

for 迴圈

- 請填入以下程式空白處，使程式結果輸出如下數列。

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

```
int main() {
    int i;
```

```
    for(i=____; ____; ____ ) {
```

```
        cout << i << ", " ;
```

```
    }
```

```
    cout << endl;
```

```
    return 0;
```

```
}
```

```
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
```

```
10, 12, 14, 16, 18, 20, 22, 24, 26, 28,
```

```
5, 10, 15, 20, 25, 30, 35, 40, 45,
```

```
45, 40, 35, 30, 25, 20, 15, 10, 5,
```

```
-10, -7, -4, -1, 2, 5, 8,
```

```
1, 2, 4, 8, 16, 32, 64,
```

4-8.cpp

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

```
int main() {
```

```
    int saving = 1;
```

```
    int total = 0;
```

```
    for(int i=1;i<=30;i++) {
```

```
        total = total + saving;
```

```
        cout << "第 " << i << " 天: ";
```

```
        cout << "存入 " << saving << " 元, ";
```

```
        cout << "共有 " << total << " 元" << endl;
```

```
        saving = saving * 2;
```

```
    }
```

```
    return 0;
```

```
}
```

```
第 1 天: 存入 1 元, 共有 1 元
第 2 天: 存入 2 元, 共有 3 元
第 3 天: 存入 4 元, 共有 7 元
第 4 天: 存入 8 元, 共有 15 元
第 5 天: 存入 16 元, 共有 31 元
第 6 天: 存入 32 元, 共有 63 元
第 7 天: 存入 64 元, 共有 127 元
第 8 天: 存入 128 元, 共有 255 元
第 9 天: 存入 256 元, 共有 511 元
.
```

```
第 28 天: 存入 134217728 元, 共有 268435455 元
第 29 天: 存入 268435456 元, 共有 536870911 元
第 30 天: 存入 536870912 元, 共有 1073741823 元
```

如果現在想要知道 60 天後的結果呢？

4-9.cpp

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

```
int main() {
```

```
    int count=0;
```

```
    for(int i=1;i<=50;i++) {
```

```
        if(i%3 == 0 || i%5 == 0) {
```

```
            cout << i << ", ";
```

```
            count++;
```

```
        }
```

```
    }
```

```
    cout << "\n" << count << endl;
```

```
    return 0;
```

```
}
```

- 變數在使用前宣告即可。
- 此處的變數 *i* 僅在 **for** 迴圈內有被定義，以此範例中，*i* 變數只有在緊接的 {...} 內看得到，到了最後的 **cout** 處其實 *i* 變數沒有被定義。

Lecture 5

break / continue

while 迴圈

do ... while 迴圈

簡單 3D 繪圖

今日內容

- break / continue
- while 迴圈
- do ... while 迴圈

break / continue

迴圈 (loop)

- 迴圈的用途
 - 用來重複執行某一段程式 → 執行**重複性**的事務
 - 之後會介紹到配合陣列 (array) 以對大量資料進行處理 (其實，就是重複性的計算)
- C/C++ 的迴圈 (loop) 指令
 - for(...;...;...) { ... } – 多應用於具固定次數的重複性
 - while(...) { ... } – 應用於重複次數不固定的場合
 - do { ... } while(...); – 應用於重複次數不固定的場合

break / continue

- break;
 - To pre-terminate **a loop**
 - 用來提早結束一個迴圈的執行。
 - 跳脫目前所在的區塊、從目前所在區塊外下面一行敘述繼續執行。
- continue;
 - To pre-terminate **an iteration**
 - 用來提早結束一次迭代的執行。
- 可使用在 for、do/while、while三種迴圈中

5-1.cpp & 5-2.cpp

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

```
int main() {
    int i;

    for(i=0;i<15;i++) {
        if(i>=5 && i<10) break;
        cout << "i=" << i <<
        endl;
    }
    cout << "i-迴圈結束" <<
    endl;
}
```

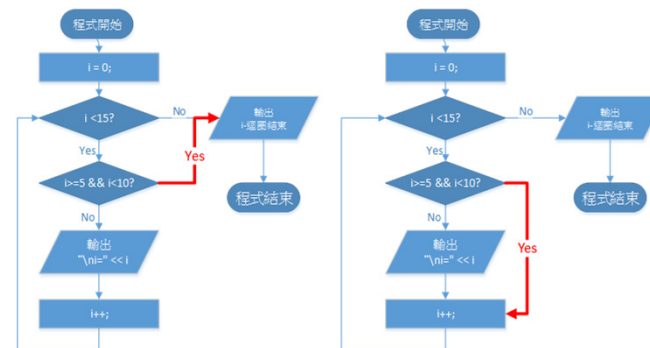
```
return 0;
```

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

```
int main() {
    int i;

    for(i=0;i<15;i++) {
        if(i>=5 && i<10)
            continue;
        cout << "i=" << i <<
        endl;
    }
    cout << "i-迴圈結束" <<
    endl;
}
```

5-1.cpp & 5-2.cpp 的流程圖比較



5-1 & 5-2 執行結果

```
i=0
i=1
i=2
i=3
i=4
i-迴圈結束
```

```
for(i=0;i<15;i++) {
    if(i>=5 && i<10) break;
    cout << "i=" << i <<
    endl;
}
cout << "i-迴圈結束" <<
endl;
```

```
i=0
i=1
i=2
i=3
i=4
i=10
i=11
i=12
i=13
i=14
i-迴圈結束
```

while 迴圈

while 的語法

- `while` (條件運算式) {
 欲重複執行的程式片段;
 ...
}
- 其中條件運算式 / 布林運算式的部份與 `if` 的寫法一樣
 - `count > 0`
 - `a < 10 && b < 10`
 - ...
- `while` 和 `for` 迴圈一樣，都會先檢查條件運算式的結果決定迴圈的內容是否會被執行，所以有可能一次都不會被執行。

5-3.cpp

```
#include <iostream>
using namespace std;
int main() {
    int count, sum=0;

    cout << "請輸入一個整數: ";
    cin >> count;
    while(count>0) {
        sum += count;
        cout<< "count=" << count << ", sum=" << sum << endl;
        count--;
    }

    return 0;
}
```

請輸入一個整數: 10
count=10, sum=10
count=9, sum=19
count=8, sum=27
count=7, sum=34
count=6, sum=40
count=5, sum=45
count=4, sum=49
count=3, sum=52
count=2, sum=54
count=1, sum=55

while 迴圈

```
int a = 0;
1 while( a < 3) {
2     cout << "a";
3     a++;
4 }
5 cout << "b";
```

- 1 判斷迴圈內容是否執行或繼續執行
- 2 迴圈內欲重複執行的敘述
- 3 迴圈結束後要執行的敘述

1	0 < 3 → true
2	cout << "a";
2	a++;
1	1 < 3 → true
2	cout << "a";
2	a++;
1	2 < 3 → true
2	cout << "a";
2	a++;
1	3 < 3 → false
3	cout << "b";

5-4.cpp

```
#include <iostream>
using namespace std;
int main() {
    int ans;
    bool isOdd = true;

    while(isOdd) {
        cout << "請輸入一個偶數: ";
        cin >> ans;

        isOdd = (ans % 2 == 1);
    }
    cout << "你輸入了一個偶數: " << ans << endl;

    return 0;
}
```

請輸入一個偶數: 1
請輸入一個偶數: 3
請輸入一個偶數: 5
請輸入一個偶數: 7
請輸入一個偶數: 6
你輸入了一個偶數: 6

5-5.cpp

- 費氏級數 (Fibonacci number) 之定義如下:
 - 1, 1, 2, 3, 5, 8, 13, 21, ...
 - 級數中每個數字皆為級數中的前兩個數字加總
- 如何寫一個程式依序列出級數中小於 150 的各項?
 - 我們自己會怎麼作?
 - 過程當中我們需要同時記得幾個資料?

do/while 迴圈

5-5.cpp

```
#include <iostream>
using namespace std;
int main() {
    int a=1, b=1, c=a+b;

    cout << a << ", " << b << ", ";
    while(c<=150) {
        a=b;
        b=c;
        cout << b << ", ";
        c=a+b;
    }

    return 0;
}
```

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,

do/while 迴圈

```
int a = 0;

do {
    1  cout << "a";
    2  a++;
    3  } while(a<3);
    cout << "b";
```

1	cout << "a";
1	a++;
2	1 < 3 → true
1	cout << "a";
1	a++;
2	2 < 3 → true
1	cout << "a";
1	a++;
2	3 < 3 → false
3	cout << "b";

1 迴圈內欲重複執行的敘述
 2 迴圈要執行或要被繼續執行的條件
 3 迴圈結束後要執行的敘述

do/while 的語法

- do {
 欲重複執行的程式片段;
 ...
} while (布林運算式); ← 很重要
- 其中布林運算式的部份與 if 的寫法一樣
 - count>0
 - a<10 && b<10
 - ...
- do/while 迴圈與 while 或 for 最大不一樣的地方在於它的條件判斷在迴圈內容之後，亦即它的迴圈內容 **被保證至少執行到一次**。
- 記得 while (...) 之後要加分號！

5-7.cpp

```
#include <iostream>
using namespace std;
int main() {
    int ans;

    do {
        cout << "請輸入一個偶數: ";
        cin >> ans;
    } while(ans%2 != 0);

    cout << "你輸入了一個偶數" << endl;

    return 0;
}
```

請輸入一個偶數: 1
請輸入一個偶數: 3
請輸入一個偶數: 5
請輸入一個偶數: 7
請輸入一個偶數: 6
你輸入了一個偶數

5-6.cpp

```
#include <iostream>
using namespace std;
int main() {
    char again;
    float weight, height, BMI;
    do {
        cout << "請輸入你的身高 (公分): ";
        cin >> height;
        cout << "請輸入你的體重 (公斤): ";
        cin >> weight;
        BMI = weight*10000 / height / height;
        cout << "你的 BMI 值為: " << BMI << endl;
        cout << "再算一次請按 y: ";
        cin >> again;
    } while(again=='y' || again=='Y');
    return 0;
}
```

請輸入你的身高 (公分): 170
請輸入你的體重 (公斤): 60
你的 BMI 值為: 20.7612
再算一次請按 y: y
請輸入你的身高 (公分): 180
請輸入你的體重 (公斤): 60
你的 BMI 值為: 18.5185
再算一次請按 y: y
請輸入你的身高 (公分): 172
請輸入你的體重 (公斤): 55
你的 BMI 值為: 18.5911
再算一次請按 y: n

小結

- while
- do ... while
- 注意, 之前介紹過的 break 與 continue 一樣也可以使用在 while / do-while 迴圈內

簡單 3D 繪圖

5-8.cpp

```
#include "3d.h"
using namespace TT3D;

int main() {
    creator world("d:\\test.html");
    world.setZoom(25);

    world.addBall(1, 1.5, 1, 0.5);
    world.addBall(2, 0, 0, 1.5, BLACK);
    world.addBall(2, 0, 2, 1.5, YELLOW);
    world.addBall(0, 0, 2, 1.5, colors::CYAN);
    world.addBall(0, 0, 0, 1.5, colors::MAGENTA);

    world.addLine(0, 0, 0, 10, 0, 0, RED);
    world.addLine(0, 0, 0, 0, 10, 0, GREEN);
    world.addLine(0, 0, 0, 0, 0, 10, BLUE);

    world.show();
    return 0;
}
```

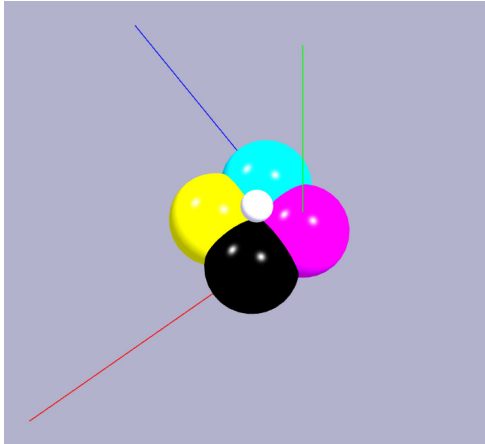
3D 電腦繪圖

- 背後牽涉的數學、矩陣、向量等運算是非常多的。
 - 請修習 **CT5706701**「電腦繪圖與視覺模擬」
- 在這門課裡，我們把一些電腦繪圖裡複雜的事情給包裝起來了(透過函式、類別、...)，所以我沒有包裝的東西、功能、你就沒辦法做了。
- 能做到的事
 - 放置球體
 - 放置六面體(方盒子)
 - 放置英文文字、數字、標點符號
 - 放置已建好的 3D 模型

5-8.cpp

- **world.setZoom(25);**
 - 設定縮小倍率，數字愈大站得愈遠(東西看起來愈小)
- **world.addBall(x, y, z, r, color);**
 - 用來放置球體
 - 至少要放置的球的球心 (x, y, z) 與半徑 r
 - 可再額外指定顏色(預設為白色)
 - 顏色的指定方式可用英文(全部大寫，僅認識 RED, GREEN, BLUE, YELLOW, CYAN, MAGENTA, BLACK, WHITE等八種)
 - 也可以使用 color(紅光強度、綠光強度、藍光強度) 來指定
 - e.g. color(1.0, 0.0, 0.0) ← RED
- **world.addLine(x1, y1, z1, x2, y2, z2, color)**
 - 由 (x1, y1, z1) 畫一條線段至 (x2, y2, z2)
 - 可額外指定顏色color，預設為白色

29



31



30

5-9.cpp

```
#include "3d.h"
using namespace TT3D;

int main() {
    creator world("d:\\test.html");
    world.setZoom(25);

    world.addBox(1, 1, 1, 2, 2, 2, color(0.50, 0.00, 0.75));
    world.addBox(0.5, 2.5, 0.5, 1, 1, 1, RED);
    world.addBox(1.5, 2.5, 0.5, 1, 1, 1, BLUE);
    world.addBox(1.5, 2.5, 1.5, 1, 1, 1, GREEN);
    world.addBox(0.5, 2.5, 1.5, 1, 1, 1, color(0.5, 0.5, 0.1));

    world.addText(0, 0, 0, "Hi! 3D", RED, 3);
    world.addText(0, 0, 0, "Hi# 3D", GREEN, 6, 1, rotate(1, 0, 0, 1.57));
    world.addText(0, 0, 0, "Hi@ 3D", BLUE, 9, 5, rotate(0, 1, 0, -1.57));

    world.show();
    return 0;
}
```

32

5-9.cpp

- **world.addBox(cx, cy, cz, sx, sy, sz, color, rotate)**
 - 放置一以 (cx, cy, cz) 為中心、sx, sy, sz 為三邊長的盒子
 - 可額外指定顏色 color，預設為白色
 - 可再額外指定旋轉量 rotate
- **world.addText(x, y, z, “要寫的字”, color, size, width, rotate)**
 - 在 (x, y, z) 位置開始寫字 (英文、數字、標點符號)
 - 可額外指定顏色 color，預設為白色
 - 可再額外設定字的大小為 size
 - 可再額外設定筆劃寬度為 width (不明顯)
 - 可再額外指定旋轉量 rotate，指定方式為 rotate(x, y, z, r)，代表為繞著 (x, y, z) 方向旋轉 r 這麼多程度。

5-10.cpp

```
#include "3d.h"
using namespace TT3D;

int main() {
    creator world("d:\\test.html");
    world.setZoom(25);

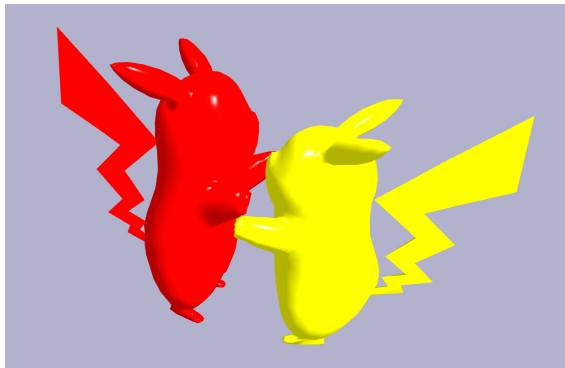
    model pikachu("https://yo-1.ct.ntust.edu.tw/Courses/CVB/2017/#####.obj");
    pikachu.setColor(YELLOW);
    pikachu.setRotation(rotate(0, 1, 0, 1.57f));
    world.addModel(pikachu);

    pikachu.setColor(RED);
    pikachu.setPosition(3, 0, 0);
    pikachu.setRotation(rotate(0, 1, 0, -1.57f));
    pikachu.setScale(scale(1.0f, 1.2f, 1.0f));
    world.addModel(pikachu);

    world.show();
    return 0;
}
```

5-10.cpp

- `model pikachu` (“模型網址”);
 - 由模型的網址下載模型並存入 `pikachu` 這個變數/物件
 - 請由 <https://yo-1.ct.ntust.edu.tw/Courses/CVB/2017/system/modelUpload.php> 上傳你要使用的模型
 - 可接受的模型格式為
 - md2: Alias Wavefront
 - 3ds: Autodesk 3Ds Max 3D 製作出來的模型
 - obj: id Software
- `pikachu.setColor(color...)`;
 - 設定模型繪製時的顏色，顏色指定方法同前
- `pikachu.setRotation(rotate...)`;
 - 設定模型的旋轉量，設定方法同前
- `world.addModel(pikachu)`;
 - 將設定好的 `pikachu` 加入場景中



5-10.cpp

- `pikachu.setPosition(x, y, z)`;
 - 設定模型放置於 (x, y, z) 的位置上
- `pikachu.setScale(sx, sy, sz)`;
 - 設定模型在三個方向上的縮放比例為 (sx, sy, sz)
- `world.addModel(pikachu)`;
 - 將設定好的 `pikachu` 加入場景中
- 找尋 3D 模型
 - Google: model download 3ds
 - Google: model download obj

3.24 隨堂練習

假設一顆球的初始位置在 $(0, 0, 0)$ ，請撰寫一程式，讓使用者輸入該顆球的初始速度 (V_x, V_y, V_z) ，以及時間間隔 dt ，接下來計算該球體每 dt 秒所在的位置與速度直到該球體的 y 座標 ≤ 0 為止（重力作用於 y 方向），最後列印出來在空中飛行的時間，並利用今日所介紹的 3D 繪圖，將該球體的運動軌跡繪製出來。

$$x = x_0 + v_x \times dt$$

$$y = y_0 + v_y \times dt - \frac{g}{2} \times dt^2$$

$$z = z_0 + v_z \times dt$$

HW04

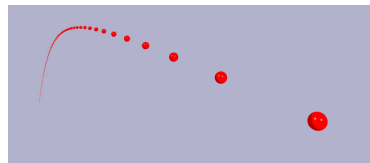
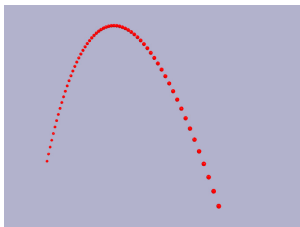
Due: 3/31/2017

請輸入 V_x, V_y, V_z 與 dt : 10 30 10 0.1

請輸入 V_x, V_y, V_z 與 dt : 10 30 100 0.1

球在 6.2 秒後落地。

球在 6.2 秒後落地。



1. 儲蓄

請撰寫一程式，輸入某人的月薪、第一個月的花費、以及之後每個月花費的成長倍數，你的程式要強制使用者輸入一大於1的成長倍數！輸入完成後，輸出每個月的花費、每個月的剩餘金額直到餘額為0為止。其中必需注意到最後一個月的手費不能使餘額為負值。

請輸入月薪：40000
請輸入第一個月的花費：25000
請輸入每個月花費的成長倍數（需大於1）：1.2

第1個月，花費 25000 元，剩 15000 元。
第2個月，花費 30000 元，剩 25000 元。
第3個月，花費 36000 元，剩 29000 元。
第4個月，花費 43200 元，剩 25800 元。
第5個月，花費 51840 元，剩 13960 元。
第6個月，花費 53960 元，剩 0 元。

請輸入月薪：100000 請輸入第一個月的花費：30000 請輸入每個月花費的成長倍數 (需大於1)：0.5 請輸入每個月花費的成長倍數 (需大於1)：0.75 請輸入每個月花費的成長倍數 (需大於1)：1.2	
第 1 個月，花費 30000 元，剩 70000元。 第 2 個月，花費 36000 元，剩 134000元。 第 3 個月，花費 43200 元，剩 190800元。 第 4 個月，花費 51840 元，剩 238960元。 第 5 個月，花費 62208 元，剩 276752元。 第 6 個月，花費 74649 元，剩 302103元。 第 7 個月，花費 89578 元，剩 312525元。 第 8 個月，花費 107493 元，剩 305032元。 第 9 個月，花費 128991 元，剩 276041元。 第 10 個月，花費 154789 元，剩 221252元。 第 11 個月，花費 185746 元，剩 135506元。 第 12 個月，花費 222895 元，剩 12611元。 第 13 個月，花費 112611 元，剩 0元。	請輸入月薪：25000 請輸入第一個月的花費：12000 請輸入每個月花費的成長倍數 (需大於1)：1.1 第 1 個月，花費 12000 元，剩 13000元。 第 2 個月，花費 13200 元，剩 24800元。 第 3 個月，花費 14520 元，剩 35280元。 第 4 個月，花費 15972 元，剩 44308元。 第 5 個月，花費 17569 元，剩 51739元。 第 6 個月，花費 19325 元，剩 57414元。 第 7 個月，花費 21257 元，剩 61157元。 第 8 個月，花費 23382 元，剩 62775元。 第 9 個月，花費 25720 元，剩 62055元。 第 10 個月，花費 28292 元，剩 58763元。 第 11 個月，花費 31121 元，剩 52642元。 第 12 個月，花費 34233 元，剩 43409元。 第 13 個月，花費 37656 元，剩 30753元。 第 14 個月，花費 41421 元，剩 14332元。 第 15 個月，花費 39332 元，剩 0元。