輸入 & 運算子 & 選擇結構

今天會留比較多實作時間 QQ

變數 × 型別

資料的儲存容器

- int
- long long int
- double
- char
- string
- bool

int (整數)

範圍: $-2^{31} \sim 2^{31} - 1$ (約 $-2 \times 10^9 \sim 2 \times 10^9$)

如果整數超過範圍程式通常會WA

long long int (長整數)

範圍: $-2^{63} \sim 2^{63} - 1$ (約 $-6 \times 10^{18} \sim 6 \times 10^{18}$)

好耶，有這個東西就會少吃很多WA了 :D

double (雙精度浮點數)

請不要用float

用來儲存小數，不過並不是完全精確

由於用了更多空間，比起float還要精確很多

char (字元)

單個字母、數字、符號、特殊按鍵（打不出來）

裡面的東西由兩個單引號（'）包住

特別的是，字元實際上是用數字轉換

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

可以使用一些函數來得到ASCII值

```
cout << int('a') << '\n';
```

或是把整數換成字元

```
cout << char(97) << '\n';
```

備註: '\n'也是一個字元，代表換行

string (字串)

由一連串的字元組合而成

裡面的東西由兩個雙引號（"）包住

bool (布林)

~~就跟206一樣，不是0就是1~~

0代表false，1代表true

可以節省很多空間，並且增加可讀性

宣告

```
1 int a = 1;
2 long long int b = 2;
3 double d = 2.564;
4 char c = 'a';
5 string e = "ABC";
6 bool f = true;
```

也可以這樣寫

```
1 int a=1, b=69, c=69420;
```

請注意！如果不是要處理輸入的變數
請務必設定初始值，**否則會發生不可預期的錯誤！**

輸入

程式的靈魂

cin: 輸入函數

和cout的用法很像

會將輸入的內容隔開，並且把資料放進變數裡面

語法範例

```
1 int a;
2 string b;
3 bool c;
4
5 cin >> a >> b >> c;
```

這裡代表要輸入整數、字串以及布林值各一個

請注意，cin的箭頭方向是相反的

並且裡面不能有變數以外的東西

也可以這樣寫

```
1 int a;
2 string b;
3 bool c;
4
5 cin >> a;
6 cin >> b;
7 cin >> c;
```

好耶 來看看題目

輸入的東西為**字串**，得先宣告一個string的變數

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

int main(){
  string s; // 用來存輸入的變數，名稱叫做s

  return 0;
}
```

由於是從題目端輸入，得先寫cin

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

int main(){
  string s; // 用來存輸入的變數，名稱叫做s

  cin >> s; // 從題目裡面獲得字串的內容

  return 0;
}
```

輸出答案

```
1  上禮拜的內容，自己寫 :)
```

四則運算

簡單的加減乘除 :D

C++如同數學一樣**先乘除，後加減，括號最優先**

以下是常用的運算子

- +: 加法
- -: 減法
- *: 乘法
- /: 除法
- %: 取餘數

例題1

AC code

例題2

WA code

code:

```
1 int a = 5, b = 3;
2 cout << a/b << '\n';
```

output:

```
1
```

原因很簡單，在C++裡面，如果整數除於整數，得到的也會是整數，因此程式就自動忽略了

獲得小數的除法

C++提供了兩個函式來幫助我們

fixed: 小數點以下

setprecision: 設定位數

備註: 需要載入<iomanip>才能使用這兩個函式

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

int main(){
    int a = 5, b = 3;
    cout << fixed << setprecision(10) << 1.0*a/b << '\n';
  // 1.0是浮點數，因此做乘法會變成浮點數
  // 由於前面的都是浮點數，因此除的結果也是浮點數
}
```

選擇結構

人生總有很多選擇題
就像選擇紅醬或白醬義大利麵

直接給code (・ω・)✧

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

int a,b;

int main(){
    cin >> a >> b;

    if(a > b){
        cout << a << " is bigger than " << b << '\n';
    }else if(a == b){
        cout << a << " is equal to " << b << '\n';
    }else{
        cout << a << " is smaller than " << b << '\n';
```

以下是要注意的點

- 條件寫在()裡，要做的事放進{}裡
- 有多個判斷的事，要多寫else if
- 把非以上條件的執行的話，要多寫else
- if-else if-...-else裡面只會執行一件事