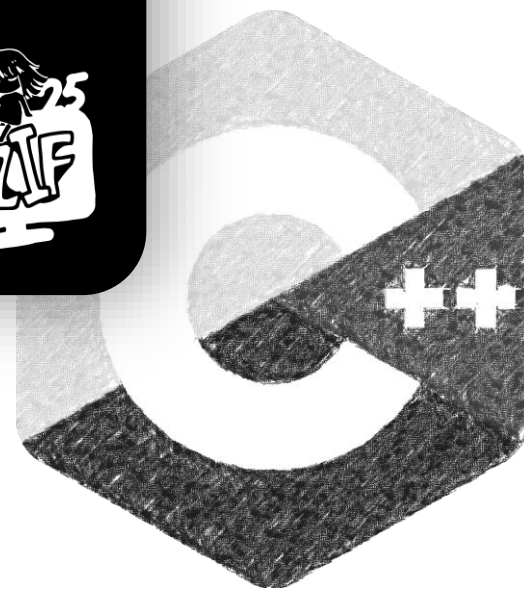




迴圈

1028資訊社



邏輯運算 – AND 「而且」 邏輯

AND運算二側的運算元須同時為真，其結果才為真

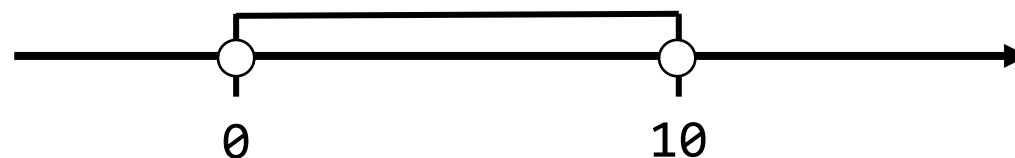
AND運算	A: true	A: false
B: true	true	false
B: false	false	false

在C++中為「&&」或「and」

邏輯運算 – AND 「而且」 邏輯

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int a;
6      cin >> a;
7      if (0 < a && a < 10)
8          cout << "0 < " << a << " < 10\n";
9  }
```

範例：判斷一數是否在某一區間內



需要 $0 < a$ 和 $a < 10$
同時為true，運算結果才
為true

邏輯運算 – OR 「或者」 邏輯

OR運算二側的運算元只要有一者為真，其結果恆為真

OR運算	A: true	A: false
B: true	true	true
B: false	true	false

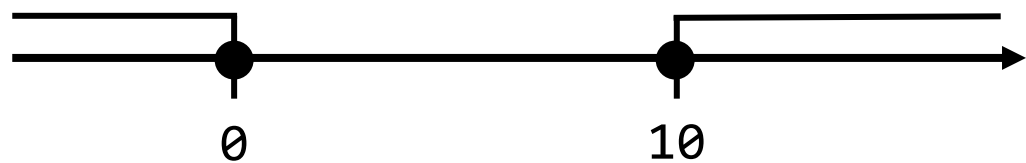
在C++中為「||」或「or」



邏輯運算 – OR 「或者」 邏輯

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int a;
6      cin >> a;
7      if (a <= 0 || a >= 10)
8          cout << "a <= 0 || a >= 10\n";
9  }
```

範例：判斷一數是否在某一區間內



需要 $a \leq 0$ 或 $a \geq 10$
任一為true，運算結果為
true

邏輯運算 – NOT「反向」邏輯

當運算元為真時，NOT運算結果為非。當運算元為非時，NOT運算結果為真。

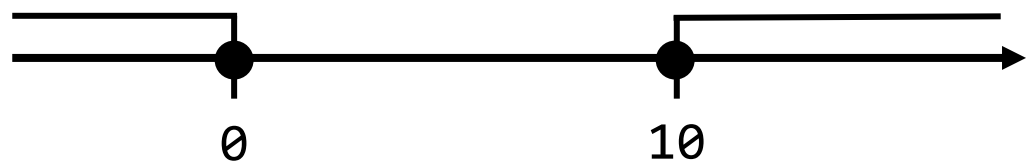
NOT運算	A: true	A: false
	false	true

在C++中為「!」或「not」

邏輯運算 – NOT「反向」邏輯

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int a;
6      cin >> a;
7      if (!(0 < a && a < 10))
8          cout << "!(0 < a && a < 10)";
9  }
```

範例：判斷一數是否在某一區間內



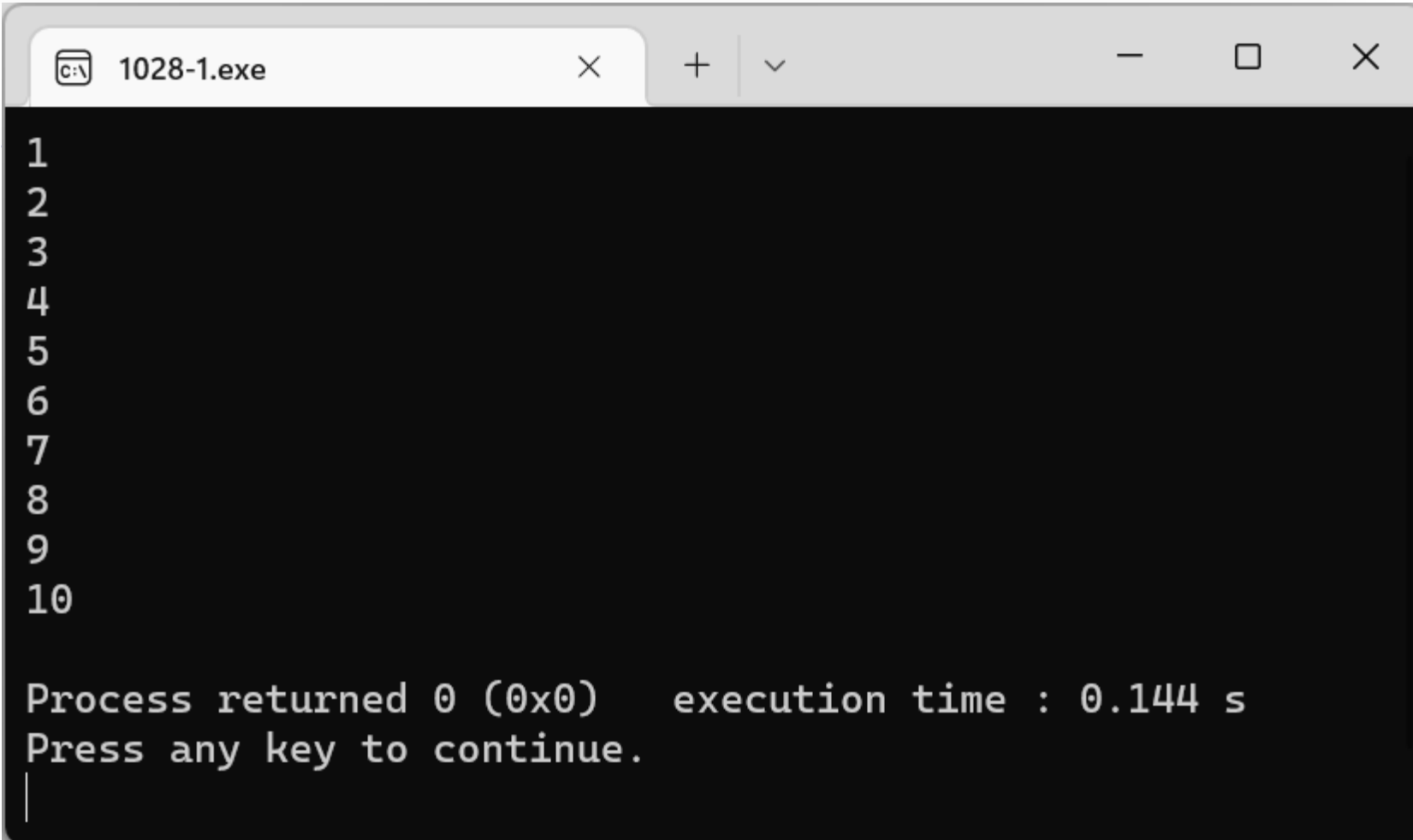
需要 $0 < a \ \&\& \ a < 10$
的結果為false時，運算結果才為true

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << 1 << endl;
7      cout << 2 << endl;
8      cout << 3 << endl;
9      cout << 4 << endl;
10     cout << 5 << endl;
11     cout << 6 << endl;
12     cout << 7 << endl;
13     cout << 8 << endl;
14     cout << 9 << endl;
15     cout << 10 << endl;
16     return 0;
17 }
```


如果今天想要列出1~10間的整數？

```
1  #inc
2  usin
3
4  int
5  {
6
7
8
9
```



```
1
2
3
4
5
6
7
8
9
10

Process returned 0 (0x0)    execution time : 0.144 s
Press any key to continue.
```

如果今天想列出1~1000間的整數呢？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << 1 << endl;
7      cout << 2 << endl;
8      cout << 3 << endl;
9      cout << 4 << endl;
10     cout << 5 << endl;
11     cout << 6 << endl;
12     cout << 7 << endl;
13     cout << 8 << endl;
14     cout << 9 << endl;
15     cout << 10 << endl;
16     cout << 11 << endl;
17     cout << 12 << endl;
18     cout << 13 << endl;
19     cout << 14 << endl;
20     cout << 15 << endl;
21     cout << 16 << endl;
```

以下省略...

如果今天想列出1~1000間的整數呢？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6
7
8
9
10
11      cout << 6 << endl;
12      cout << 7 << endl;
13      cout << 8 << endl;
14      cout << 9 << endl;
15      cout << 10 << endl;
16      cout << 11 << endl;
17      cout << 12 << endl;
18      cout << 13 << endl;
19      cout << 14 << endl;
20      cout << 15 << endl;
21      cout << 16 << endl;
```

這樣效率也太低了吧！

有沒有效率更高的方法呢？

迴圈介紹

迴圈(loop)是電腦科學運算領域的用語，也是一種常見的控制流程。迴圈是一段在程式中只出現一次，但可能會連續執行多次的程式碼。迴圈中的程式碼會執行特定的次數，或者是執行到特定條件成立時結束迴圈，或者是針對某一集合中的所有項目都執行一次。

有了迴圈，我們可以讓一段程式
區塊依照特定的條件重複執行

Source: 維基百科 – 迴圈

C++中的迴圈

在C++中，迴圈有3種：

- **while** 迴圈 (**while** loop)
- **do-while** 迴圈 (**do-while** loop)
- **for** 迴圈 (**for** loop)

while 迴圈

```
while(條件式(布林值)){  
    程式區塊;  
}
```

當條件式成立時，就執行程式區塊

-> 當條件式成立時，就執行程式區塊

-> 當條件式成立時，就執行程式區塊

-> 當條件式成立時，就執行程式區塊

-> 當條件式成立時，就執行程式區塊

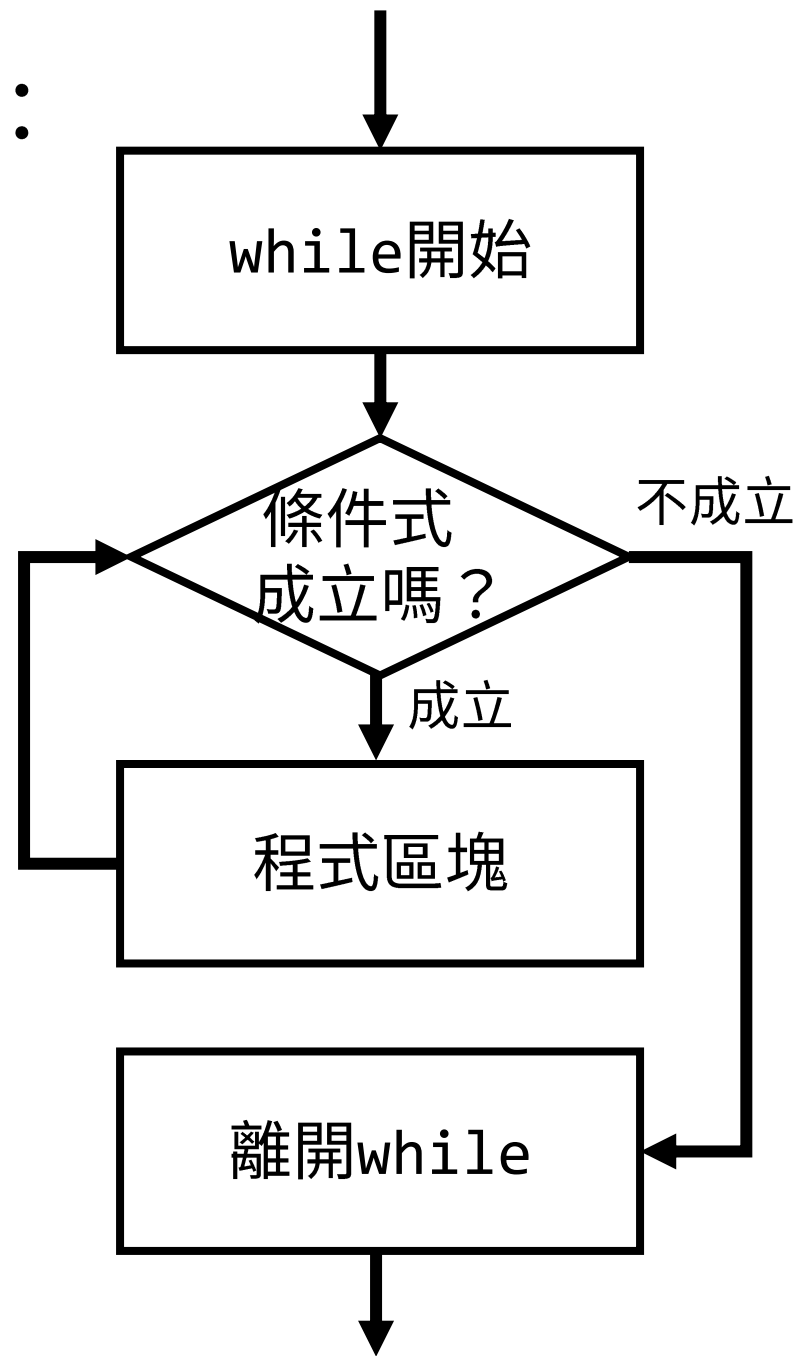
...

-> 當條件式不成立時，離開while迴圈

while 迴圈

```
while(條件式(布林值)){  
    程式區塊;  
}
```

流程圖：



while 迴圈

舉例來說：

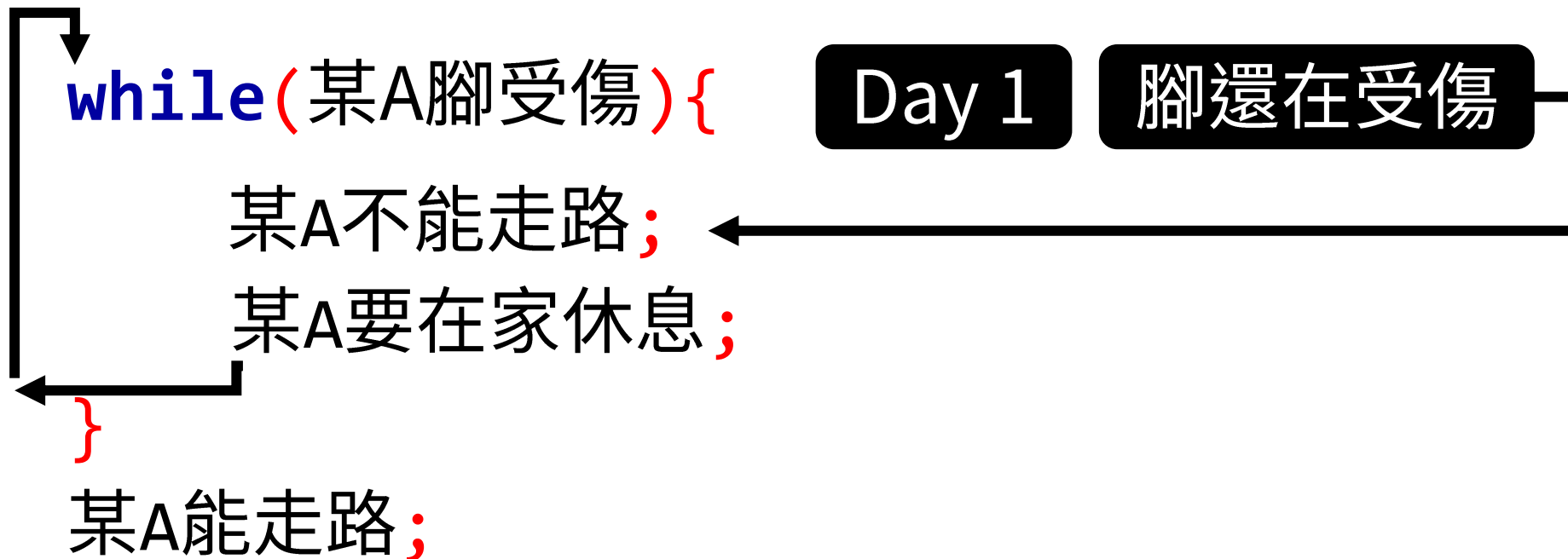
某A腳受傷，需要在家休息，傷好了才能走路

```
while(某A腳受傷){  
    某A不能走路;  
    某A要在家休息;  
}  
某A能走路;
```


while 迴圈

舉例來說：

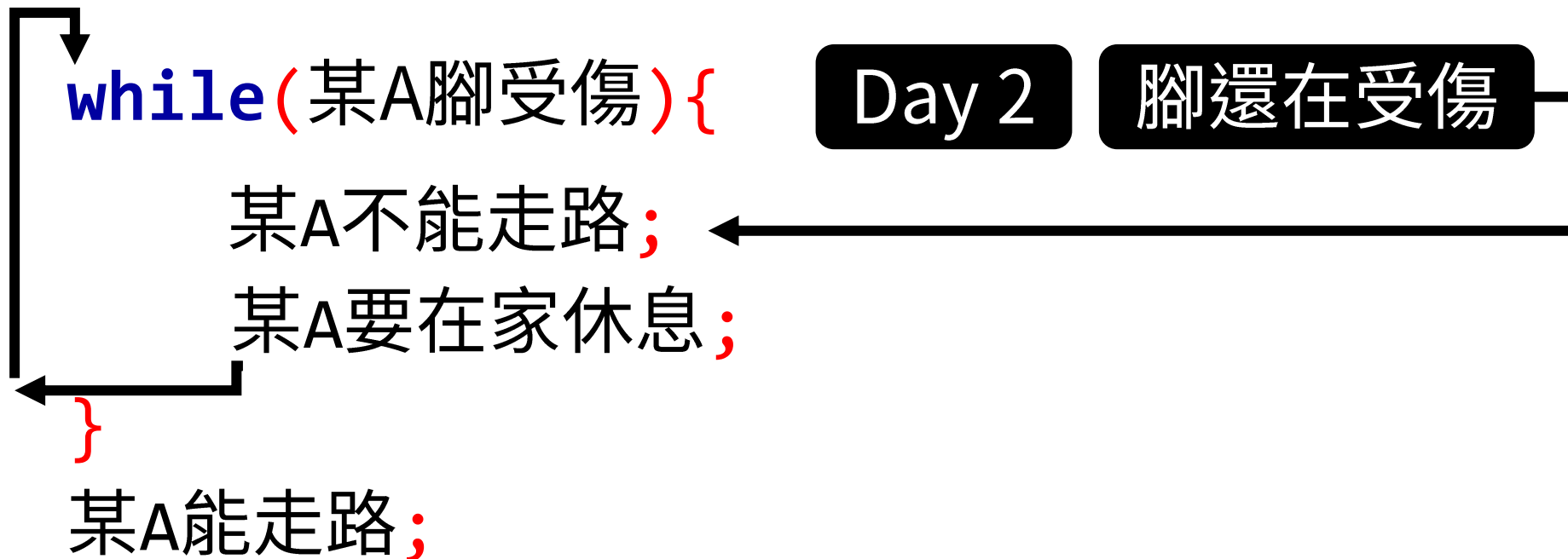
某A腳受傷，需要在家休息，傷好了才能走路



while 迴圈

舉例來說：

某A腳受傷，需要在家休息，傷好了才能走路



while 迴圈

舉例來說：

某A腳受傷，需要在家休息，傷好了才能走路

```
while(某A腳受傷){  
    某A不能走路;  
    某A要在家休息;  
}
```

Day 3

腳傷好了

離開while迴圈

某A能走路;



while 迴圈

舉例來說：

某A腳受傷，需要在家休息，傷好了才能走路

```
while(某A腳受傷){  
    某A不能走路;  
    某A要在家休息;  
}
```

Day 3

腳傷好了

離開while迴圈

某A能走路;

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 1

輸出：

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 1

i <= 10 為真

輸出：

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 1

i <= 10 為真

輸出 i

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 2

i <= 10 為真

輸出 i

i = i + 1

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 2

i <= 10 為真

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 2

i <= 10 為真

輸出 i

輸出：

1

2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 3

i <= 10 為真

輸出 i

i = i + 1

輸出：

1

2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 3

i <= 10 為真

輸出：

1

2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 3

i <= 10 為真

輸出 i

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 4

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 4

i <= 10 為真

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 4

i <= 10 為真

輸出 i

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 5

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 5

i <= 10 為真

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 5

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 6

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 6

i <= 10 為真

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 6

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 7

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 7

i <= 10 為真

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 7

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 8

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 8

i <= 10 為真

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 8

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 9

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 9

i <= 10 為真

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 9

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 10

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 10

i <= 10 為真

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 10

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 11

i <= 10 為真

輸出 i

i = i + 1

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 11

i <= 10 為非

離開while迴圈

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```

i = 11

i <= 10 為非

離開while迴圈

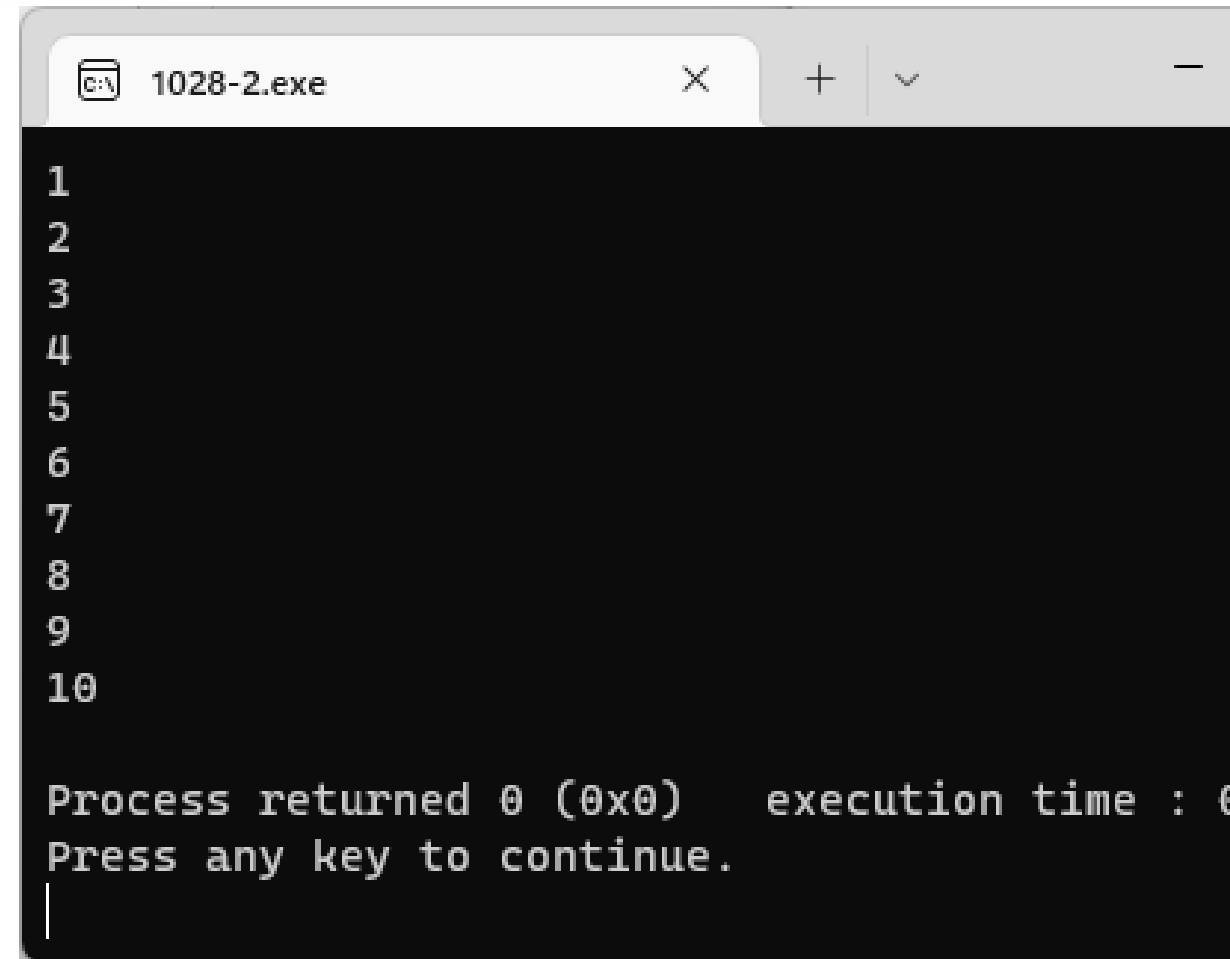
程式結束

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i = 1;
6      while(i <= 10){
7          cout << i << endl;
8          i++;
9      }
10     return 0;
11 }
```



```
1028-2.exe
1
2
3
4
5
6
7
8
9
10

Process returned 0 (0x0)   execution time : 0
Press any key to continue.
|
```

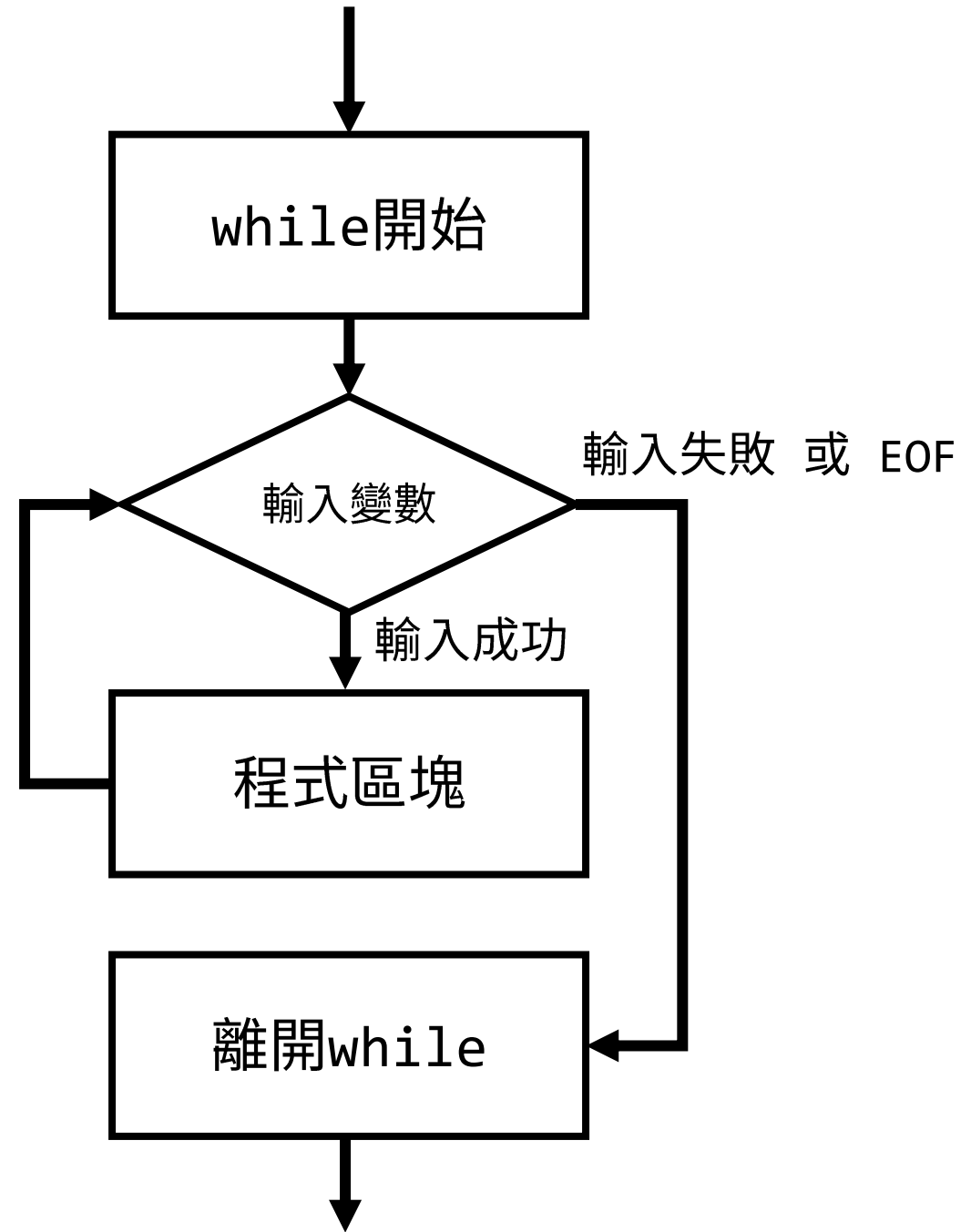
while 重複輸入

```
while(cin >> 變數){  
    程式區塊;  
}
```

EOF (End of File) 檔案結尾

顧名思義，就是檔案的結尾

在 Windows 上可以利用 Ctrl-Z 輸入
在 Linux 上可以利用 Ctrl-D 輸入



範例：累加計算

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i, sum = 0;
6      宣告整數 i 記錄輸入的數字
7      整數 sum 記錄累加的數值
8      cout << sum << " + ";
9      先輸出未加過的累加數值
```

```
9
10     while(cin >> i){
11         sum += i; sum = sum + i
12         cout << "= " << sum << "\n\n";
13         cout << sum << " + ";
14     }
15     return 0;
16 }
```

輸入要加的數值到 i 中

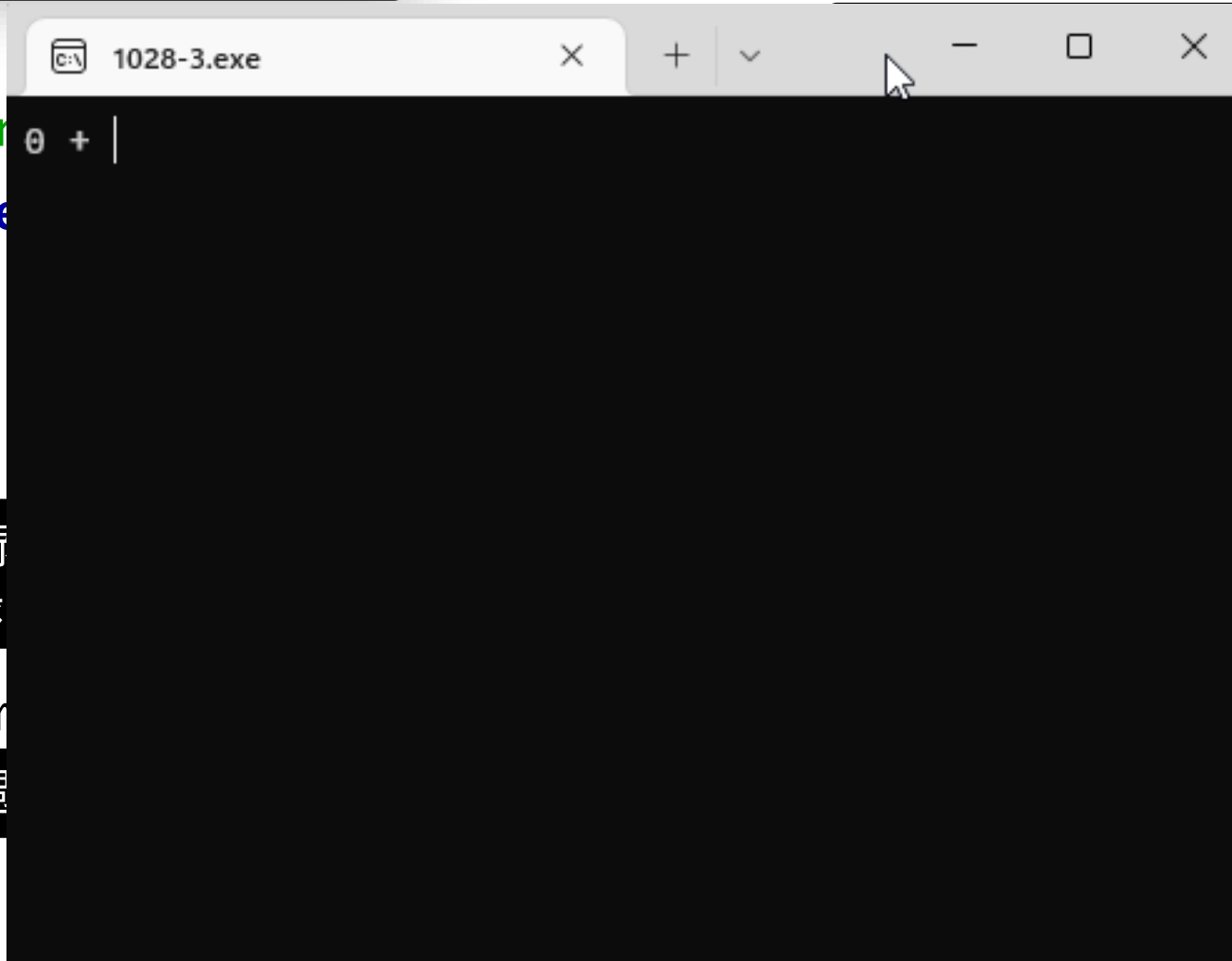
輸出累加後的結果

範例：累加計算

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int i, sum;
6
7
8      cout << sum;
```

宣告整數 i 記錄
整數 sum 記錄

先輸出未加過



i 中

um + i

sum << "\n\n";

+ ";

輸出累加後的結果

do-while 迴圈

與while迴圈類似，
不過會先執行程式
區塊再檢查條件式

```
do{  
    程式區塊;  
} while(條件式(布林值));
```

do-while 迴圈

與while迴圈類似，
不過會先執行程式
區塊再檢查條件式

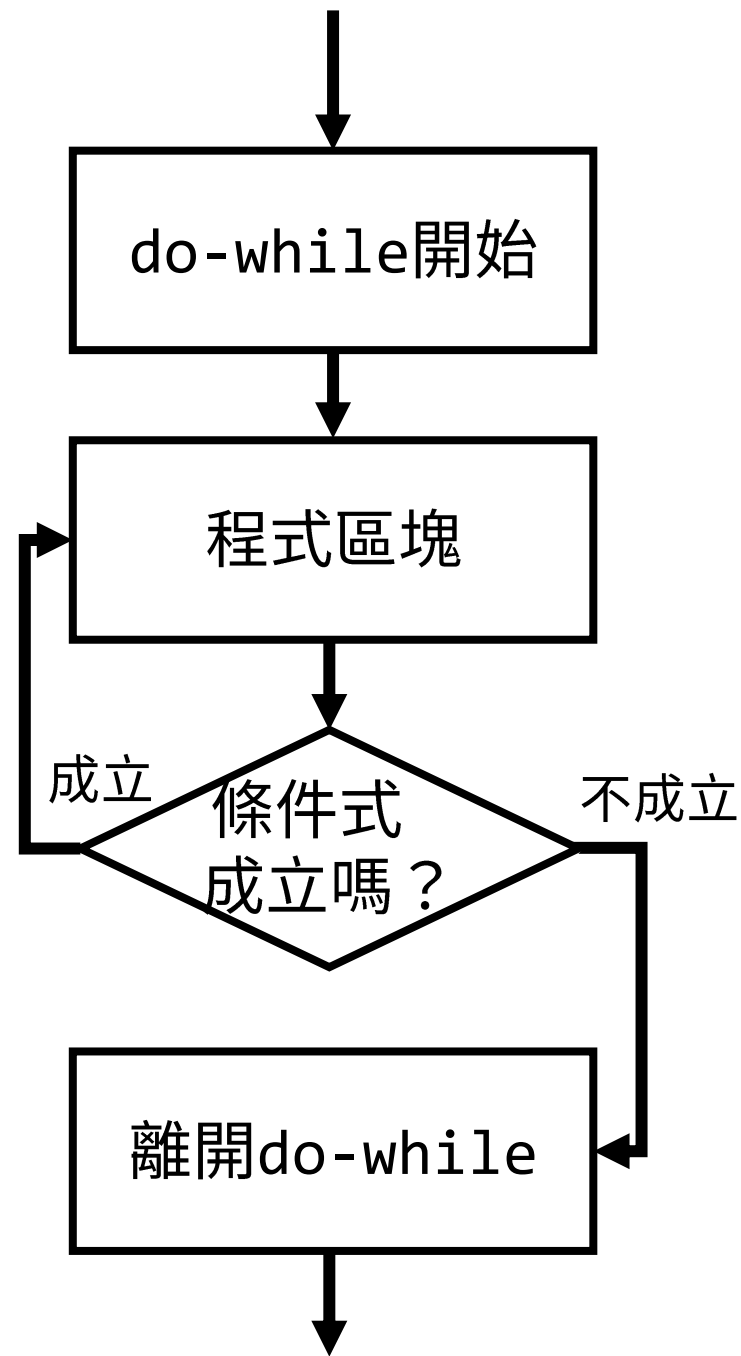
```
do{  
    程式區塊;  
} while(條件式(布林值));
```

要加；

do-while 迴圈

```
do{  
    程式區塊;  
} while(條件式(布林值));
```

流程圖：



for 迴圈

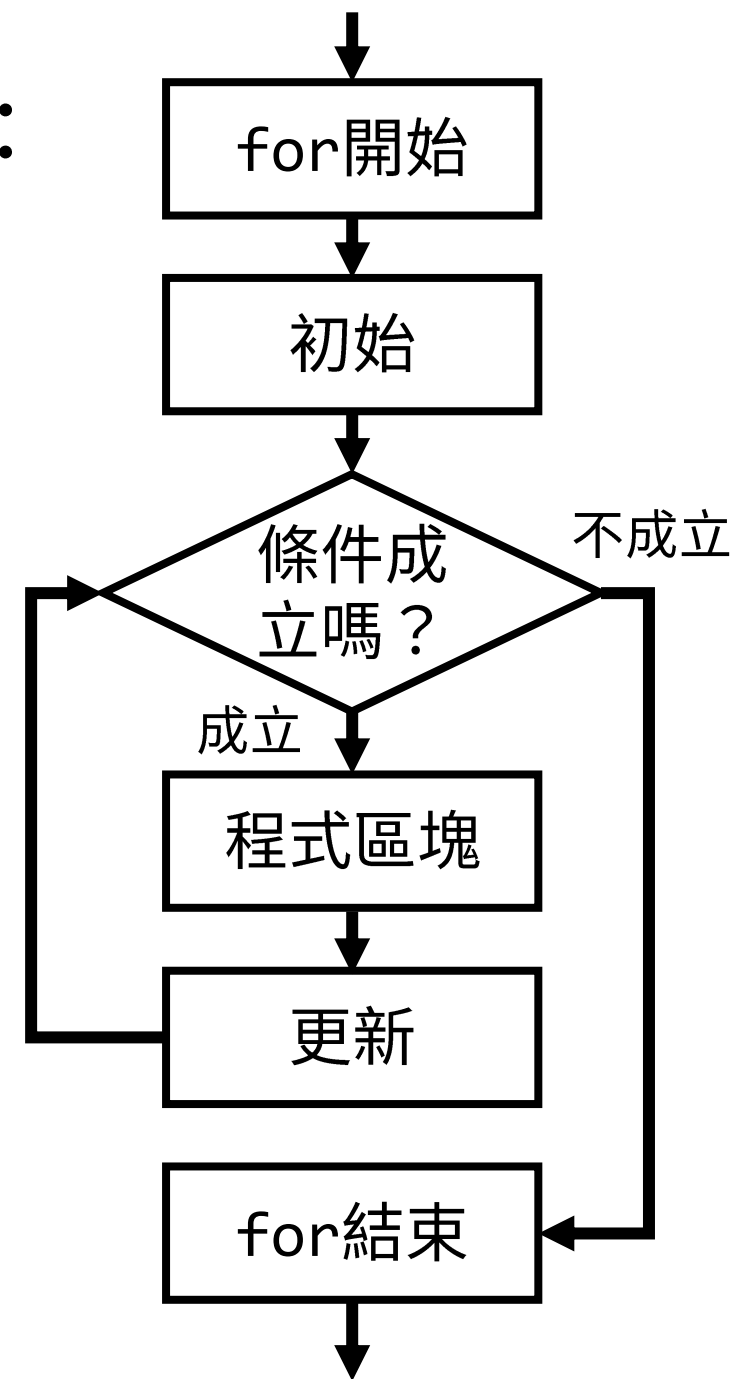
```
for (初始; 條件; 更新){  
    程式區塊;  
}
```

初始：只會執行一次的程式

條件：繼續執行迴圈的判斷依據

更新：執行完程式區塊後執行的程式

流程圖：



如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

宣告整數 i = 1

輸出：

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

宣告整數 i = 1

i = 1

輸出：

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 1

i <= 10 為真

輸出：

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 1

i <= 10 為真

輸出 i

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 2

i <= 10 為真

i = i + 1

輸出 i

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為真

i = 2

輸出：
1

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為真

i = 2

輸出 i

輸出：

1
2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 3

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 3

i <= 10 為真

輸出：

1
2

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為真

i = 3

輸出 i

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 4

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 4

i <= 10 為真

輸出：

1
2
3

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 4

i <= 10 為真

輸出 i

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 5

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 5

i <= 10 為真

輸出：

1
2
3
4

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 5

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 6

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 6

i <= 10 為真

輸出：

1
2
3
4
5

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 6

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 7

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 7

i <= 10 為真

輸出：

1
2
3
4
5
6

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 7

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 8

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 8

i <= 10 為真

輸出：

1
2
3
4
5
6
7

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 8

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 9

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 9

i <= 10 為真

輸出：

1
2
3
4
5
6
7
8

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為真

i = 9

輸出 i

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 10

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 10

i <= 10 為真

輸出：

1
2
3
4
5
6
7
8
9

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 10

i <= 10 為真

輸出 i

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 11

i <= 10 為真

i = i + 1

輸出 i

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為非

i = 11

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i = 11

i <= 10 為非

跳出迴圈

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```

i <= 10 為非

i = 11

跳出迴圈

程式結束

輸出：

1
2
3
4
5
6
7
8
9
10

如果今天想要列出1~10間的整數？

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for (int i = 1; i <= 10; i++){
7          cout << i << endl;
8      }
9      return 0;
10 }
```



1028-4.exe

```
1
2
3
4
5
6
7
8
9
10
```

Process return
Press any key

```
|
```

如果今天想要畫一張 $n*m$ 的乘法表？

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

```
9
10
11
12
13
14
15
16
17 }
```

```
    for (int i = 1; i <= n; i++){
        for (int j = 1; j <= m; j++){
            cout << i << "*" << j << "=" << i*j << "\t";
        }
        cout << "\n";
    }

    return 0;
```

利用2層for迴圈來排列組合出乘法表

換行

如果今天想要畫一張 $n*m$ 的乘法表？

```
1  #include <iostream>          9
2                                10
3  using namespace std;         11
4                                12
5  int main()                   13
6  {                             14
7      int n, m;                 15
8      cin >> n >> m;           16
9                                17
10                               for (int i = 1; i <= n; i++){
11                               for (int j = 1; j <= m; j++){
12                               cout << i << "*" << j << "=" << i*j << "\t";
13                               }
14                               cout << "\n";
15                               }
16                               return 0;
17                               }
```

假設 $n = 2, m = 3$

輸出：

如果今天想要畫一張 $n*m$ 的乘法表？

宣告整數 $i = 1$

$i = 1$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;        10         for (int j = 1; j <= m; j++){
4                                11             cout << i << "*" << j << "=" << i*j << "\t";
5  int main()                  12         }
6  {                            13         cout << "\n";
7                                14     }
8      int n, m;                15
9                                16     return 0;
10                               17 }
11
12
13
14
15
16
17
```

假設 $n = 2, m = 3$

輸出：

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1$

$i \leq n$ 為真

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;         10         for (int j = 1; j <= m; j++){
4                                     11             cout << i << "*" << j << "=" << i*j << "\t";
5  int main()                    12         }
6  {                              13         cout << "\n";
7      int n, m;                 14     }
8      cin >> n >> m;           15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：

如果今天想要畫一張 $n*m$ 的乘法表？

$i \leq n$ 為真

$i = 1, j = 1$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

宣告整數j = 1

假設 $n = 2, m = 3$

輸出：

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 1$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;         10         for (int j = 1; j <= m; j++){
4
5  int main()                   11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12         }
7      int n, m;                 13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 1$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j = i*j$

輸出：
 $1*1=1$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 2$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：
 $1*1=1$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 2$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;         10         for (int j = 1; j <= m; j++){
4
5  int main()                   11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12         }
7      int n, m;                 13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：
 $1*1=1$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 2$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j=i*j$

輸出：

$1*1=1$ $1*2=2$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：
 $1*1=1$ $1*2=2$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j=i*j$

輸出：
 $1*1=1$ $1*2=2$ $1*3=3$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1, j = 4$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2 1*3=3

如果今天想要畫一張 $n*m$ 的乘法表？

$i \leq n$ 為真

$i = 1, j = 4$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;         10         for (int j = 1; j <= m; j++){
4
5  int main()                   11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12         }
7      int n, m;                 13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：

$1*1=1$ $1*2=2$ $1*3=3$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1$

$i \leq n$ 為真

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

$j \leq m$ 為非

跳出迴圈

假設 $n = 2, m = 3$

輸出：

$1*1=1$ $1*2=2$ $1*3=3$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 1$

```
1  #include <iostream>          9  for (int i = 1; i <= n; i++){
2
3  using namespace std;         10
4                                11      for (int j = 1; j <= m; j++){
5  int main()                   12          cout << i << "*" << j << "=" << i*j << "\t";
6  {                             13      }
7      int n, m;                 14
8      cin >> n >> m;           15
                                16      cout << "\n";
                                17  }
                                18  return 0;
```

$i \leq n$ 為真

$j \leq m$ 為非

跳出迴圈

換行

假設 $n = 2, m = 3$

輸出：

$1*1=1$ $1*2=2$ $1*3=3$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;         10         for (int j = 1; j <= m; j++){
4
5  int main()                   11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12         }
7      int n, m;                 13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2 1*3=3

如果今天想要畫一張 $n*m$ 的乘法表？

$i \leq n$ 為真

$i = 2, j = 1$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17 }
```

宣告整數 $j = 1$

輸出：

$1*1=1$ $1*2=2$ $1*3=3$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 1$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;        10         for (int j = 1; j <= m; j++){
4
5  int main()                  11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                            12         }
7      int n, m;                13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2 1*3=3

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 1$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j=i*j$

輸出：

$1*1=1$ $1*2=2$ $1*3=3$
 $2*1=2$

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 2$

```
1  #include <iostream>          9      for (int i = 1; i <= n; i++){
2
3  using namespace std;        10         for (int j = 1; j <= m; j++){
4
5  int main()                  11             cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12         }
7      int n, m;                13         cout << "\n";
8      cin >> n >> m;           14     }
                                15
                                16     return 0;
                                17 }
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2 1*3=3
2*1=1

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 2$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：
1*1=1 1*2=2 1*3=3
2*1=1

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 2$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j = i*j$

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4
```

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4
```

如果今天想要畫一張 $n*m$ 的乘法表？

$i \leq n$ 為真

$i = 2, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4
```

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 3$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

輸出 $i*j = i*j$

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 4$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
16 }
17
```

假設 $n = 2, m = 3$

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張 $n*m$ 的乘法表？

$i = 2, j = 4$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10
11          for (int j = 1; j <= m; j++){
12              cout << i << "*" << j << "=" << i*j << "\t";
13          }
14          cout << "\n";
15      }
16      return 0;
17 }
```

$i \leq n$ 為真

$j \leq m$ 為真

跳出迴圈

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張n*m的乘法表？

i = 2

```
1  #include <iostream>          9  for (int i = 1; i <= n; i++){
2
3  using namespace std;         10     for (int j = 1; j <= m; j++){
4
5  int main()                   11         cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12     }
7      int n, m;                 13     cout << "\n";
8      cin >> n >> m;           14 }
                                15
                                16 return 0;
                                17 }
```

i <= n 為真

j <= m 為非

跳出迴圈

換行

假設n = 2, m = 3

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```


如果今天想要畫一張 $n*m$ 的乘法表？

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15
16     return 0;
17 }
```

$i \leq n$ 為真

$i++$

$i = 3$

$j \leq m$ 為非

跳出迴圈

換行

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張n*m的乘法表？

i = 3

```
1  #include <iostream>          9  for (int i = 1; i <= n; i++){
2
3  using namespace std;         10     for (int j = 1; j <= m; j++){
4
5  int main()                   11         cout << i << "*" << j << "=" << i*j << "\t";
6  {                             12     }
7      int n, m;                 13     cout << "\n";
8      cin >> n >> m;           14 }
                                15
                                16 return 0;
                                17 }
```

假設n = 2, m = 3

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張 $n*m$ 的乘法表？

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設 $n = 2, m = 3$

```
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15     return 0;
17 }
```

$i \leq n$ 為非

$i++$

$j \leq m$ 為非

跳出迴圈

換行

跳出迴圈

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```

如果今天想要畫一張n*m的乘法表？

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      cin >> n >> m;
```

假設n = 2, m = 3

```
9      for (int i = 1; i <= n; i++){
10         for (int j = 1; j <= m; j++){
11             cout << i << "*" << j << "=" << i*j << "\t";
12         }
13         cout << "\n";
14     }
15
16     return 0;
```

i <= n 為非

i++

j <= m 為非

跳出迴圈

換行

程式結束

輸出：

```
1*1=1  1*2=2  1*3=3
2*1=1  2*2=4  2*3=6
```