

練習寫程式時的要求

NTOUCSE

丁培毅

怎麼說這個?! 一板一眼的

怎麼說這個?! 一板一眼的

- 三十幾年來, 這是一個成功率不高的課程!!



怎麼說這個?! 一板一眼的

- 三十幾年來, 這是一個成功率不高的課程!!

- 台灣資訊產業的未來在哪裡??



怎麼說這個?! 一板一眼的

- 三十幾年來, 這是一個成功率不高的課程!!



- 台灣資訊產業的未來在哪裡??
- 台灣已經沒有人才可以浪費囉!!



練習寫程式時的基本要求

- 要求你要做到:

練習寫程式時的基本要求

- 要求你要做到:
「看過/分析過/想過 問題和範例程式,

練習寫程式時的基本要求

- 要求你要做到:
「看過/分析過/想過 問題和範例程式,
遮起來, 在電腦上打出來

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

練習寫程式時的基本要求

- 要求你要做到:
 - 「看過/分析過/想過 問題和範例程式,
遮起來, 在電腦上打出來
測試執行, 觀察結果, 分析整理與修改」
- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

只學到了怎樣使用 鍵盤+滑鼠 操作!!

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

只學到了怎樣使用 鍵盤+滑鼠 操作!!

錯了 $\Rightarrow$

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

只學到了怎樣使用 鍵盤+滑鼠 操作!!

錯了 $\Rightarrow$ { 拷貝別的

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

只學到了怎樣使用 鍵盤+滑鼠 操作!!

錯了 $\Rightarrow$ { 拷貝別的
隨便改改

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- 不是 拷貝貼上 $\Rightarrow$ 測試 沒有錯 $\Rightarrow$ 完成

只學到了怎樣使用 鍵盤+滑鼠 操作!!

錯了 $\Rightarrow$ { 拷貝別的
隨便改改 $\Rightarrow$ 測試, 說不定錯誤會不見

練習寫程式時的基本要求

- 要求你要做到:

「看過/分析過/想過 問題和範例程式,

遮起來, 在電腦上打出來

測試執行, 觀察結果, 分析整理與修改」

- ~~• 不是 拷貝貼上 ⇒ 測試 沒有錯 ⇒ 完成
只學到了怎樣使用 鍵盤+滑鼠 操作!!
錯了 ⇒ { 拷貝別的
隨便改改 ⇒ 測試, 說不定錯誤會不見~~


```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

```

#include <stdio.h>
#include <stdlib.h>
int main() {
    int i,n,j;
    const char* const day[12]={"first","second","third","fourth",
                                "fifth","sixth","seventh","eighth",
                                "ninth","tenth","eleventh","twelfth"};
    const char* const line[11]={"2 Turtle Doves","3 French Hens",
                                "4 Calling Birds","5 Golden Rings","6 Geese a Laying",
                                "7 Swans a Swimming","8 Maids a Milking","9 Ladies Dancing",
                                "10 Lords a Leaping","11 Pipers Piping","12 Drummers Drumming"};
    scanf("%d",&n);
    for (i=0;i<n;i++) {
        if (i>0) printf("\n");
        printf("On the %s day of Christmas\n", day[i]);
        printf("my true love sent to me:\n");
        for (j=12-i; j<12; j++) printf("%s\n", line[11-j]);
        if(i==0) printf("A"); else printf("and a");
        printf(" Partridge in a Pear Tree\n");
    }
    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#define MAXNUMBER 20
void rotate(int size, char x[]);
int rotate_n_check(int size, char x[]);
void printPermutation(int size, char data[]);
int checkOutOfOrderCountsK(int size, char data[], int K);

int main(void)
{
    int i, num = -1, K = -1;
    char digit[MAXNUMBER];
    scanf("%d%d", &num,&K);
    for (i=0; i<num; i++) digit[i] = i;
    do
        if (checkOutOfOrderCountsK(num, digit, K))
            printPermutation(num, digit);
    while (rotate_n_check(num, digit));
    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#define MAXN 1000000
void rotate(int size, char x[])
void rotate(int size, char x[])
{
    int i;
    char tmp = x[size-1];
    for (i=size-1; i>0; i--)
        x[i] = x[i-1];
    x[0] = tmp;
}
int main(void)
{
    int i, num = 0;
    char digit[MAXN];
    scanf("%d", &num);
    for (i=0; i<num; i++)
    {
        do
        {
            if (checkOut(i, digit))
                printPe(i, digit);
            while (rotate(i, digit))
                return 0;
        }
        return 0;
    }
}

int rotate_n_check(int size, char data[])
{
    int i;
    for (i=size; i>=2; i--)
    {
        rotate(i, data);
        if (data[i-1] != i-1) return 1;
    }
    return 0;
}

```

```

#include <stdio.h>
#include <stdlib.h>
#define MAXN 1000000

void rotate(int size, char x[])
{
    int i;
    char tmp = x[size-1];
    for (i=size-1; i>0; i--)
        x[i] = x[i-1];
    x[0] = tmp;
}

void printPermutation(int size, char data[])
{
    static int count=0;
    int i;
    for (i=0; i<size-1; i++)
        printf("%d ", data[i]);
    printf("%d", data[size-1]);
    printf("\n");
}

int checkOutOfOrderCountsK(int size, char data[], int K)
{
    int i, j, count=0;
    for (i=0; i<size; i++)
        for (j=i+1; j<size; j++)
            if (data[i]>data[j])
                count++;
    return count%K==0;
}

int main(void)
{
    int i, num = 0;
    char digit[MAXN];
    scanf("%d", &num);
    for (i=0; i<num; i++)
    {
        do
        {
            if (checkOutOfOrderCountsK(i, digit, 2))
                printPermutation(i, digit);
            while (rotate(i, digit));
        } while (rotate(i, digit));
        return 0;
    }
}

```

- 什麼!!!

- 什麼!!!
- 不會吧!! 你認真??

- 什麼!!!
- 不會吧!! 你認真??

這在「背書」喔???

- 什麼!!!
- 不會吧!! 你認真??

這在「背書」喔???

這麼會背我就去唸 **一類** 的系了

說不定哪天你會在底下幫我喊「凍蒜」

- 不是啦!!!

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

```
#include <stdio.h>
```

```
int
```

```
    #include <stdio.h>
```

```
    int main() {
```

```
        printf("I am a novice to programming!\n");
```

```
        return 0;
```

```
    }
```

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

有些東西很自由, 可以變化
有些東西不可以亂改

```
#include <stdio.h>
```

```
int #include <stdio.h>
```

```
printf #include <stdio.h>
```

```
int main() {  
    printf("我是新手, 請多多指教!\n");  
    return 0;  
}
```

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

有些東西很自由, 可以變化
有些東西不可以亂改

```
#include <stdio.h>
int main() {
    printf("我是新手, 請多多指教!\n");
    return 0;
}
```

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

有些東西很自由, 可以變化
有些東西不可以亂改

```
#include <stdio.h>

int
p
re
}

#include <stdio.h>

int
p
re
}

#include <stdio.h>

int
p
re
}

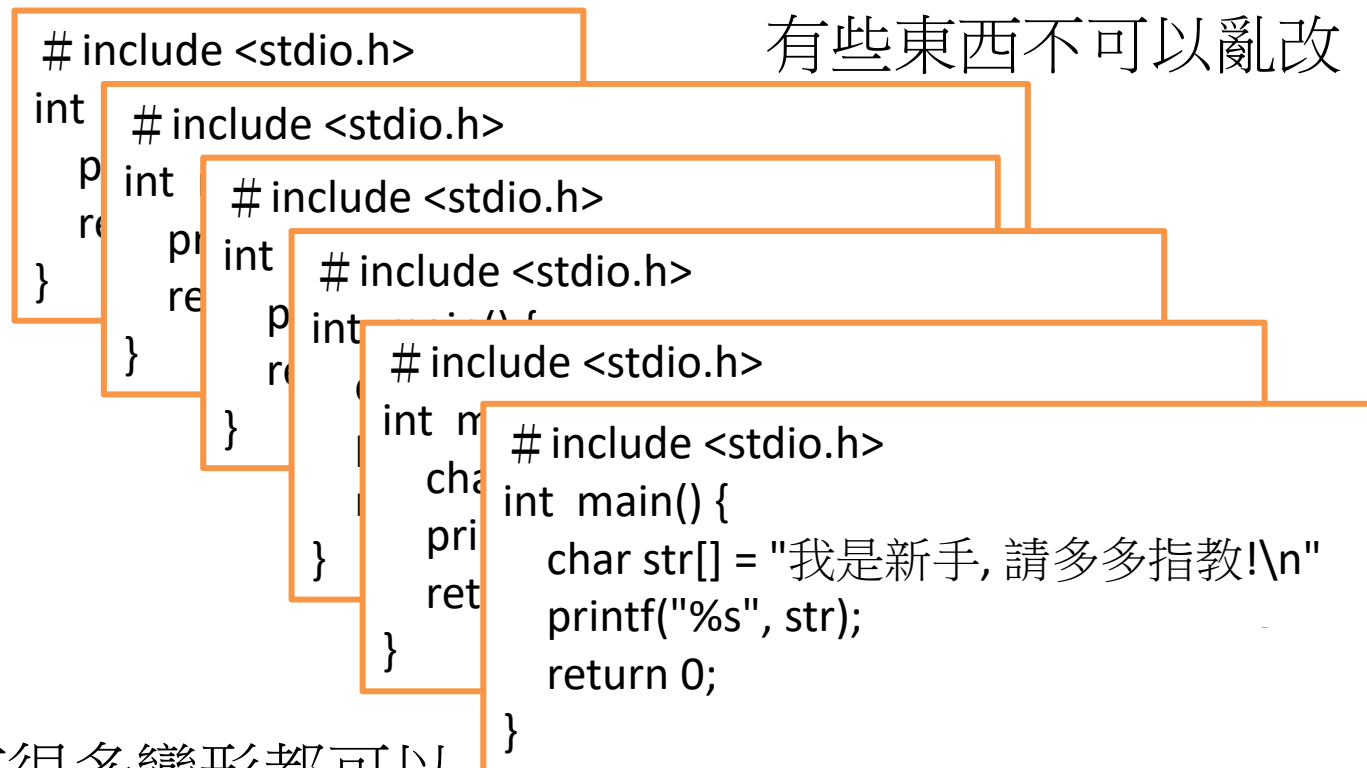
#include <stdio.h>

int
p
re
}

#include <stdio.h>
int main() {
    char str[] = "我是新手, 請多多指教!\\n"
    printf(str);
    return 0;
}
```


- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

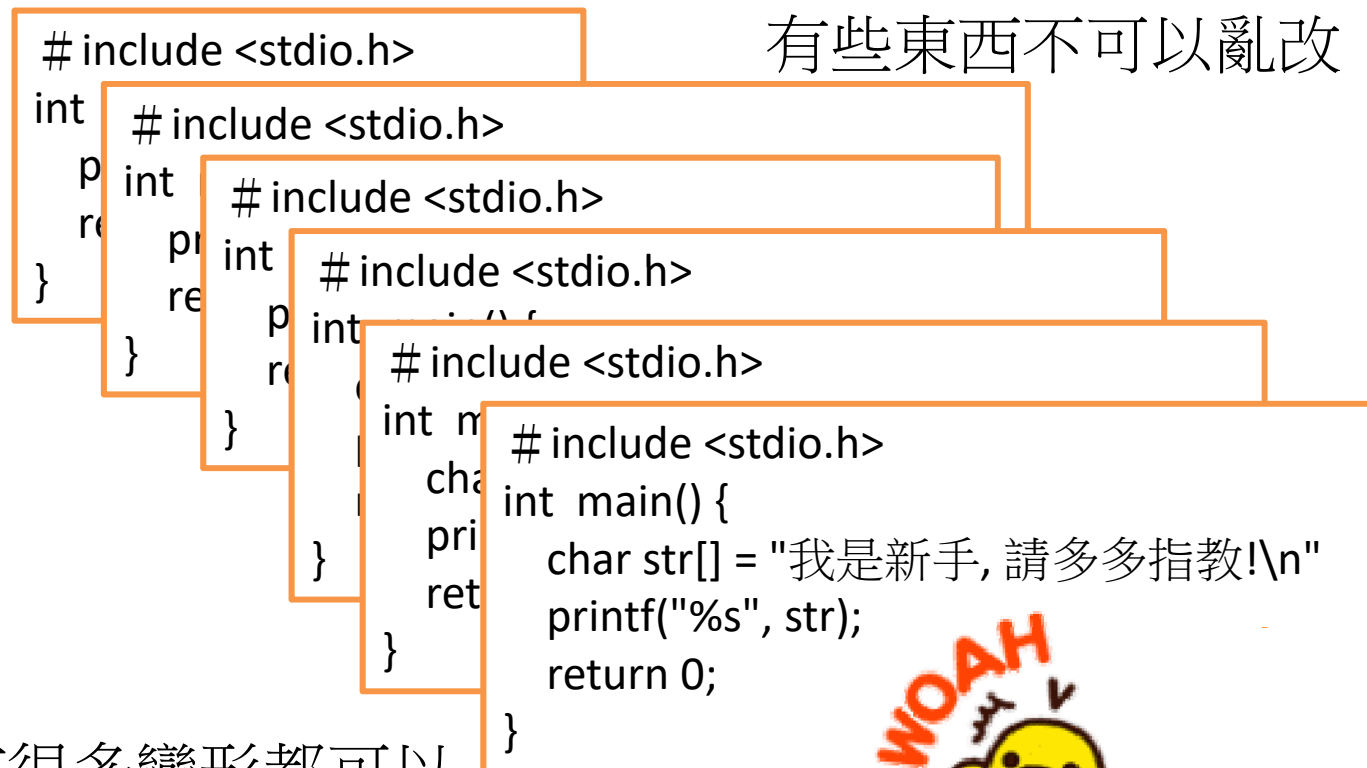
有些東西很自由, 可以變化
有些東西不可以亂改



有很多變形都可以
得到同樣的效果!!

- 不是啦!!!
- 把這些直統統地背下來也是沒有用的

有些東西很自由, 可以變化
有些東西不可以亂改



有很多變形都可以
得到同樣的效果!!



這麼多怎麼背
你玩我喔!!

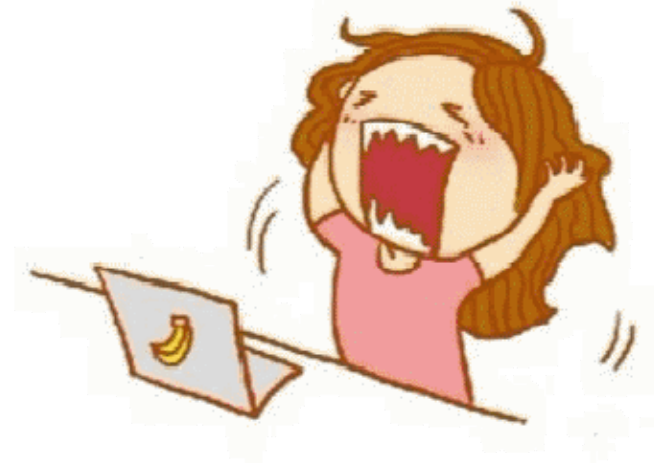
遮起來, 打出程式來...

遮起來, 打出程式來...

也盡量不要邊看邊打, 更不要直接拷貝

遮起來, 打出程式來...

也盡量不要邊看邊打, 更不要直接拷貝



這不是背下來是什麼??

這不是背下來是什麼??

這不是背下來是什麼??

- 什麼叫做 背

• 什麼叫做 背

白居易的「琵琶行」

潯陽江頭夜送客，楓葉荻花秋瑟瑟。
主人下馬客在船，舉酒欲飲無管弦；
醉不成歡慘將別，別時茫茫江浸月。
忽聞水上琵琶聲，主人忘歸客不發。
尋聲暗問彈者誰？琵琶聲停欲語遲。
移船相近邀相見，添酒回燈重開宴。
千呼萬喚始出來，猶抱琵琶半遮面。

...

今夜聞君琵琶語，如聽仙樂耳暫明。
莫辭更坐彈一曲，為君翻作琵琶行。
感我此言良久立，卻坐促弦弦轉急；
淒淒不似向前聲，滿座重聞皆掩泣。
座中泣下誰最多？江州司馬青衫溼。

• 什麼叫做 背

白居易的「琵琶行」

88句616字

潯陽江頭夜送客，楓葉荻花秋瑟瑟。
主人下馬客在船，舉酒欲飲無管弦；
醉不成歡慘將別，別時茫茫江浸月。
忽聞水上琵琶聲，主人忘歸客不發。
尋聲暗問彈者誰？琵琶聲停欲語遲。
移船相近邀相見，添酒回燈重開宴。
千呼萬喚始出來，猶抱琵琶半遮面。

...

今夜聞君琵琶語，如聽仙樂耳暫明。
莫辭更坐彈一曲，為君翻作琵琶行。
感我此言良久立，卻坐促弦弦轉急；
淒淒不似向前聲，滿座重聞皆掩泣。
座中泣下誰最多？江州司馬青衫溼。

• 什麼叫做 背

白居易的「琵琶行」

88句616字

潯陽江頭	江州司馬青衫溼	座中泣下誰最多
主人下馬	滿座重聞皆掩泣	淒淒不似向前聲
醉不成歡	卻坐促弦弦轉急	感我此言良久立
忽聞水上	為君翻作琵琶行	莫辭更坐彈一曲
尋聲暗問	如聽仙樂耳暫明	今夜聞君琵琶語
移船相邀	...	
千呼萬喚	猶抱琵琶半遮面	千呼萬喚始出來
	添酒回燈重開宴	移船相近邀相見
	琵琶聲停欲語遲	尋聲暗問彈者誰
今夜聞君	主人忘歸客不發	忽聞水上琵琶聲
莫辭更坐	別時茫茫江浸月	醉不成歡慘將別
感我此言	舉酒欲飲無管弦	主人下馬客在船
淒淒不似	楓葉荻花秋瑟瑟	潯陽江頭夜送客
座中泣下		

- 什麼叫做背

白居易的「琵琶行」

88句616字

[illegible]

考試型的學習

- 套公式就好

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

$$\frac{(4 + 50) 47}{2}$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

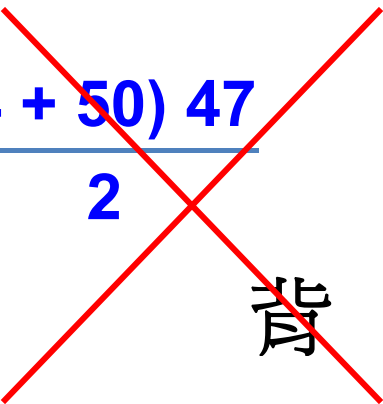
$$\frac{(4 + 50) \cdot 47}{2}$$

背

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$


$$\frac{(4 + 50) 47}{2}$$

背

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

$$\frac{(4 + 50) 47}{2}$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$4 + 5 + 6 + 7 + \dots + 50$

$$\frac{(4 + 50) 47}{2}$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{r} 4 + 5 + 6 + 7 + \dots + 50 \\ 50 + 49 + 48 + 47 + \dots + 4 \end{array}$$

$$\frac{(4 + 50) 47}{2}$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ \hline 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

└──┘
47 組

$$\frac{(4 + 50) 47}{2}$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

└──┘
47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

└──┘
47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$

考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

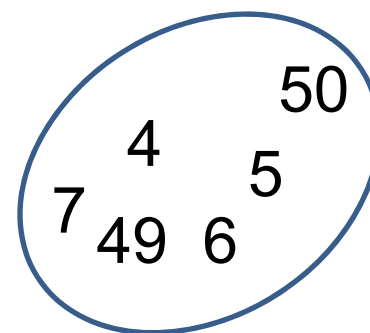
$$\begin{array}{ccccccc} 4 & + & 5 & + & 6 & + & 7 & + & \dots & + & 50 \\ 50 & + & 49 & + & 48 & + & 47 & + & \dots & + & 4 \\ \hline 54 & & 54 & & & & & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

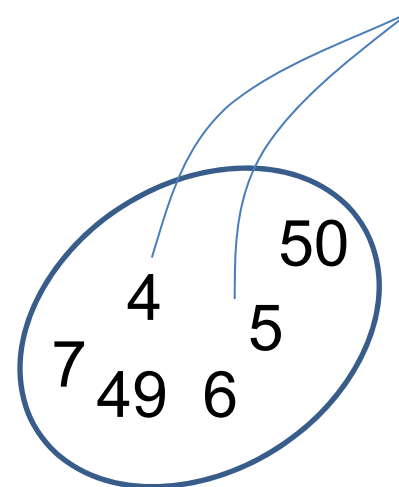
$$\begin{array}{ccccccc} 4 & + & 5 & + & 6 & + & 7 & + & \dots & + & 50 \\ 50 & + & 49 & + & 48 & + & 47 & + & \dots & + & 4 \\ \hline 54 & & 54 & & & & & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

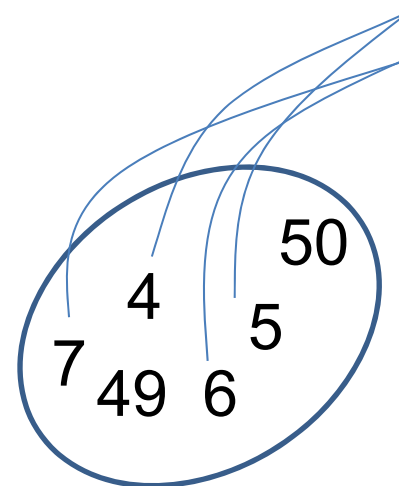
$$\begin{array}{ccccccc} 4 & + & 5 & + & 6 & + & 7 & + & \dots & + & 50 \\ 50 & + & 49 & + & 48 & + & 47 & + & \dots & + & 4 \\ \hline 54 & & 54 & & & & & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

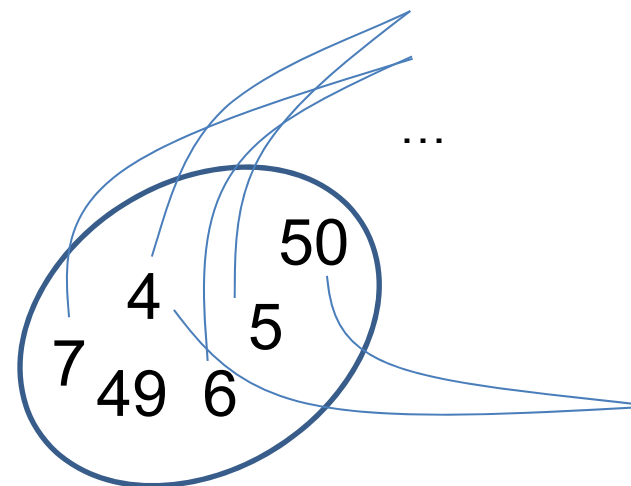
$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

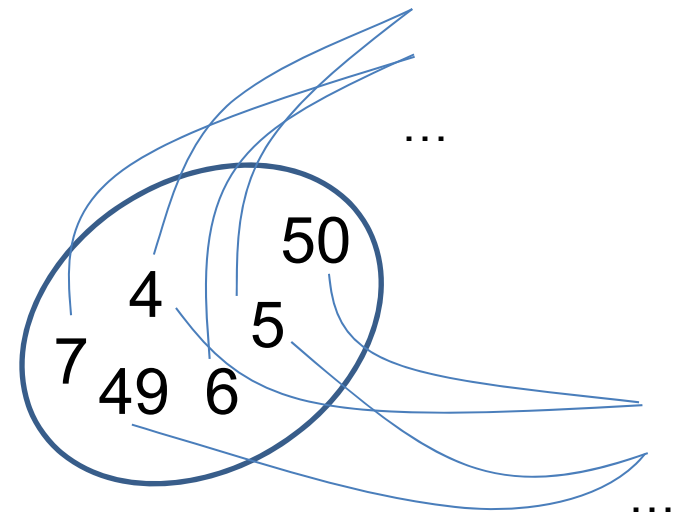
$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{ccccccc} 4 + & 5 + & 6 + & 7 + & \dots + & 50 \\ 50 + & 49 + & 48 + & 47 + & \dots + & 4 \\ \hline 54 & 54 & & & & 54 \end{array}$$

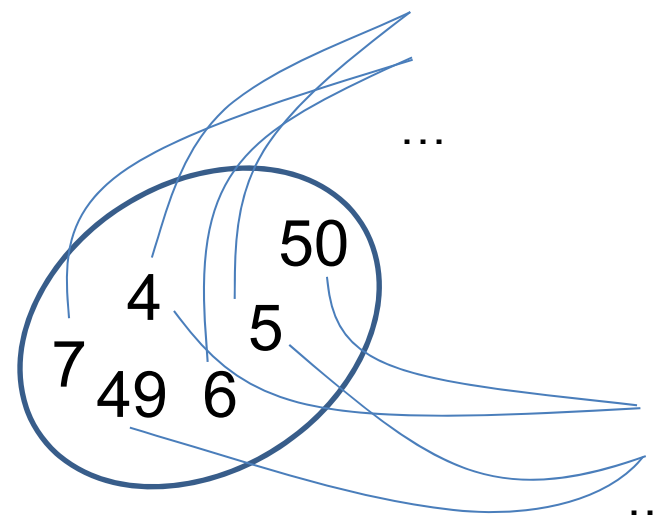
└──┘
47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$

為什麼要知道為什麼呢？



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

$$\begin{array}{ccccccccccc} 4 & + & 5 & + & 6 & + & 7 & + & \dots & + & 50 \\ 50 & + & 49 & + & 48 & + & 47 & + & \dots & + & 4 \\ \hline 54 & & 54 & & & & & & & & 54 \end{array}$$

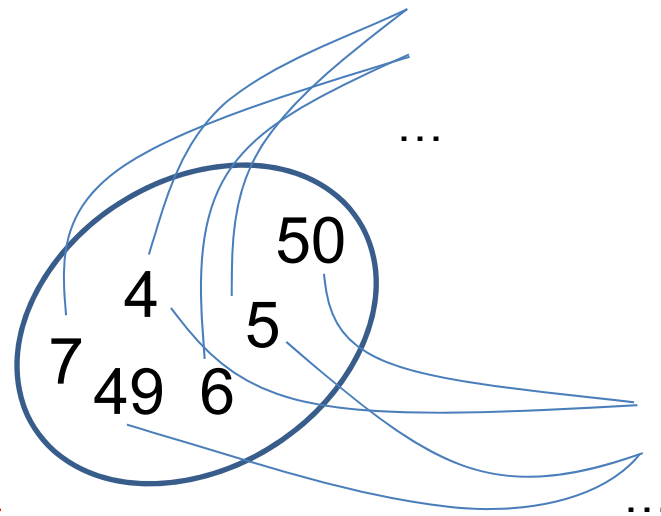
47 組

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$

為什麼要知道為什麼呢？ 對付各種變化

$$\frac{(4 + 50) 47}{2}$$



考試型的學習

- 套公式就好

計算 $4 + 5 + 6 + 7 + \dots + 50$

為什麼？

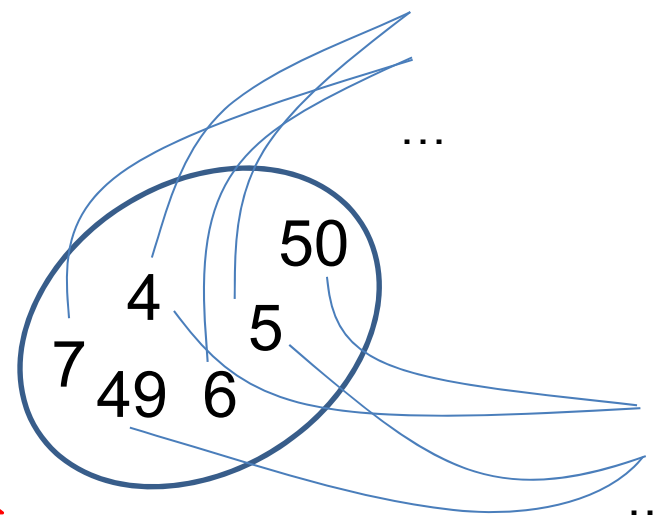
$$\begin{array}{ccccccc} 4 & + & 5 & + & 6 & + & 7 & + & \dots & + & 50 \\ 50 & + & 49 & + & 48 & + & 47 & + & \dots & + & 4 \\ \hline 54 & & 54 & & & & & & & & 54 \end{array}$$

47 組

$$\frac{(4 + 50) 47}{2}$$

怎麼會有怪物這麼想呢？

$$4 + 5 + 6 = 4 + 6 + 5$$



為什麼要知道為什麼呢？ 對付各種變化

常常練習，就算沒有人告訴你步驟，自己整理出來也不是太困難

寫程式是在說故事、在講話、
用固定格式的指令操控機器做許多事情

寫程式是在說故事、在講話、

用固定格式的指令操控機器做許多事情

重點是你想講什麼、你要叫電腦做什麼?

寫程式是在說故事、在講話、

用固定格式的指令**操控機器**做許多事情

重點是你**想講什麼**、你要**叫電腦做什麼?**

如果你是公司老闆, 永遠搞不清楚下屬名字和

部門的分工, 沒有名單就無法指揮, 這樣會很好笑,

沒有什麼說服力吧!! 很難專心在公司營運目標吧!!

寫程式是在說故事、在講話、

用固定格式的指令**操控機器**做許多事情

重點是你**想講什麼**、你要**叫電腦做什麼?**

如果你是公司老闆, 永遠搞不清楚下屬名字和

部門的分工, 沒有名單就無法指揮, 這樣會很好笑,

沒有什麼說服力吧!! 很難專心在公司營運目標吧!!

練習寫程式的時候, 除了格式要符合規定, 重點是**思考**
程式的內容, 一旦你照著資料抄, 甚至用複製貼上, 的確
可以省下很多時間, 但是常常也分不清楚什麼是格
式、什麼是內容了, 通常你只會記得跟誰抄的, 去哪
裡抄的...

寫程式是在說故事、在講話、

用固定格式的指令**操控機器**做許多事情

重點是你**想講什麼**、你要**叫電腦做什麼**？

如果你是公司老闆，永遠搞不清楚下屬名字和

部門的分工，沒有名單就無法指揮，這樣會很好笑，

沒有什麼說服力吧!! 很難專心在公司營運目標吧!!

練習寫程式的時候，除了格式要符合規定，重點是**思考**
程式的內容，一旦你照著資料抄，甚至用複製貼上，的
確可以省下很多時間，但是常常也分不清楚什麼是格
式、什麼是內容了，通常你只會記得跟誰抄的，去哪
裡抄的...

這樣很不好，你才剛進來

不小心就開始跟資訊業說 **byebye**

寫程式是在說故事、在講話、

用固定格式的指令**操控機器**做許多事情

重點是你**想講什麼**、你要**叫電腦做什麼**？

如果你是公司老闆，永遠搞不清楚下屬名字和

部門的分工，沒有名單就無法指揮，這樣會很好笑，

沒有什麼說服力吧!! 很難專心在公司營運目標吧!!

練習寫程式的時候，除了格式要符合規定，重點是**思考**
程式的內容，一旦你照著資料抄，甚至用複製貼上，的
確可以省下很多時間，但是常常也分不清楚什麼是格
式、什麼是內容了，通常你只會記得跟誰抄的，去哪
裡抄的...

這樣很不好，你才剛進來

不小心就開始跟資訊業說 **byebye**

- 程式有固定的格式: 每一段程式功能可以不同, 但是需要遵循一定的規則 – 有些部份是**規定**要有的

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

- 程式有固定的格式: 每一段程式功能可以不同, 但是需要遵循一定的規則 – 有些部份是**規定**要有的

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

- 像是英文裡面的語法: 主詞 + 動詞 + 受詞 + 副詞

- 程式有固定的格式: 每一段程式功能可以不同, 但是需要遵循一定的規則 – 有些部份是**規定**要有的

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

- 像是英文裡面的語法: 主詞 + 動詞 + 受詞 + 副詞
- 後續會逐步介紹 C 程式的語法

- 程式有固定的格式: 每一段程式功能可以不同, 但是需要遵循一定的規則 – 有些部份是**規定**要有的

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
    return 0;
}
```

- 像是英文裡面的語法: 主詞 + 動詞 + 受詞 + 副詞
- 後續會逐步介紹 C 程式的語法
- **不要過度擔心:** 人和機器溝通的語法比人和人溝通的語法**簡單得多**, **規則**少、沒有背不完的**單字**、也沒有因為習慣而產生許許多多的**例外**

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門
 - 長久 ... 練習英文打字吧! (不是高職生的專利)

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門
 - 長久 ... 練習英文打字吧! (不是高職生的專利)
 - 暫時 ... 一列一列拷貝 ... (認真的)

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門
 - 長久 ... 練習英文打字吧! (不是高職生的專利)
 - 暫時 ... 一列一列拷貝 ... (認真的)
你在心裡問自己: 接下來要寫什麼?

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門
 - 長久 ... 練習英文打字吧! (不是高職生的專利)
 - 暫時 ... 一系列拷貝 ... (認真的)
你在心裡問自己: 接下來要寫什麼?
是 int, 是 (, 是 }, 是 return, 還是其它敘述?

- 你需要練習到不需要查書、不需要查資料, 就可以正確使用基本的語法, 你的腦筋才能夠專注在該放什麼命令, 該以什麼順序來下達命令上面
- 這就是先想再打練習的主要目標
- **Q:** 重頭打一遍好浪費時間, 都在找按鍵在哪裡
這時候太想要節省時間, 又害你走不進資訊系的大門
 - 長久 ... 練習英文打字吧! (不是高職生的專利)
 - 暫時 ... 一系列拷貝 ... (認真的)
你在心裡問自己: 接下來要寫什麼?
是 int, 是 (, 是 }, 是 return, 還是其它敘述?
想清楚了, 確定是什麼了, 驗證一下, 才拷貝一系列

- Q: 看過範例程式了, 覺得再怎麼看也沒有幫助了

- Q: 看過範例程式了, 覺得再怎麼看也沒有幫助了

```
void bucketSort(int ndata, int data[]) {  
    int buckets[8][MAX], nBucket[8];  
    int i, j, k, index, mult=1, iBucket;  
    int len = maxNumDigits(ndata, data);  
    for (i=0; i<len; i++) {  
        for (j=0; j<8; j++) nBucket[j] = 0;  
        for (j=0; j<ndata; j++) {  
            iBucket = data[j] / mult % 8;  
            buckets[iBucket][nBucket[iBucket]++] = data[j];  
        }  
        index = 0;  
        for (j=0; j<8; j++)  
            for (k=0; k<nBucket[j]; k++)  
                data[index++] = buckets[j][k];  
        mult *= 8;  
    }  
}
```

- Q: 看過範例程式了, 覺得再怎麼看也沒有幫助了
 - 可是遮起來以後一點點都寫不出來

- Q: 看過範例程式了, 覺得再怎麼看也沒有幫助了
 - 可是遮起來以後一點點都寫不出來
 - 範例不好: 辨識不出它的架構

- **Q:** 看過範例程式了, 覺得再怎麼看也沒有幫助了
 - 可是遮起來以後一點點都寫不出來
 - 範例不好: 辨識不出它的架構

```
#include <stdio.h>
#define BING(x,y) ((x)<<y)
#define BANG(x) (1<<x)
#define BOOM    1
int main () {
    int x,y,z,w;
    int V[3] = {BING(x=227380393,BANG(BOOM)+BOOM), x+(w=BOOM+BANG(BANG(BOOM))),
                BING(47*(y=17453197),BOOM)), x+y+BING(w*w*17*185527,BANG(BOOM))};
    char *p=(char *)V;
    while(*(p-BOOM)!=BOOM+BING(BOOM,w)) putchar(*p++);
    return 0;
} /* Mind the comma operator! */
```

- **Q:** 看過範例程式了, 覺得再怎麼看也沒有幫助了
 - 可是遮起來以後一點點都寫不出來
 - 範例不好: 辨識不出它的架構

```
#include <stdio.h>
#define BING(x,y) ((x)<<y)
#define BANG(x) (1<<x)
#define BOOM    1
int main () {
    int x,y,z,w;
    int V[3] = {BING(x=227380393,BANG(BOOM)+BOOM), x+(w=BOOM+BANG(BANG(BOOM))),
                BING(47*(y=17453197),BOOM)), x+y+BING(w*w*17*185527,BANG(BOOM))};
    char *p=(char *)V;
    while(*(p-BOOM)!=BOOM+BING(BOOM,w)) putchar(*p++);
    return 0;
} /* Mind the comma operator! */
```

```
q:\>g++ prog.cpp -o prog.exe

q:\>prog
Hello World!
```

- 有看沒懂:

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
    int i, j, k, index, mult=1, iBucket;
    int len = maxNumDigits(ndata, data);
    for (i=0; i<len; i++) {
        for (j=0; j<8; j++) nBucket[j] = 0;
        for (j=0; j<ndata; j++) {
            iBucket = data[j] / mult % 8;
            buckets[iBucket][nBucket[iBucket]++] = data[j];
        }
        index = 0;
        for (j=0; j<8; j++)
            for (k=0; k<nBucket[j]; k++)
                data[index++] = buckets[j][k];
        mult *= 8;
    }
}
```

- 有看沒懂: 沒有還原出程式運作時的狀況

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
    int i, j, k, index, mult=1, iBucket;
    int len = maxNumDigits(ndata, data);
    for (i=0; i<len; i++) {
        for (j=0; j<8; j++) nBucket[j] = 0;
        for (j=0; j<ndata; j++) {
            iBucket = data[j] / mult % 8;
            buckets[iBucket][nBucket[iBucket]++] = data[j];
        }
        index = 0;
        for (j=0; j<8; j++)
            for (k=0; k<nBucket[j]; k++)
                data[index++] = buckets[j][k];
        mult *= 8;
    }
}
```


- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子
 - 有沒有出現北車大迷宮的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子
 - 有沒有出現北車大迷宮的樣子
 - 有沒有出現捷運月台的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子
 - 有沒有出現北車大迷宮的樣子
 - 有沒有出現捷運月台的樣子
 - 有沒有出現捷運淡水車站的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子
 - 有沒有出現北車大迷宮的樣子
 - 有沒有出現捷運月台的樣子
 - 有沒有出現捷運淡水車站的樣子
 - 有沒有出現等候公車的樣子

- 有看沒懂: 沒有還原出程式運作時的狀況
- 想像一下你現在要寫一個由海大到淡江的指引
 - 有沒有出現北寧路公車站的樣子
 - 有沒有出現103/104公車上擁擠的樣子
 - 有沒有出現基隆新站的樣子
 - 有沒有出現電聯車上的樣子
 - 有沒有出現北車大迷宮的樣子
 - 有沒有出現捷運月台的樣子
 - 有沒有出現捷運淡水車站的樣子
 - 有沒有出現等候公車的樣子

要把別人教會之前, 心中通常有很清楚的圖像

- 有看沒懂: 沒有還原出程式運作時的狀況

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
    int i, j, k, index, mult=1, iBucket;
    int len = maxNumDigits(ndata, data);
    for (i=0; i<len; i++) {
        for (j=0; j<8; j++) nBucket[j] = 0;
        for (j=0; j<ndata; j++) {
            iBucket = data[j] / mult % 8;
            buckets[iBucket][nBucket[iBucket]++] = data[j];
        }
        index = 0;
        for (j=0; j<8; j++)
            for (k=0; k<nBucket[j]; k++)
                data[index++] = buckets[j][k];
        mult *= 8;
    }
}
```

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型
- 由小到大排好 n 個八進位表示的整數
 - e.g. 100, 003, 667, 027, 120, 013, 325


$$3 \cdot 8^2 + 2 \cdot 8 + 5$$

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

- 由小到大排好 n 個八進位表示的整數
 - e.g. 100, 003, 667, 027, 120, 013, 325
- 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0					
1					
2					
3				...	
4					
5					
6					
7					

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

– 由小到大排好 n 個八進位表示的整數

• e.g. 100, 003, 667, 027, 120, 013, 325

– 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0					
1					
2					
3				...	
4					
5					
6					
7					

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

– 由小到大排好 n 個八進位表示的整數

- e.g. 100, 003, 667, 027, 120, 013, 325

– 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

- 由小到大排好 n 個八進位表示的整數
 - e.g. 100, 003, 667, 027, 120, 013, 325
- 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

– 由小到大排好 n 個八進位表示的整數

• e.g. 100, 003, 667, 027, 120, 013, 325

– 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

	0	1	2		n-1
0	100	003			
1	013				
2	120	325	027		
3					
4					
5					
6	667				
7					

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

– 由小到大排好 n 個八進位表示的整數

• e.g. 100, 003, 667, 027, 120, 013, 325

– 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

	0	1	2		n-1
0	100	003			
1	013				
2	120	325	027		
3					
4					
5					
6	667				
7					

100	003	013	120	325	027	667
-----	-----	-----	-----	-----	-----	-----

- 有看沒懂: 沒有還原出程式運作時的狀況
需要了解程式運作的模型

– 由小到大排好 n 個八進位表示的整數

• e.g. 100, 003, 667, 027, 120, 013, 325

– 如下圖準備八列空間來暫存資料, 每一列有 n 格

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

	0	1	2		n-1
0	100	003			
1	013				
2	120	325	027		
3				...	
4					
5					
6	667				
7					

100	003	013	120	325	027	667
-----	-----	-----	-----	-----	-----	-----

	0	1	2		n-1
0	003	013	027		
1	100	120			
2	325				
3				...	
4					
5					
6	667				
7					

003	013	027	100	120	325	667
-----	-----	-----	-----	-----	-----	-----

- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
```

- 由小到大排好 n 個

- e.g. 100, 003, 6

- 如下圖準備八列空

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

0	
1	
2	
3	
4	
5	
6	
7	

100	003	013	120	325	027	667
-----	-----	-----	-----	-----	-----	-----

003	013	027	100	120	325	667
-----	-----	-----	-----	-----	-----	-----

- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

- 由小到大排好 n 個
 - e.g. 100, 003, 6
- 如下圖準備八列空

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
    int i, j, k, index, mult=1, iBucket;
    int len = maxNumDigits(ndata, data);
```

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

0	
1	
2	
3	
4	
5	
6	
7	

100	003	013	120	325	027	667
-----	-----	-----	-----	-----	-----	-----

003	013	027	100	120	325	667
-----	-----	-----	-----	-----	-----	-----

- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

- 由小到大排好 n 個

- e.g. 100, 003, 6

- 如下圖準備八列空

	0	1	2		n-1
0	100	120			
1					
2					
3	003	013		...	
4					
5	325				
6					
7	667	027			

100	120	003	013	325	667	027
-----	-----	-----	-----	-----	-----	-----

```
void bucketSort(int ndata, int data[]) {
    int buckets[8][MAX], nBucket[8];
    int i, j, k, index, mult=1, iBucket;
    int len = maxNumDigits(ndata, data);
    for (i=0; i<len; i++) {
```

```
    }
}
```

- 需要了解程序
- ```
void bucketSort(int ndata, int data[]) {
```

- e.g. 100, 003, 6

- 1   2   ...   n-1   |

|   | 0   | 1   | 2 | n-1 |     |  |
|---|-----|-----|---|-----|-----|--|
| 0 | 100 | 120 |   |     |     |  |
| 1 |     |     |   |     |     |  |
| 2 |     |     |   |     |     |  |
| 3 | 003 | 013 |   |     | ... |  |
| 4 |     |     |   |     |     |  |
| 5 | 325 |     |   |     |     |  |
| 6 |     |     |   |     |     |  |
| 7 | 667 | 027 |   |     |     |  |

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| 100 | 120 | 003 | 013 | 325 | 667 | 027 |
|-----|-----|-----|-----|-----|-----|-----|

```
void bucketSort(int ndata, int data[]) {
 int buckets[8][MAX], nBucket[8];
 int i, j, k, index, mult=1, iBucket;
 int len = maxNumDigits(ndata, data);
 for (i=0; i<len; i++) {
 for (j=0; j<8; j++) nBucket[j] = 0;
```



- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

- 由小到大排好  $n$  個

- e.g. 100, 003, 6

- 如下圖準備八列空

|   | 0   | 1   | 2 |     | n-1 |
|---|-----|-----|---|-----|-----|
| 0 | 100 | 120 |   |     |     |
| 1 |     |     |   |     |     |
| 2 |     |     |   |     |     |
| 3 | 003 | 013 |   | ... |     |
| 4 |     |     |   |     |     |
| 5 | 325 |     |   |     |     |
| 6 |     |     |   |     |     |
| 7 | 667 | 027 |   |     |     |

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| 100 | 120 | 003 | 013 | 325 | 667 | 027 |
|-----|-----|-----|-----|-----|-----|-----|

```

void bucketSort(int ndata, int data[]) {
 int buckets[8][MAX], nBucket[8];
 int i, j, k, index, mult=1, iBucket;
 int len = maxNumDigits(ndata, data);
 for (i=0; i<len; i++) {
 for (j=0; j<8; j++) nBucket[j] = 0;
 for (j=0; j<ndata; j++) {
 iBucket = data[j] / mult % 8;
 buckets[iBucket][nBucket[iBucket]++] = data[j];
 }
 }
}

```

- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

- 由小到大排好  $n$  個

- e.g. 100, 003, 6

- 如下圖準備八列空

|   | 0   | 1   | 2 |     | n-1 |
|---|-----|-----|---|-----|-----|
| 0 | 100 | 120 |   |     |     |
| 1 |     |     |   |     |     |
| 2 |     |     |   |     |     |
| 3 | 003 | 013 |   | ... |     |
| 4 |     |     |   |     |     |
| 5 | 325 |     |   |     |     |
| 6 |     |     |   |     |     |
| 7 | 667 | 027 |   |     |     |

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| 100 | 120 | 003 | 013 | 325 | 667 | 027 |
|-----|-----|-----|-----|-----|-----|-----|

```

void bucketSort(int ndata, int data[]) {
 int buckets[8][MAX], nBucket[8];
 int i, j, k, index, mult=1, iBucket;
 int len = maxNumDigits(ndata, data);
 for (i=0; i<len; i++) {
 for (j=0; j<8; j++) nBucket[j] = 0;
 for (j=0; j<ndata; j++) {
 iBucket = data[j] / mult % 8;
 buckets[iBucket][nBucket[iBucket]++] = data[j];
 }
 index = 0;
 for (j=0; j<8; j++)
 for (k=0; k<nBucket[j]; k++)
 data[index++] = buckets[j][k];
 }
}

```

- 有看沒懂: 沒有還原出程式運作時的狀況

需要了解程式

- 由小到大排好  $n$  個

- e.g. 100, 003, 6

- 如下圖準備八列空

|   | 0   | 1   | 2 |     | n-1 |
|---|-----|-----|---|-----|-----|
| 0 | 100 | 120 |   |     |     |
| 1 |     |     |   |     |     |
| 2 |     |     |   |     |     |
| 3 | 003 | 013 |   | ... |     |
| 4 |     |     |   |     |     |
| 5 | 325 |     |   |     |     |
| 6 |     |     |   |     |     |
| 7 | 667 | 027 |   |     |     |

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| 100 | 120 | 003 | 013 | 325 | 667 | 027 |
|-----|-----|-----|-----|-----|-----|-----|

```

void bucketSort(int ndata, int data[]) {
 int buckets[8][MAX], nBucket[8];
 int i, j, k, index, mult=1, iBucket;
 int len = maxNumDigits(ndata, data);
 for (i=0; i<len; i++) {
 for (j=0; j<8; j++) nBucket[j] = 0;
 for (j=0; j<ndata; j++) {
 iBucket = data[j] / mult % 8;
 buckets[iBucket][nBucket[iBucket]++] = data[j];
 }
 index = 0;
 for (j=0; j<8; j++)
 for (k=0; k<nBucket[j]; k++)
 data[index++] = buckets[j][k];
 mult *= 8;
 }
}

```

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

```
int i, num = -1, K = -1;
char digit[MAXNUMBER];
scanf("%d%d", &nun, &K);
```

Error: 識別項 "nun" 未定義

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

```
int i, num = -1, K = -1;
char digit[MAXNUMBER];
scanf("%d%d", &nun,&K);
```

Error: 識別項 "nun" 未定義

```
for (i=0; i<num; i++) digit[i] = i;
do
 if (checkOutOfOrderCounts)
```

Error: 識別項 "checkOutOfOrderCounts" 未定義



# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

```
int i, num = -1, K = -1;
char digit[MAXNUMBER];
scanf("%d%d", &num, &K);
```

Error: 識別項 "nun" 未定義

```
for (i=0; i<num; i++) digit[i] = i;
do
```

```
 if (checkOutOfOrderCounts)
```

Error: 識別項 "checkOutOfOrderCounts" 未定義

- **線上語法說明**

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

```
int i, num = -1, K = -1;
char digit[MAXNUMBER];
scanf("%d%d", &num, &K);
```

Error: 識別項 "num" 未定義

```
for (i=0; i<num; i++) digit[i] = i;
do
```

```
 if (checkOutOfOrderCounts)
```

Error: 識別項 "checkOutOfOrderCounts" 未定義

- **線上語法說明**

```
do
```

```
 if (checkOutOfOrderCountsK(num, digit, K))
```

```
while (r
return 0;
```

int checkOutOfOrderCountsK(int size, char \*data, int K)

# 協助編碼的工具

- 撰寫程式時還是需要記住好多東西, 都沒有工具可以幫忙嗎?
  - **編譯器**: 檢查你寫的程式是否符合語法
  - **自動程式碼完成工具**: Intellisense (MS), Eclipse, Code completion plugin of Code::Blocks 主要是提示函式名稱, 函式參數, 語法一寫錯就立刻提示

```
int i, num = -1, K = -1;
char digit[MAXNUMBER];
scanf("%d%d", &num, &K);
```

Error: 識別項 "num" 未定義

```
for (i=0; i<num; i++) digit[i] = i;
do
```

```
 if (checkOutOfOrderCounts)
```

Error: 識別項 "checkOutOfOrderCounts" 未定義

- **線上語法說明**
- **google**

```
do
```

```
 if (checkOutOfOrderCountsK(num, digit, K))
```

```
while (r
return 0;
```

```
int checkOutOfOrderCountsK(int size, char *data, int K)
```

# 協助編碼的工具 (cont'd)

- 太開心了, 有了這些工具, 語法不必記下來了吧? 有更重要的東西需要我的腦袋!!
  - **編譯器**是最基本的工具, 一定要使用
  - 不要太依賴**自動程式碼完成工具**
    - 很多開發環境中**不提供**
    - 比較複雜的程式架構中**建議會錯**
    - 對語法越熟悉, **撰寫程式時才可以專心構思程式的功能**, 不會一直被不確定的語法干擾, 隨時可以評估一個想法是否寫得出來

# 練習寫程式的三大殺手

# 練習寫程式的三大殺手

- 及格就好: 不是 60 分就及格了嗎?!

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於使用這個程式的人是沒有用的

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於**使用這個程式的人**是沒有用的
- **二分法**: 以為程式不是對的就是錯的



# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於**使用這個程式的人**是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於使用這個程式的人是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...      換一個系 😞

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於**使用這個程式的人**是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...  
對使用程式的人沒有用, 對**撰寫這個程式的人**可是有非常大的用途的

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於**使用這個程式的人**是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...  
對使用程式的人沒有用, 對**撰寫這個程式的人**可是有非常大的用途的
- **考試型的學習, 效率掛帥**:

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於**使用這個程式的人**是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...  
對使用程式的人沒有用, 對**撰寫這個程式的人**可是有非常大的用途的
- **考試型的學習, 效率掛帥**:  
以為程式發生錯誤時,  
正確答案在課本上, 換成正確的就好, 省時省力

# 練習寫程式的三大殺手

- **及格就好**: 不是 60 分就及格了嗎?!  
一個程式有一點點(1%)邏輯錯誤時, 輸出的答案就是錯誤的, 對於使用這個程式的人是沒有用的
- **二分法**: 以為程式不是對的就是錯的  
以為程式發生錯誤時, 所有的結果都是沒有用的!!  
換一個程式, 換一個方法...  
對使用程式的人沒有用, 對撰寫這個程式的人可是有非常大的用途的
- **考試型的學習, 效率掛帥**:  
以為程式發生錯誤時,  
正確答案在課本上, 換成正確的就好, 省時省力

**debug**: 分析目前的程式和結果, 一點一點修改程式來得到想要的結果

# 實習課程的特點

# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案



# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案
- 直接讓你看問題的解決範例程式或是虛擬程式碼

# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案
- 直接讓你看問題的解決範例程式或是虛擬程式碼
- 可是學習設計程式時最大的障礙在於
  - 為什麼會想到運用某一種語法解決問題
  - 怎麼簡化與分析問題, 一步一步得到解決問題的方法

# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案
- 直接讓你看問題的解決範例程式或是虛擬程式碼
- 可是學習設計程式時最大的障礙在於
  - 為什麼會想到運用某一種語法解決問題
  - 怎麼簡化與分析問題, 一步一步得到解決問題的方法
- 只有題目, 同學很難進入狀況; 直接給虛擬程式碼或是程式碼, 練習的時候其實已經沒有思考的空間了

# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案
- 直接讓你看問題的解決範例程式或是虛擬程式碼
- 可是學習設計程式時最大的障礙在於
  - 為什麼會想到運用某一種語法解決問題
  - 怎麼簡化與分析問題, 一步一步得到解決問題的方法
- 只有題目, 同學很難進入狀況; 直接給虛擬程式碼或是程式碼, 練習的時候其實已經沒有思考的空間了
- 所以在這個課程以及實習裡, 我盡量避免直接讓你看到最後的程式碼, 而是一步驟一步驟說明思考的重點、需求、方向、以及轉折點

# 實習課程的特點

- 大部分實習課程只說明語法或是只給程式題目/答案
- 直接讓你看問題的解決範例程式或是虛擬程式碼
- 可是學習設計程式時最大的障礙在於
  - 為什麼會想到運用某一種語法解決問題
  - 怎麼簡化與分析問題, 一步一步得到解決問題的方法
- 只有題目, 同學很難進入狀況; 直接給虛擬程式碼或是程式碼, 練習的時候其實已經沒有思考的空間了
- 所以在這個課程以及實習裡, 我盡量避免直接讓你看到最後的程式碼, 而是一步驟一步驟說明思考的重點、需求、方向、以及轉折點
- 不直接讓你看到程式碼, 目的不在吊你胃口, 也不是不考慮你其他課程的負擔, 而是希望問題在你的腦中成型, 由多個角度去辨認這個問題, 抽象化這個問題, 從而思考解決方案