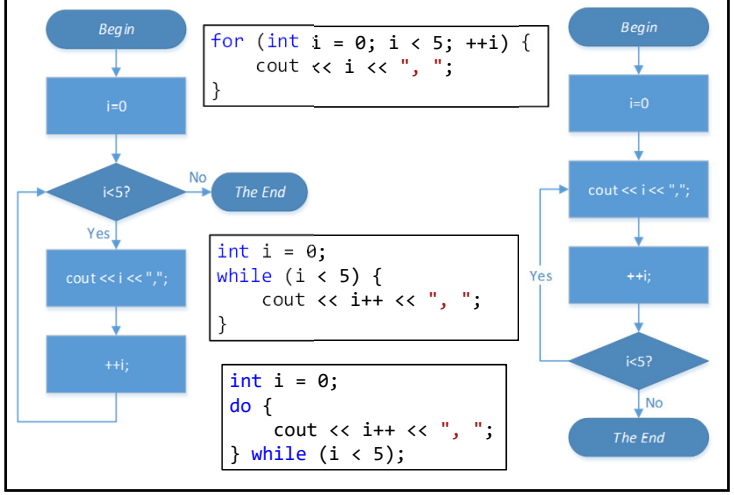
1

回顧

- for (____; ____; ____) { … }
  - for(int i=0;i<5;++i) cout << i << “, “ ;
- while(…) { … }
  - int i=0; while(i<5) {cout << i++ << “, ”; }
- do { … } while(…);
  - int i=0; do {cout << i++ << “, ”;} while(i<5);

2



3

5-1.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;
}
```

```
i=0
i=1
i=2
i=3
i=4
i-迴圈結束
```

4

5-2.cpp

```
#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;

  return 0;
}
```

```
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
```

5

6-1.cpp

```
do {
  cout << "\n請輸入一個正整數; ";
  cin >> x;
} while(x<=0);
```

```
int x, i;
cout << "\n請輸入一個正整數; ";
cin >> x;

for(i=2;i<=x;i++) {
  if(x % i == 0) break;
}

if(i == x)
  cout << x << " 為質數。";
else
  cout << x << " 不是質數，可以被" << i << " 整除。";
```

6

Lecture 6

無窮迴圈
巢狀迴圈
一維陣列與陣列初始化

7

今日內容

- 無窮迴圈
- 巢狀迴圈
- 一維陣列
- 陣列初始化

8

無窮迴圈

Infinite loop

9

無窮迴圈 (Infinite loop)

- 沒有終止條件的迴圈
  - 用於無論在迭代開始 (for, while) 或是結束 (do/while) 來判斷是否進行下次迭代都不合適的時候，一般配合 break 使用。
  - 真的不想結束程式的時候 ...

- `for(;;) {…}`
- `while(true) {…}`
- `do {…} while(1);`

10

6-2.cpp

這個 else 有沒有差別？

對程式執行結果而言，沒有！
就程式效率而言，有！

```
int choice;

for(;;) {
  cout << "\n歡迎使用本程式";
  cout << "\n輸入 1) 問候, 2) 感謝, 3) 再見: ";
  cin >> choice;
  if(choice==3) break;
  if(choice==1) {
    cout << "\nHello there.  How are you? ";
  }
  else if(choice==2) {
    cout << "\nThank you very much. ";
  }
}

cout << "\nSee you next time" << endl;
```

11

```
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 1

Hello there.  How are you?
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 2

Thank you ver much.
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 1

Hello there.  How are you?
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 3

See you next time
```

12

巢狀迴圈

nested loop

13

巢狀迴圈

- 就是迴圈裡面還有迴圈
  - for, while, do/while 三種迴圈可以互相混合
- 層數幾乎沒有限制
- 被包起來的迴圈稱作「內迴圈」(inner-loop)，包其它迴圈的稱作「外迴圈」(outer-loop)。
  - 外迴圈 (e.g. i) 通常不會使用或參考到內迴圈的計數器 (e.g. j)
  - 內外層的迴圈計數器的增量**通常**獨立控制
  - 內迴圈 (e.g. j) 有時會參考到外迴圈 (i) 的計數器

第一層迴圈 / 第二層迴圈

```
for(int i=0;i<3;i++) {
  cout << "*, i=" << i << endl;
  for(int j=0;j<4;j++) {
    cout << "**, i=" << i << ", j=" << j << endl;
  }
}
```

14

6-3.cpp

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

int main() {
  int j;

  for(int i=10; i<=20; i=i+5) {
    cout << "* " << i << endl;
    for(j=2; j<=4; j++) {
      cout << "** " << i << ", " << j << endl;
    }
    cout << "+ " << i << ", " << j << endl;
  }
  return 0;
}
```

```
* 10
** 10, 2
** 10, 3
** 10, 4
+ 10, 5
* 15
** 15, 2
** 15, 3
** 15, 4
+ 15, 5
* 20
** 20, 2
** 20, 3
** 20, 4
+ 20, 5
```

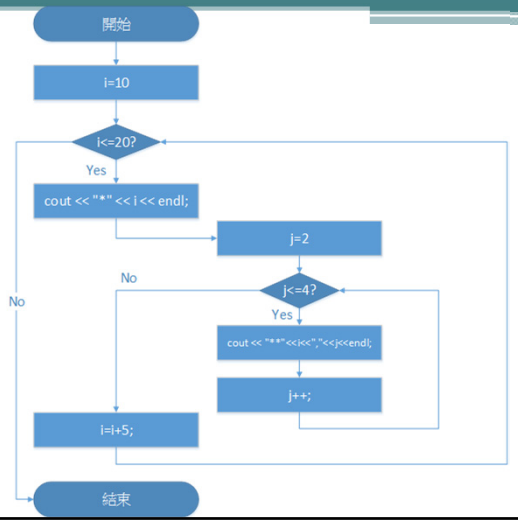
15

6-3.cpp

```
int j;
                1a    1b    1c
for(int i=10; i<=20; i=i+5) {
  cout << "* " << i << endl;
    3a    3b    3c
  for(j=2; j<=4; j++) {
    cout<<"** "<<i<<", "<<j<<endl;
  }
  cout<<"+ "<<i<<", "<<j<<endl;
}
```

1a	1b		2	* 10
3a	3b	4		** 10, 2
3c	3b	4		** 10, 3
3c	3b	4		** 10, 4
3c	3b	5		+ 10, 5
1c	1b		2	* 15
3a	3b	4		** 15, 2
3c	3b	4		** 15, 3
3c	3b	4		** 15, 4
3c	3b	5		+ 15, 5
1c	1b		2	* 20
3a	3b	4		** 20, 2
3c	3b	4		** 20, 3
3c	3b	4		** 20, 4
3c	3b	5		+ 20, 5
1c	1b		6	

16



17

6-4.cpp

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

int main() {
  int i, j;

  for(i=0; i<5; i++) {
    for(j=0; j<4; j++) {
      cout << "*";
    }
    cout << endl;
  }

  return 0;
}
```

```
****
****
****
****
****
```

18

6-5.cpp

這個範例裡，內迴圈的次數相依於外迴圈的計數器 i

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

int main() {
  int i, j;

  for(i=0; i<10; i++) {
    for(j=0; j<i; j++) {
      cout << "*";
    }
    cout << endl;
  }

  return 0;
}
```

```
*
**
***
****
*****
******
*******
********
*********
```

19

6-6.cpp

- 我們想列出兩位數字中 (總共有幾個?) 各位數不相同者 ，並計算總共有幾個符合條件的數字。
  - e.g. 11, 22, 33 等不符合條件
  1. 我們怎麼列出所有兩位數?

```
for(int i=10;i<100;++i) {
  int a = i / 10;
  int b = i % 10;
}
```

```
for(int a=1;a<10;++a) {
  for(int b=0;b<10;++b) {

  }
}
```

  2. 我們怎麼將需求之條件 (各個位數不相同者) 寫成程式?

20

6-6.cpp

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

  for (int i = 1; i < 10; i++) {
    for (int j = 0; j < 10; j++) {
      if (i != j) {
        cout << i << j << ",";
        count++;
      }
    }
  }
  cout << "\n共有: " << count << "個" << endl;
  return 0;
}
```

6-7.cpp

請撰寫一程式列出兩顆骰子可能的點數組合

	1	2	3	4	5	6
1	1:1	2:1	3:1	4:1	5:1	6:1
2	1:2	2:2	3:2	4:2	5:2	6:2
3	1:3	2:3	3:3	4:3	5:3	6:3
4	1:4	2:4	3:4	4:4	5:4	6:4
5	1:5	2:5	3:5	4:5	5:5	6:5
6	1:6	2:6	3:6	4:6	5:6	6:6

1, 1
1, 2
1, 3
1, 4
1, 5
1, 6
2, 2
2, 3
2, 4
2, 5
2, 6
3, 3
3, 4
3, 5
3, 6
4, 4
4, 5
4, 6
5, 5
5, 6
6, 6

6-7.cpp

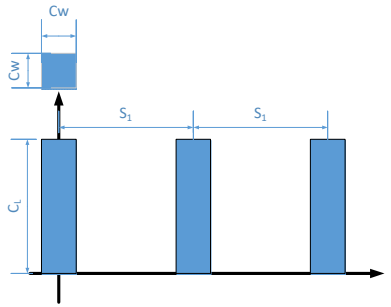
```
#include <iostream>
using namespace std;

int main() {
    for (int i = 1; i <= 6; ++i) {
        for (int j = i; j <= 6; ++j) {
            cout << i << ", " << j << "\n";
        }
    }

    return 0;
}
```


期中考

- 日期：??
- 地點：??
- 時間：9:10開始，原則上兩個小時，視需要增加。
- 範圍：至今日上課內容
  - 變數：變數名稱、變數宣告、如何選用變數型態
  - 運算：各種運算子
  - 輸入輸出：如何列印、如何從鍵盤讓使用者輸入資料
  - 流程控制：if ... else, switch case
  - 迴圈：for, while, do ... while
- 不會有：3D 繪圖、格式化輸出
- 每人限帶一張「自己」準備的「備忘錄」，A4 紙、兩面可用。

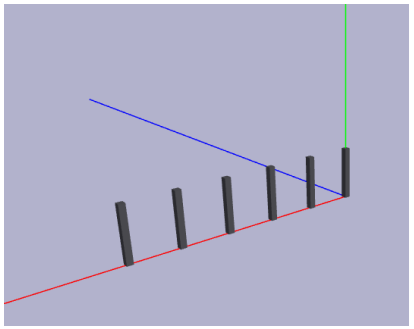
3.31隨堂練習(1/4)

請撰寫一程式，讓使用者輸正方形斷面柱子的邊長C_W與高度C_L、以及1方向上的間距S_1與根數N_1，並利用上週介紹的簡單3D繪圖依據使用者輸入的參數繪出對應的3D圖形。



25

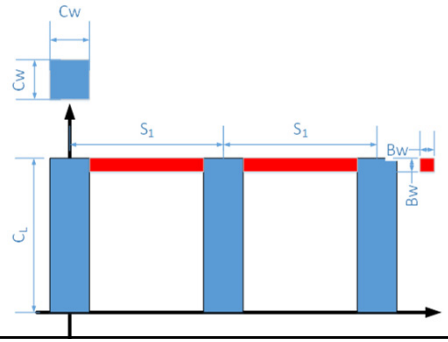
請輸入柱斷面尺寸 Cw: 0.4
請輸入柱長度 Cl: 3.6
請輸入 1 方向上的柱間隔 S1: 3.2
請輸入 1 方向上的柱根數 N1: 6



26

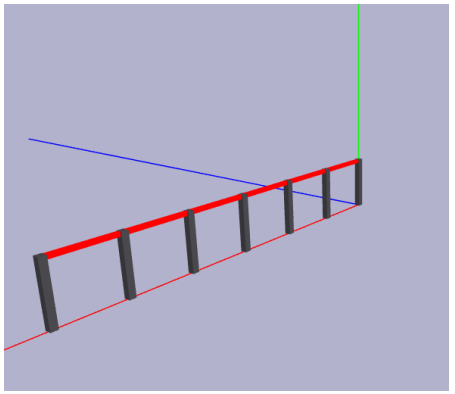
3.31隨堂練習(2/4)

承上，並額外輸入正方面梁斷面尺寸 Bw,並在兩兩柱之間最頂端繪製梁



27

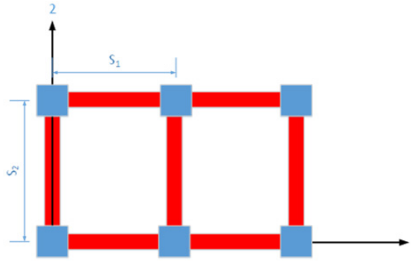
請輸入柱斷面尺寸 Cw: 0.4
請輸入柱長度 Cl: 3.6
請輸入 1 方向上的柱間隔 S1: 4
請輸入 1 方向上的柱根數 N1: 7
請輸入梁斷面尺寸 Bw: 0.25



28

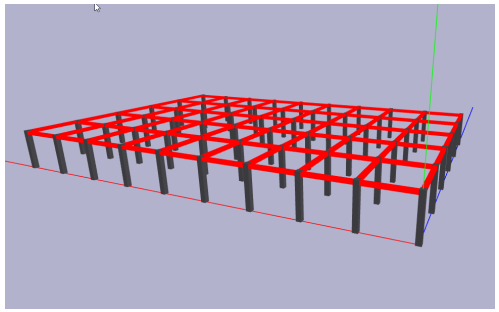
3.31隨堂練習(3/4)

承上，並額外輸入2方向上的間距S_2與根數N_2，並繪製出一層樓完整的梁與柱。



29

請輸入柱斷面尺寸 Cw: 0.4
請輸入柱長度 Cl: 3.6
請輸入梁斷面尺寸 Bw: 0.3
請輸入 1 方向上的柱間隔 S1: 4
請輸入 1 方向上的柱根數 N1: 10
請輸入 2 方向上的柱間隔 S2: 4.8
請輸入 2 方向上的柱根數 N2: 6

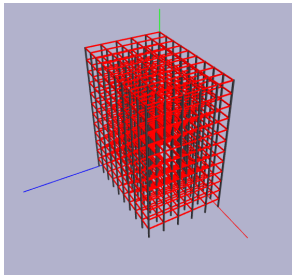


30

3.31隨堂練習(4/4)

承上，並額外輸入層數N_3，並繪製出整棟大樓的梁柱。

請輸入柱斷面尺寸 Cw: 0.4
請輸入柱長度 Cl: 3.6
請輸入梁斷面尺寸 Bw: 0.3
請輸入 1 方向上的柱間隔 S1: 4
請輸入 1 方向上的柱根數 N1: 10
請輸入 2 方向上的柱間隔 S2: 4.8
請輸入 2 方向上的柱根數 N2: 6
請輸入 3 方向上的層數 N3: 12



31

作業五

32

排列組合

請寫一程式，列出 0 – 7 所組成的三位數中互相不重複的所有數字，並印出共有幾個符合條件的數字，且每列輸出不超過7個數字。

102, 103, 104, 105, 106, 107, 120,
123, 124, 125, 126, 127, 130, 132,
134, 135, 136, 137, 140, 142, 143,
145, 146, 147, 150, 152, 153, 154,
156, 157, 160, 162, 163, 164, 165,
167, 170, 172, 173, 174, 175, 176,
201, 203, 204, 205, 206, 207, 210,
213, 214, 215, 216, 217, 230, 231,
234, 235, 236…

756, 760, 761, 762, 763, 764, 765,

共有：294個