Chapter 2.

基礎 C++ 程式設計導論 II

Introduction to Basic C++ Program Design II

講師：陳俊安 Colten

陣列 Array

- 當我想要一次宣告大量變數時，就會需要陣列的幫忙
- 陣列宣告：型態 名字[大小];

```
28      int a[20]; // 宣告 20 個名字叫做 a 的整數
```

陣列 Array

- 可以把陣列想像成很多個箱子，每一個箱子都有一個編號
- int a[20]; 宣告了 20 個名字叫做 a 的整數
- 那如果我想讓第 5 個箱子等於 100，應該怎麼做？
  - a[4] = 100;
  - 記得編號 (index 索引 值) 是從 0 開始的

陣列 Array

- 假設我有 n 個整數要輸入（1 <= n <= 100）

```
17 int a[100];
18
19 int main()
20 {
21     int n;
22     cin >> n;
23
24     for(int i=0;i<n;i++)
25     {
26         cin >> a[i];
27     }
28 }
```

陣列 Array

- 陣列初始化，如果想要一開始讓陣列所有元素為 0 可以這樣寫

```
2
3 int a[10] = {};
```

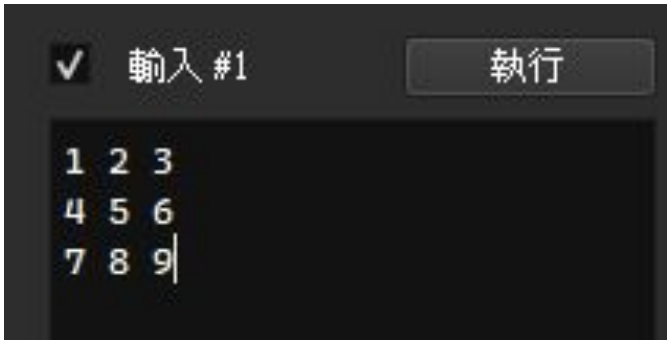
陣列 Array

- 一開始就給予陣列每一個位置的 value

```
int a[10] = {1,2,3,4,5,6,7,8,9,10};
cout << a[0] << "\n"; // 1
```

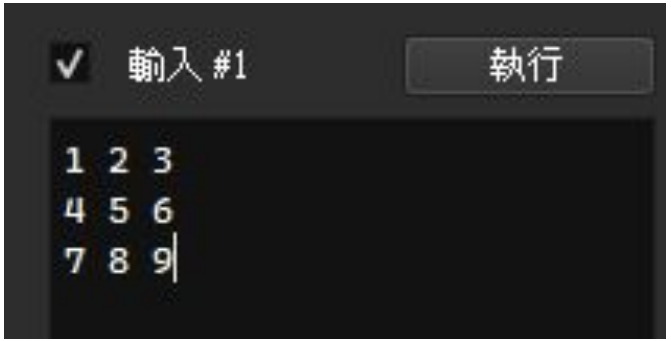
二維陣列

- 給你一個 n * m 的表格，每一個表格上的資訊都是整數
- 請你把所有資訊輸入近來
- 陣列好像只能輸入序列？



二維陣列

- 陣列是可以多維宣告的，一個中括號表示一個維度
- 因此我們可以宣告 int a[3][3]; 來表示 3 * 3 的表格
- 把資訊輸入近來




```cpp
int a[3][3];

int main()
{
    for(int i=0;i<3;i++)
    {
        for(int k=0;k<3;k++)
        {
            cin >> a[i][k];
        }
    }
}
```

二維陣列

- 二維陣列初始化

```
12      int a[3][3] = {
13          1 , 2 , 3,
14          4 , 5 , 6,
15          7 , 8 , 9,
16      };
17
18      cout << a[1][2] << "\n"; // 6
```

陣列的使用

- 如果陣列的大小可能會開很大，建議把宣告變數放到 main 函數的上面
  - 因為程式內部記憶體配置的關係
  - 如果

```
5 using namespace std;
6
7 int a[200005];
8
9 int main(void)
10 {
```

陣列的使用

- 用變數宣告陣列大小是非常不推薦的作法，很容易出問題
  - int a[n];
- 題目通常都會告訴你數字最多會輸入幾個，以那個數字作為依據即可
  - 如果題目說最多輸入 1 萬個數字就開 1 萬大小的陣列

字元與字串 char and string

- char & string
- stringstream
- ASCII

string 字串

- 字串就是一個被很多字元串接起來的東西
  - 因此在 C 中，字串的表示法為一個字元陣列
- 既然字串是一個字元陣列，那我們應該如何存取字串中某一個字元？
- 我們可以直接將字串當一般陣列使用，用索引指定字元位置
- 字串名稱.size() 這個操作可以取得字串的長度

```
string s = "Hello";

cout << s[0] << " " << s[2] << " " << s.size() << "\n"; // H l 5
```

string 字串加法

- 字串跟字串之間是可以做相加的
- 相加完的結果會是兩個字串接起來的結果
- 舉例：
  - Apple + Banana = AppleBanana

```
12      string a = "abc" , b = "OwO";
13
14      cout << a + b << "\n"; // abcOwO
```

ASCII Code

- 這個會輸出什麼東西？
- 為什麼字元可以被轉型成數字？依據是什麼？

```
19
20      cout << (int)'c' << "\n"; // 99
21
```

ASCII Code

16進制表示法：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F、10 ...

Ctrl	Dec	Hex	Char	Code
^@	0	00		NUL
^A	1	01		SOH
^B	2	02		STX
^C	3	03		ETX
^D	4	04		EOT
^E	5	05		ENQ
^F	6	06		ACK
^G	7	07		BEL
^H	8	08		BS
^I	9	09		HT
^J	10	0A		LF
^K	11	0B		VT
^L	12	0C		FF
^M	13	0D		CR
^N	14	0E		SO
^O	15	0F		SI
^P	16	10		DLE
^Q	17	11		DC1
^R	18	12		DC2
^S	19	13		DC3
^T	20	14		DC4
^U	21	15		NAK
^V	22	16		SYN
^W	23	17		ETB
^X	24	18		CAN
^Y	25	19		EM
^Z	26	1A		SUB
^[	27	1B		ESC
^\	28	1C		FS
^]	29	1D		GS
^^	30	1E	▲	RS
^-	31	1F	▼	US

Dec	Hex	Char
32	20	
33	21	!
34	22	"
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	⌂*

* ASCII 碼 127 具有代碼 DEL。在 MS-DOS 下，這個代碼與 ASCII 8 (BS) 的效果相同。DEL 代碼可以由 CTRL + BKSP 鍵產生。

ASCII Code

- 所以當你拿字元做加減乘除的運算的時候
- 程式會自動幫你把字元轉乘相對應的 ASCII Code 整數才做運算

ASCII Code

- 如果我現在想判斷一個數字有沒有某一個位數出現 2 或 3
  - 出現的話要把這一個位數 + x (1 <= x <= 5)
- 但我這個數字大到 10^100，使用 C++ 該怎麼做？
  - Example：12345 , x = 5
  - Answer：17845
- Hint: 可以掃過整個字串

ASCII Code

- 把整個字串掃過一次， 碰到 2 or 3 的時候就直接 +x
  - 字元 + 整數 x = 字元的 ASCII + x
  - 然後由於我們是存到字元裡面，所以新的字元的 ASCII Code 會等於 ASCII + x，程式會自動幫你轉回字元型態

```
20
21      int x;
22      string s;
23      cin >> s >> x;
24
25      for(int i=0;i<s.size();i++)
26      {
27          if( s[i] == '2' || s[i] == '3' ) s[i] += x;
28      }
29
30      cout << s << "\n";
```

stringstream

- #include <sstream>
- 一種處理字元的好工具
- 兩大功能
  - 切割字串
  - 型態轉型

stringstream 切割字串

- 一般陣列的題目都會先跟你說他會輸入幾個
- 但如果你遇到的題目一開始不會告訴你他要輸入幾個怎麼辦？
- 這個時候我們就只能先把整段字串先讀取進來了！

stringstream 切割字串

- stringstream 是一個工具，使用前必須先宣告

```
stringstream name;
```

stringstream 切割字串

- 接下來有兩種操作，假設我的 stringstream 的名字是 ss
  - ss << 變數; // 將變數放入 ss 裡面
  - ss >> 變數; // 將 ss 裡面的東西放入到變數

stringstream 切割字串

- 現在有一個字串 100 200 300，我想把他切成三份
- 每一份都是一個整數，依序存入 a[0],a[1],a[2]
- 如何用 stringstream 達成？
- 我們來解析一下吧！

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

getline(cin,s)

- 由於我們必須使用字串讀入，字串含有空白字元
- 所以我們必須使用 getline

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

stringstream ss; & ss << s;

- 宣告 stringstream
- 然後把我們要處理的字串丟入到這一個 stringstream 裡面

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

int number , a[3] , idx = 0;

- 宣告一個整數的 number，用來存 stringstream 切下來的東西
  - 因為我們切下來的東西是整數，所以 number 要是整數
- 宣告 a[3] 與 idx 來存取資料

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

while(ss >> number)

- stringstream 會依序把東西切下來然後放入 number
- 使用 while 迴圈的目的是：當沒有切到東西的時候才會停止

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

大功告成！

- 這麼一來我們的 a[0] = 100，a[1] = 200，a[2] = 300 了！
- 都是整數型態的哦！

```
21      string s;
22      getline(cin,s);
23
24      stringstream ss;
25
26      ss << s;
27
28      int number,a[3],idx = 0;
29
30      while( ss >> number )
31      {
32          a[idx] = number;
33          idx++;
34      }
```

stringstream 型態轉型

- 假設我有一個字串是 "1.414"，想要轉成 double 的型態？
- stringstream 也可以做到！寫法跟剛剛切字串很像

```
28      string s = "1.414";
29
30      stringstream ss;
31
32      ss << s;
33
34      double u;
35
36      ss >> u;
37
38      cout << u << "\n"; // 1.414
```

練習題

- 輸入字串 "10 20 30000" 請使用 stringstream 切割成 3 個整數
- 輸入字串 "5454"，請你使用 stringstream 轉成整數型態
- 呈上題，你能想到不使用 stringstream 的轉型方法　ㄇ？

兩個字串比大小

- 字串之間是可以比大小的
- 比較的規則使用 字典序，也就是 ASCII Code 的順序
- 我們來看看兩個字串是怎麼比較的吧！

兩個字串比大小

- 假設字串 a = "apple"，b = "apprl"，a 跟 b 誰比較大？
- 比較的時候會由左到右開始逐一字元比較，直到分出勝負
  - a = a (字元 1)
  - p = p (字元 2)
  - p = p (字元 3)
  - l < r (字元 4，分出勝負了，字串 b 比較大)

兩個字串比大小

- 即使字串長度不一樣也是可以比較的，規則一樣
  - Example：
    - abzzzzz < ac
    - gjosgjosjgojsojgosemnmn < z

兩個字串比大小 - 特殊情況

- 那 abb 跟 abba 誰比較大？
- 由於 abb 比到已經沒有東西了還沒分出勝負
- 因此 abba 比 abb 還要大

練習題

- 輸入兩個超過 long long 的正整數，請你寫一個程式判斷誰比較大
  - Hint: ASCII Code 字典序