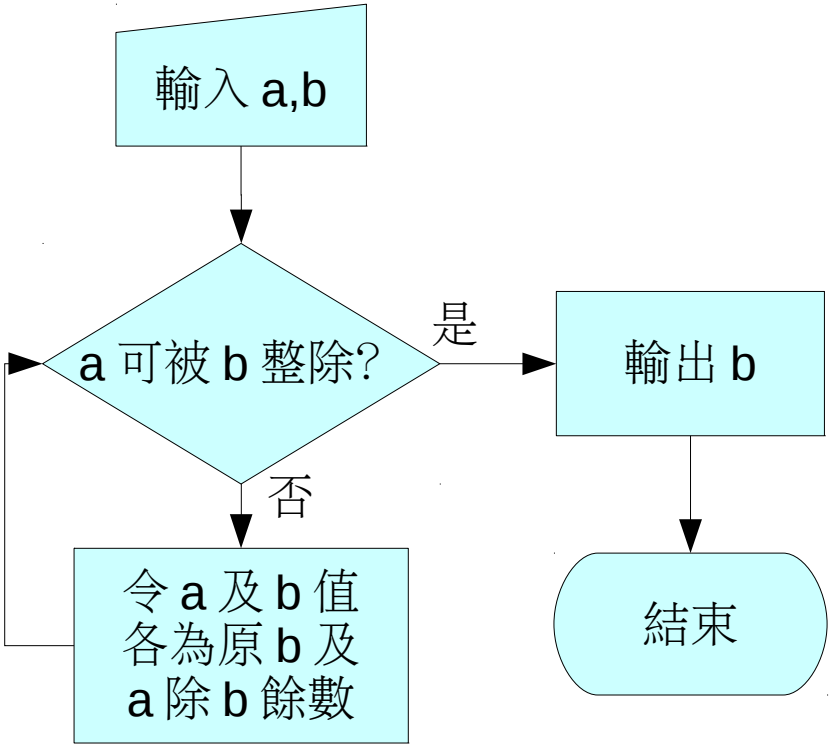
```
abcd
bacd
cabd
acbd
bcad
cbad
dabc
adbc
bdac
dbac
abdc
badc
cdab
dcab
acdb
cadb
dacb
adcb
bcda
cbda
dbca
bdca
cdba
dcba
```

演算法：最大公因數

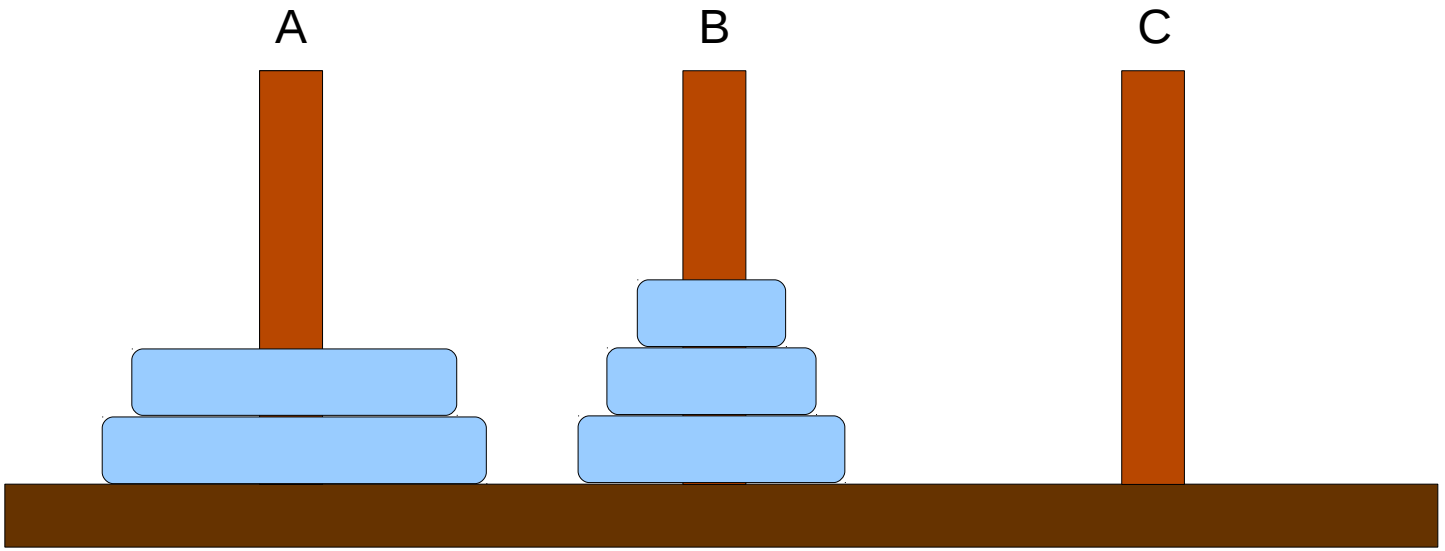
暴力法



輾轉相除法



遞迴法：河内塔問題



A→C
B→C
B→A
C→A
B→C
…

檔案輸出入

```
#include <fstream>
using namespace std;
int main()
{
	ifstream f_in("mat.txt");
	int msz;
	f_in >> msz;
	double arr[msz][msz];
	for (int i = 0; i < msz; i ++)
	for (int j = 0; j < msz; j ++) {
		f_in >> arr[i][j];
	}
	ofstream f_out("matt.txt");
	f_out << msz << '\n';
	for (int i = 0; i < msz; i ++) {
		if (i) f_out << '\n';
		for (int j = 0; j < msz; j ++) {
			if (j) f_out << '\t';
			f_out << arr[j][i];
		}
	}
	f_out << '\n';
	return 0;
}
```

```
cp1@area:~$ cat mat.txt
3
123.0	42.4	54.3
31.11	2.19	3.2
1.56	32.2	112.3
cp1@area:~$ ./a.out
cp1@area:~$ cat matt.txt
3
123	31.11	1.56
42.4	2.19	32.2
54.3	3.2	112.3
cp1@area:~$
```

本週程式作業

1.輸入兩數並輸出其最小公倍數。

2.輸入 n 個名稱並輸出其所有 2^n 個組合。

3.繪出你在 2. 所用的流程圖。（手繪的請寄掃描檔）

4.實作河內塔問題的遞迴解。

[隨意題] 以非遞迴方式解出河內塔問題

5.線性代數：由檔案輸入所有係數 a 及 c，輸出 x 的解（假定 det(a) 不為零）。

a11 x1+a12 x2+a13 x3 = c1

a21 x1+a22 x2+a23 x3 = c2

a31 x1+a32 x2+a33 x3 = c3

*** 請將程式檔以附件寄出。

程式執行範例

```
cp1@area:~$ hw4-1
input a b: 144 88
the LCM of 144 and 88 is 1584
cp1@area:~$ hw4-2
How many items? 3
item 1: apple
item 2: banana
item 3: cherry
[ ]
[cherry]
[banana]
[banana, cherry]
[apple]
[apple, cherry]
[apple, banana]
[apple, banana, cherry]
cp1@area:~$
```

```
cp1@area:~$ hw4-4
Moving from A to C
How many disks? 4
A->B
A->C
B->C
A->B
C->A
C->B
A->B
A->C
B->C
B->A
C->A
B->C
A->B
A->C
B->C
cp1@area:~$
```

程式執行範例

```
cp1@area:~$ cat input.txt
1    1    0
3    2    5
1    0    3

4    1    2
cp1@area:~$ hw4-5
(x1,x2,x3)=(8,-3.5,6.5)
cp1@area:~$
```

程式執行範例

```
cp1@area:~$ hw4-1
input a b: 144 88
the LCM of 144 and 88 is 1584
cp1@area:~$ hw4-2
How many items? 3
item 1: apple
item 2: banana
item 3: cherry
[ ]
[cherry]
[banana]
[banana, cherry]
[apple]
[apple, cherry]
[apple, banana]
[apple, banana, cherry]
cp1@area:~$
```

```
cp1@area:~$ hw4-4
Moving from A to C
How many disks? 4
A->B
A->C
B->C
A->B
C->A
C->B
A->B
A->C
B->C
B->A
C->A
B->C
A->B
A->C
B->C
cp1@area:~$
```