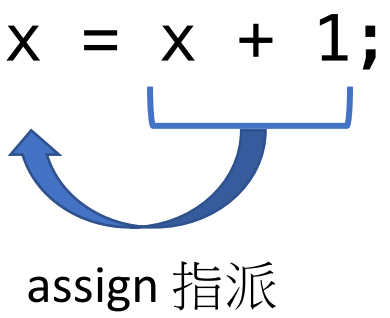
C++基礎語法 Unit-4

- 遞增遞減運算子
- 迴圈 (for)
- 一維陣列

遞增 / 遞減 運算子

遞增 (++) / 遞減 (--) 運算子

運算子	用途	寫法 1	寫法 2	寫法 3
++	遞增運算子	int x = 2; x = x + 1;	int x = 2; x += 1;	int x = 2; x++;
--	遞減運算子	int x = 2; x = x - 1;	int x = 2; x -= 1;	int x = 2; x--;



assign 指派

for 迴圈

for 迴圈

Example 4-1

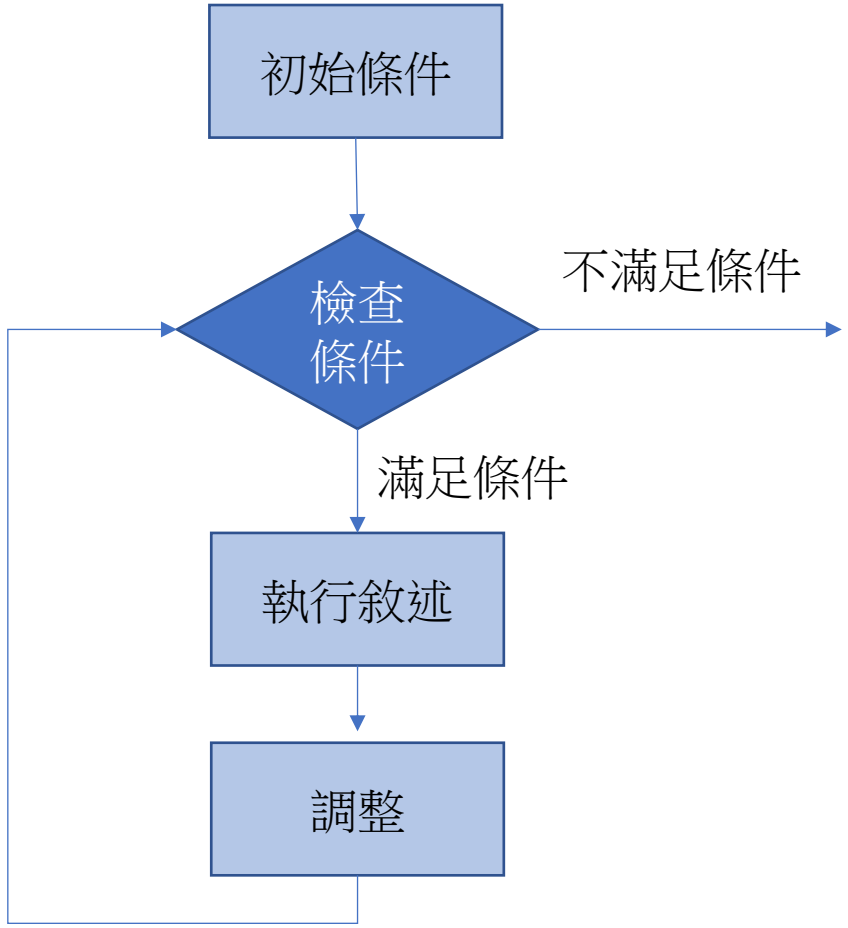
```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int N;
6      cin >> N;
7      for (int i = 0; i < N; i++) {
8          cout << i << "\n";
9      }
10     //cout << "i = " << i << "\n";
11     return 0;
12 }
```

初始條件　檢查條件　調整

```
for (int i = 0; i < N; i++) {
    //執行敘述
}
```

- 用 ; 區隔
- 迴圈的起始條件
- 繼續執行迴圈的條件（否則終止）
- 每一圈的遞增/遞減值
- 注意計數器 i 的生命週期

for 迴圈




```
i = i + 1
i += 1
i++
```

初始條件　檢查條件　調整

```
for (int i = 0; i < N; i++) {
    //執行敘述
}
```

- 用;區隔
- 迴圈的起始條件
- 繼續執行迴圈的條件（否則終止）
- 每一圈的遞增/遞減值
- 注意計數器 i 的生命週期

【練習】for 迴圈

Example 4-2

【Input】輸入一個正整數 N
【Output】輸出小於 N 的所有正奇數

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

int main() {
    int N;
    cin >> N;
    for (int i = 1; i < N;        ) {
        cout << i << "\n";
    }
    return 0;
}
```

https://pastebin.ubuntu.com/p/4wrtHN69bk/

【範例】for 迴圈

【Input】輸入一個正整數 N

【Output】判斷 N 是否為質數

Example 4-8

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

int main() {
    int N;
    cin >> N;
    for (int i = 2; i * i <= N; i++) {
        if (N % i == 0) {
            cout << "合數\n";
            return 0;
        }
    }
    cout << "質數\n";
    return 0;
}
```

Why?

https://pastebin.ubuntu.com/p/CtnHJC6GDM/

用檔案示範連續多筆測資的感覺（ZeroJudge d069）

uDebug.cpp

```cpp
#include <iostream>
using namespace std;
//#define JUDGE

int main() {
    #ifndef JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    #endif
    //================================

    int t;
    cin >> t;

    for (int i = 0; i < t; i++) {
        int y;
        cin >> y;
        if (((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0))
            cout << "a leap year\n";
        else
            cout << "a normal year\n";
    }

    return 0;
}
```

input.txt

```
4
1992
1993
1900
2000

```

伺服器的 input file

output.txt

```
a leap year
a normal year
a normal year
a leap year

```

伺服器的 output file

array 陣列

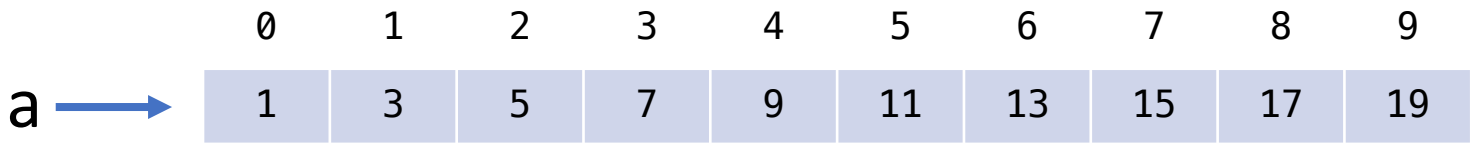
大量具同性質的變數

array 陣列

10個連續的記憶體空間，每一個存放4 bytes大小的整數。可利用下標方式取值。

宣告變數

```
int a[10];
```



index (下標)，從 0 開始
value (存放的內容)

```
a[0] 的值為1
a[1] 的值為3
. . .
a[7] 的值為？
```

陣列的宣告與初始值

Example 4-3

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

int main() {
    int a[10]; //初始值不保證是零
    int b[10] = {};
    int c[10] = {0};
    int d[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    int e[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
```

Variable-sized object may not be initialized

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

int main() {
    int n;
    cin >> n;
    int a[n];

    for (int i = 0; i < n; i++) {
        cout << "a[" << i << "] = " << a[i] << "\n";
    }
    return 0;
}
```

```
10
a[0] = 10
a[1] = 0
a[2] = 1876947969
a[3] = 0
a[4] = 0
a[5] = 0
a[6] = 0
a[7] = 0
a[8] = 48
a[9] = 0
```

兩種初始化的方式

n < 10^5，用 for-loop 比較快

```
// initialize a variable-sized array
for (int i = 0; i < n; i++) {
    a[i] = 0;
}
```

可以將 array 初始化成任意值

#include <cstring>

```
// initialize a variable-sized array
memset(a, 0, sizeof(a));
```

通常用來將 array 初始化成
0 (0x00000000)
-1 (0xFFFFFFFF)
~inf (0x3F3F3F3F)

陣列可透過 下標 (index) 改值

Example 4-4
Example 4-4B

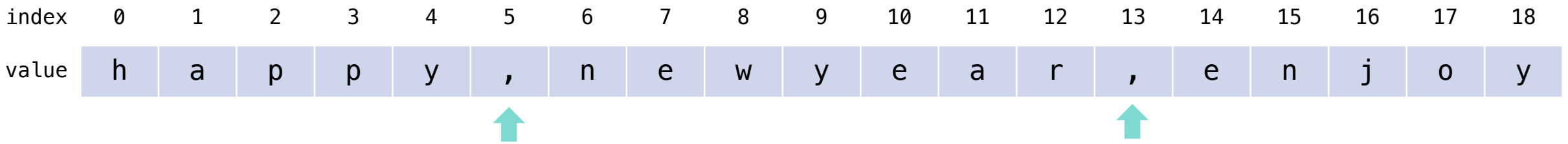
```cpp
#include <iostream>
using namespace std;

int main() {
    int a[5];

    for (int i = 0; i < 5; i++) {
        cin >> a[i];
    }

    for (int i = 0; i < 5; i++) {
        cout << "a[" << i << "] = " << a[i] << ", ";
    }
    cout << "\n";

    a[2] = 0;

    for (int i = 0; i < 5; i++) {
        cout << "a[" << i << "] = " << a[i] << ", ";
    }
    cout << "\n";
}
```

#define 陣列大小

```cpp
#include <iostream>
using namespace std;
#define ArraySize 5

int main() {
    int a[ArraySize];

    for (int i = 0; i < ArraySize; i++) {
        cin >> a[i];
    }

    for (int i = 0; i < ArraySize; i++) {
        cout << "a[" << i << "] = " << a[i] << ", ";
    }
    cout << "\n";

    a[2] = 0;

    for (int i = 0; i < ArraySize; i++) {
        cout << "a[" << i << "] = " << a[i] << ", ";
    }
    cout << "\n";
}
```

字串是一個字元陣列

【範例】輸入一個字串 s，把 s 中的「,」改成「空白」



用 for 迴圈「遍歷」一個字串

Example 4-5

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

int main() {
    string s;
    cin >> s;

    for (int i = 0; i < s.size(); i++) {
        if (s[i] == ',') {
            s[i] = ' ';
        }
    }
    cout << s << "\n";
    return 0;
}
```

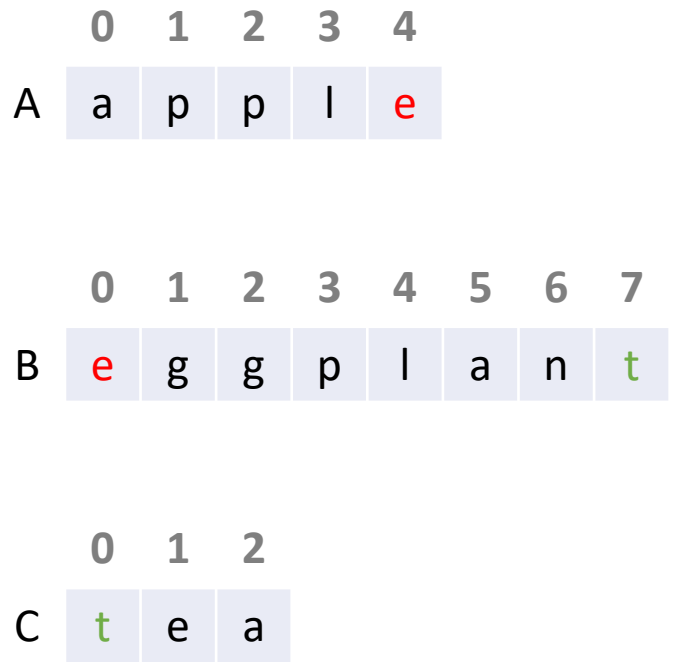
內建函數，回傳字串長度

- `s.size()`
- `s.length()`

【範例】字串接龍

輸入三個字串，判斷它們是否符合接龍的規則

```
【Input-1】 apple eggplant tea
  字串 A = ”apple”
  字串 B = ”eggplant”
  字串 C = ”tea”
【Output-1】Yes

【Input-2】 apple banana orange
【Output-2】No
```



【範例】字串接龍

Example 4-6

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

int main() {
    string A, B, C;
    cin >> A >> B >> C;

    if ((A[A.size() - 1] == B[0]) && (B[B.size() - 1] == C[0])) {
        cout << "Yes\n";
    } else {
        cout << "No\n";
    }
    return 0;
}
```

	0	1	2	3	4
A	a	p	p	l	e

	0	1	2	3	4	5	6	7
B	e	g	g	p	l	a	n	t

	0	1	2
C	t	e	a

【範例】ZeroJudge a022: 迴文

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

int main() {
    string s;
    cin >> s;

    int sz = (int)s.size();
    for (int i = 0; i < sz / 2; i++) {
        if (s[i] != s[sz - 1 - i]) {
            cout << "no\n";
            return 0;
        }
    }
    cout << "yes\n";
    return 0;
}
```

https://pastebin.ubuntu.com/p/JC2f4ThHHZ/

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

int main() {
    string s;
    cin >> s;

    int sz = (int)s.size();
    for (int l = 0, r = sz - 1; l < r; l++, r--) {
        if (s[l] != s[r]) {
            cout << "no\n";
            return 0;
        }
    }
    cout << "yes\n";
    return 0;
}
```

https://pastebin.ubuntu.com/p/ZxqMzpWtMk/

二進位與 ASCII code

二進位

十進位

十位數	個位數
0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
1	0
1	1

二進位

MSB			LSB
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1

用三個位元，可表示 8 種組合

$2^3 = 2 \times 2 \times 2 = 8$

1 + 1 = 10 ？

ASCII Code

Example 4-7

American **S**tandard **C**ode for **I**nformation **I**nterchange

00000000
.
.
.
01111111

$2^7 = 128$

10000000
.
.
.
11111111

$2^7 = 128$

0	<NUL>	32	<SPC>	64	@	96	`
1	<SOH>	33	!	65	A	97	a
2	<STX>	34	"	66	B	98	b
3	<ETX>	35	#	67	C	99	c
4	<EOT>	36	$	68	D	100	d
5	<ENQ>	37	%	69	E	101	e
6	<ACK>	38	&	70	F	102	f
7	<BEL>	39	'	71	G	103	g
8	<BS>	40	(	72	H	104	h
9	<TAB>	41	)	73	I	105	i
10	<LF>	42	*	74	J	106	j
11	<VT>	43	+	75	K	107	k
12	<FF>	44	,	76	L	108	l
13	<CR>	45	-	77	M	109	m
14	<SO>	46	.	78	N	110	n
15	<SI>	47	/	79	O	111	o
16	<DLE>	48	0	80	P	112	p
17	<DC1>	49	1	81	Q	113	q
18	<DC2>	50	2	82	R	114	r
19	<DC3>	51	3	83	S	115	s
20	<DC4>	52	4	84	T	116	t
21	<NAK>	53	5	85	U	117	u
22	<SYN	54	6	86	V	118	v
23	<ETB>	55	7	87	W	119	w
24	<CAN>	56	8	88	X	120	x
25	<EM>	57	9	89	Y	121	y
26	<SUB>	58	:	90	Z	122	z
27	<ESC>	59	;	91	[	123	{
28	<FS>	60	<	92	\	124	|
29	<GS>	61	=	93	]	125	}
30	<RS>	62	>	94	^	126	~
31	<US>	63	?	95	_	127	<DEL>

ASCII-to-binary converter

- ASCII-to-binary converter (C++)

https://pastebin.ubuntu.com/p/QQ6XSgzn8m/

- Binary-to-ASCII converter

https://www.rapidtables.com/convert/number/binary-to-ascii.html

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

string toBinary(string s) {
    string ret = "";
    for (int i = 0; i < s.size(); i++) {
        char c = s[i];
        for (int j = 7; j >= 0; j--) {
            if (c & (1 << j)) {
                ret += '1';
            } else {
                ret += '0';
            }
        }
        ret += ' ';
    }
    return ret;
}

int main() {
    string s = "I like you.";
    cout << toBinary(s) << "\n";
    return 0;
}
```

【範例】三字母縮寫

【Input-1】

advanced micro devices

【Output-1】

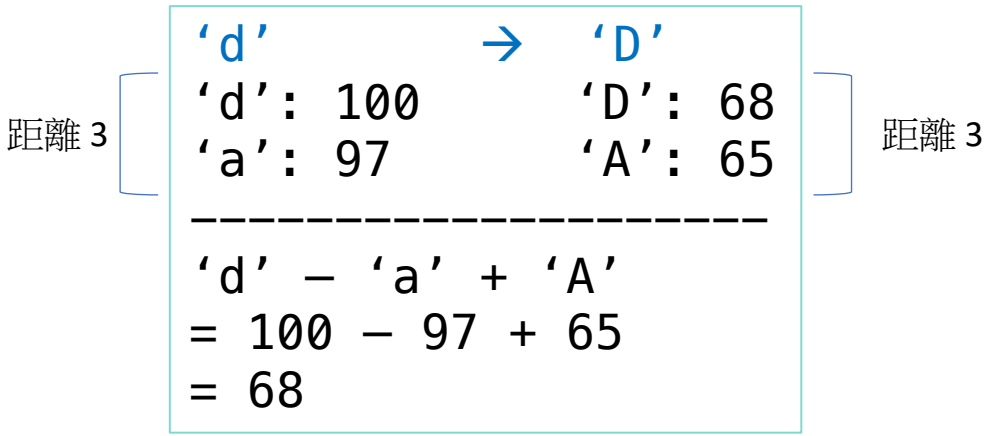
AMD

【Input-2】

three letter acronym

【Output-2】

TLA




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

int main() {
    string s1, s2, s3;
    cin >> s1 >> s2 >> s3;

    cout << (char)(s1[0] - 'a' + 'A');
    cout << (char)(s2[0] - 'a' + 'A');
    cout << (char)(s3[0] - 'a' + 'A');
    cout << "\n";
    return 0;
}
```