


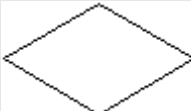




流程圖說明

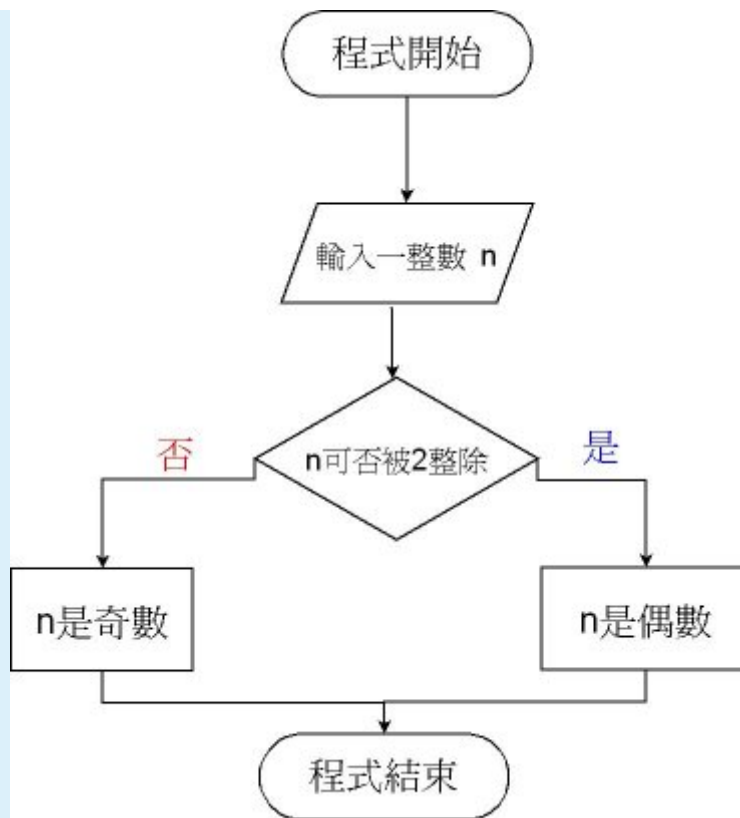
一、流程圖元件

流程圖是由下列圖形所構成：

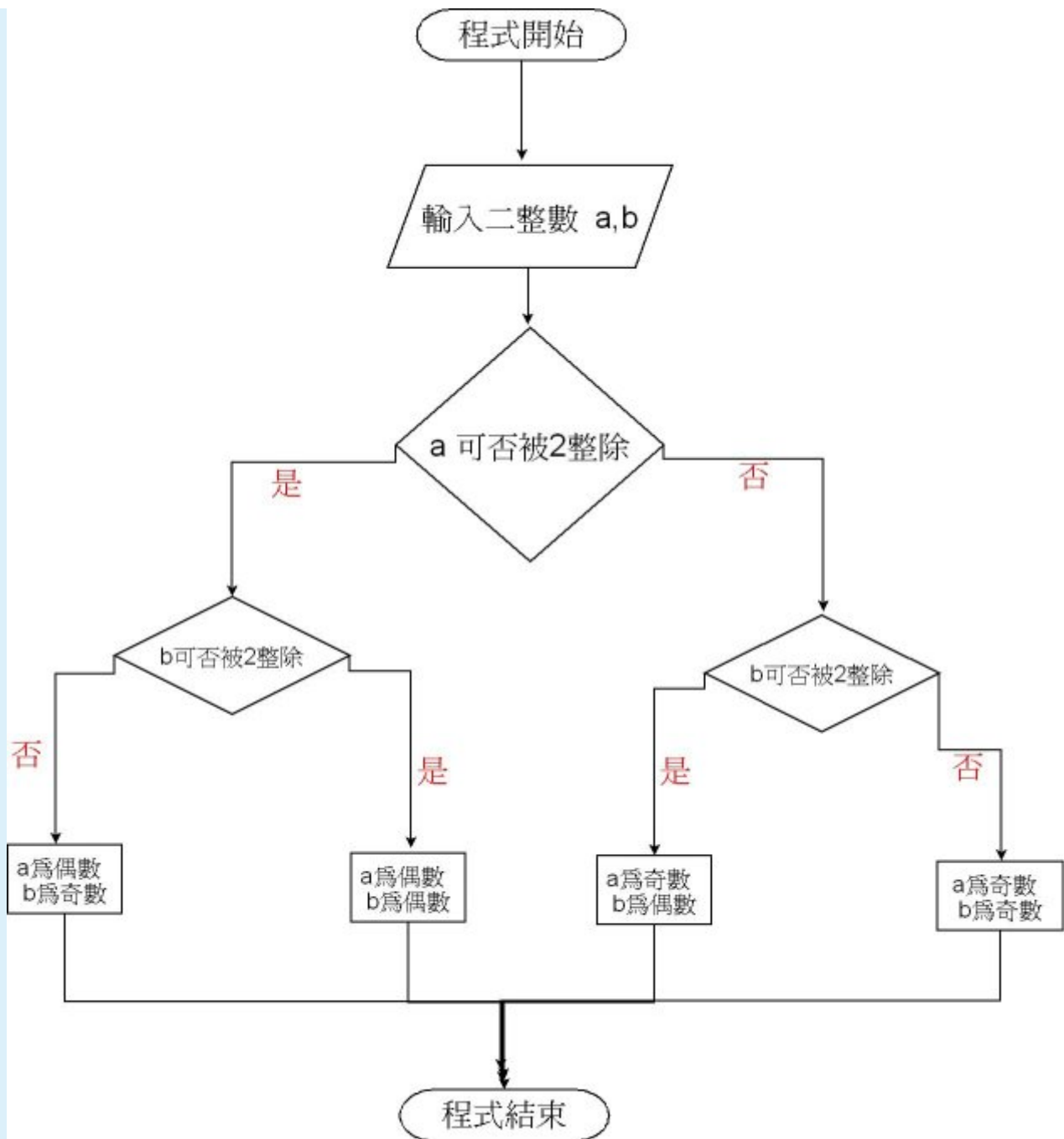
開始或結束工作的圖形	
輸入/輸出工作的圖形	
處理工作的圖形	
條件判斷的圖形	
工作流向的圖形	
連接點	

二、範例

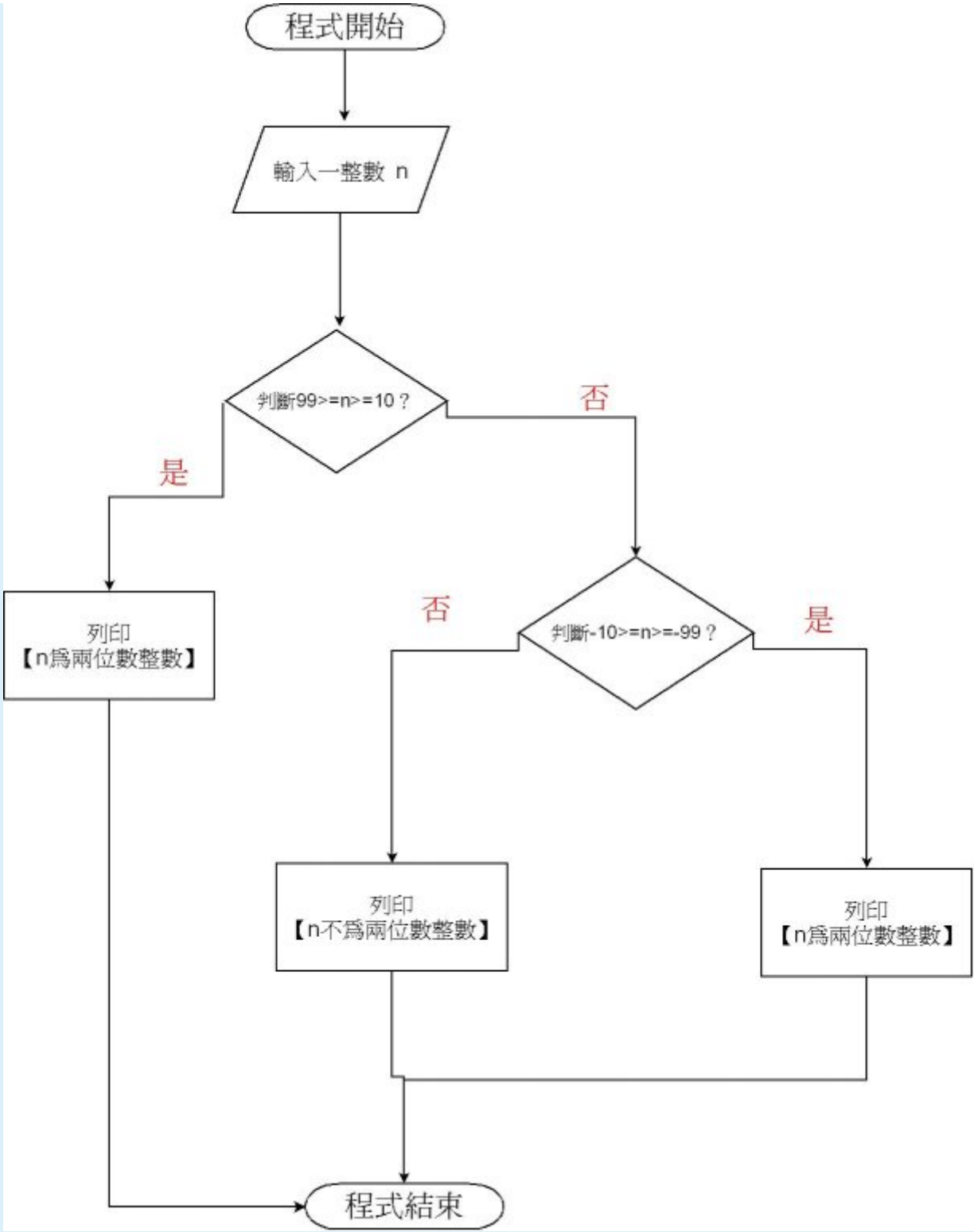
【例 2.1】輸入一正整數，判斷為奇數或偶數。用流程圖方式表示。



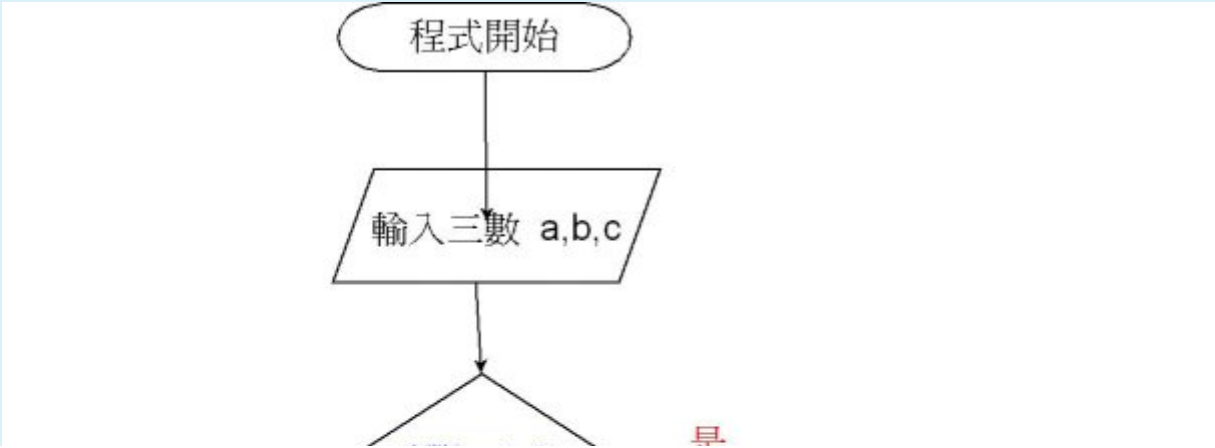
【例 2.2】輸入兩正整數，判斷此二數是否為一個奇數與一個偶數，用流程圖表示。

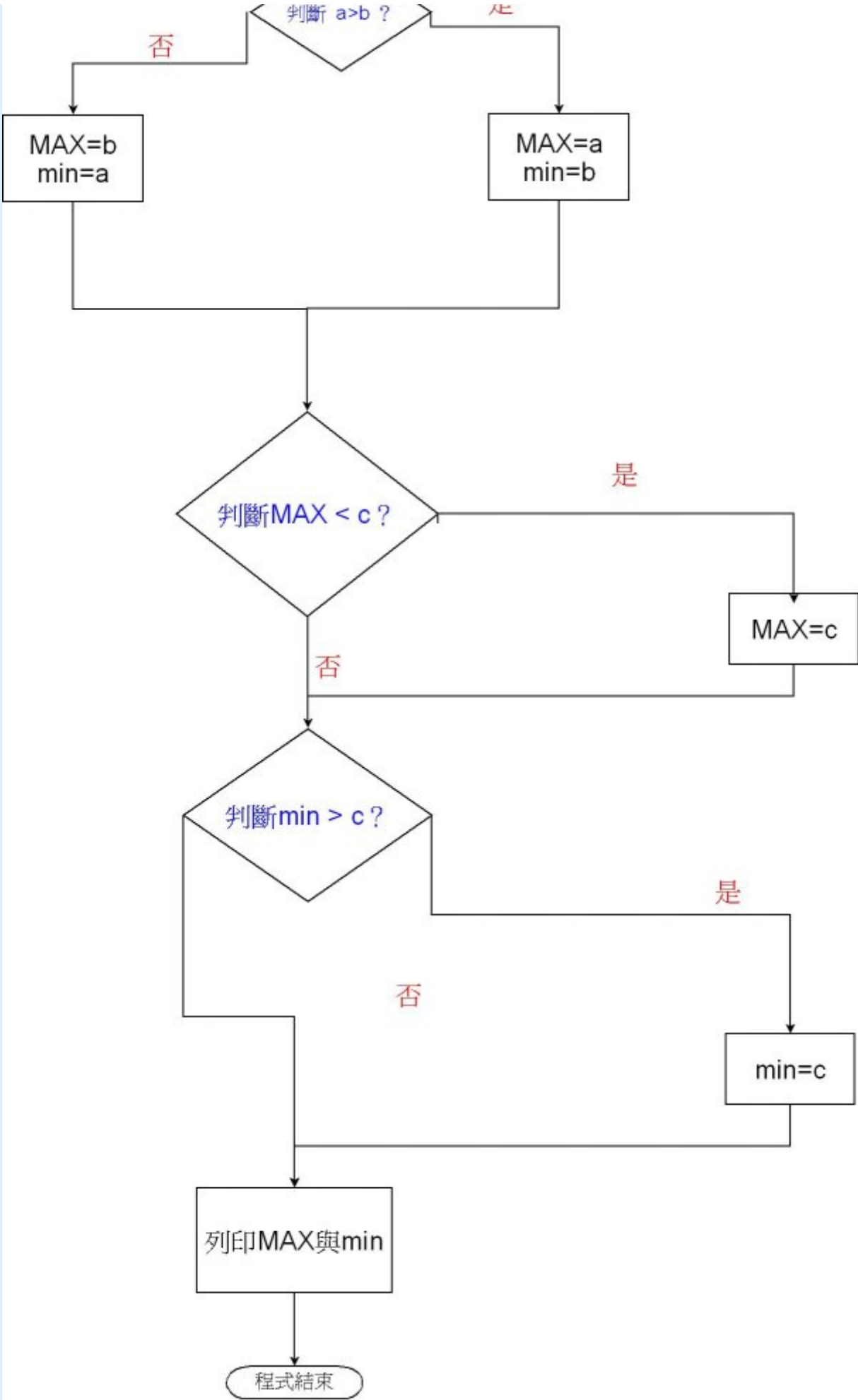


【例 2.3】輸入一個整數，請判斷這個整數為兩位數整數（請考慮99~10 與 -10~-99 兩種狀況）。

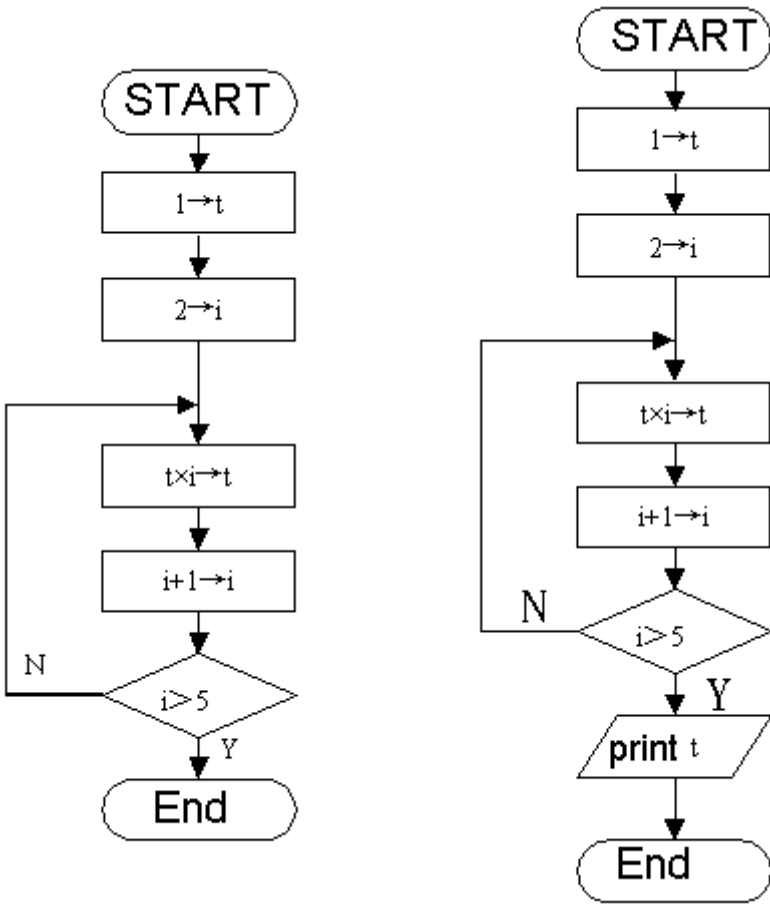


【例 2.4】 輸入三數，找出最大值與最小值。

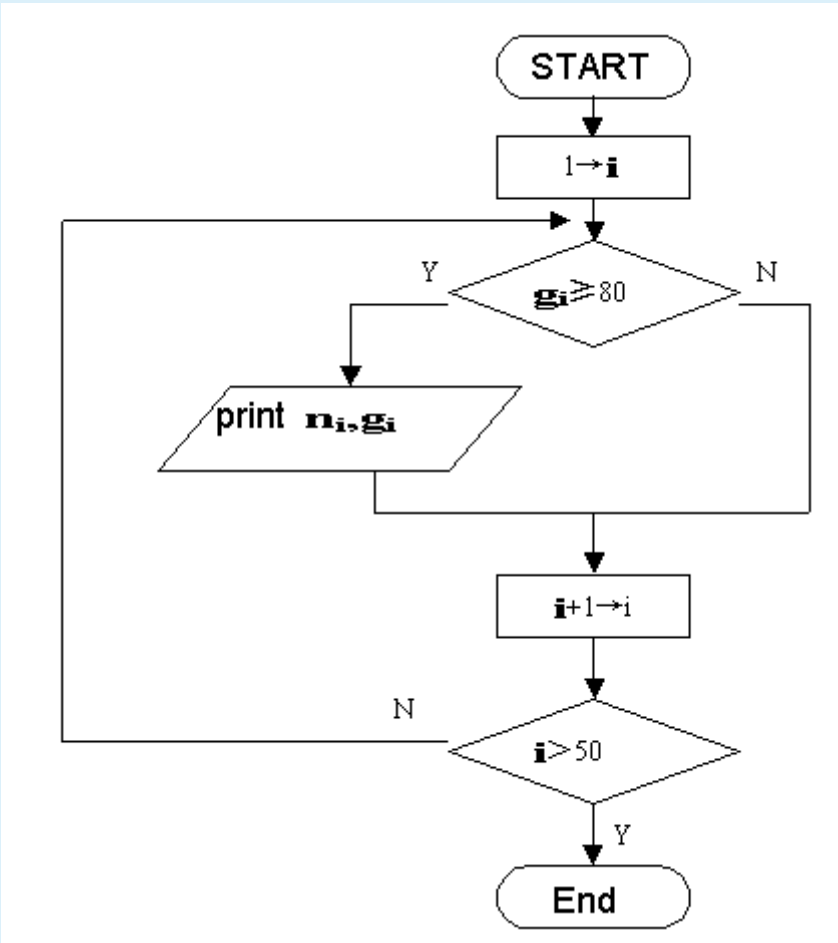




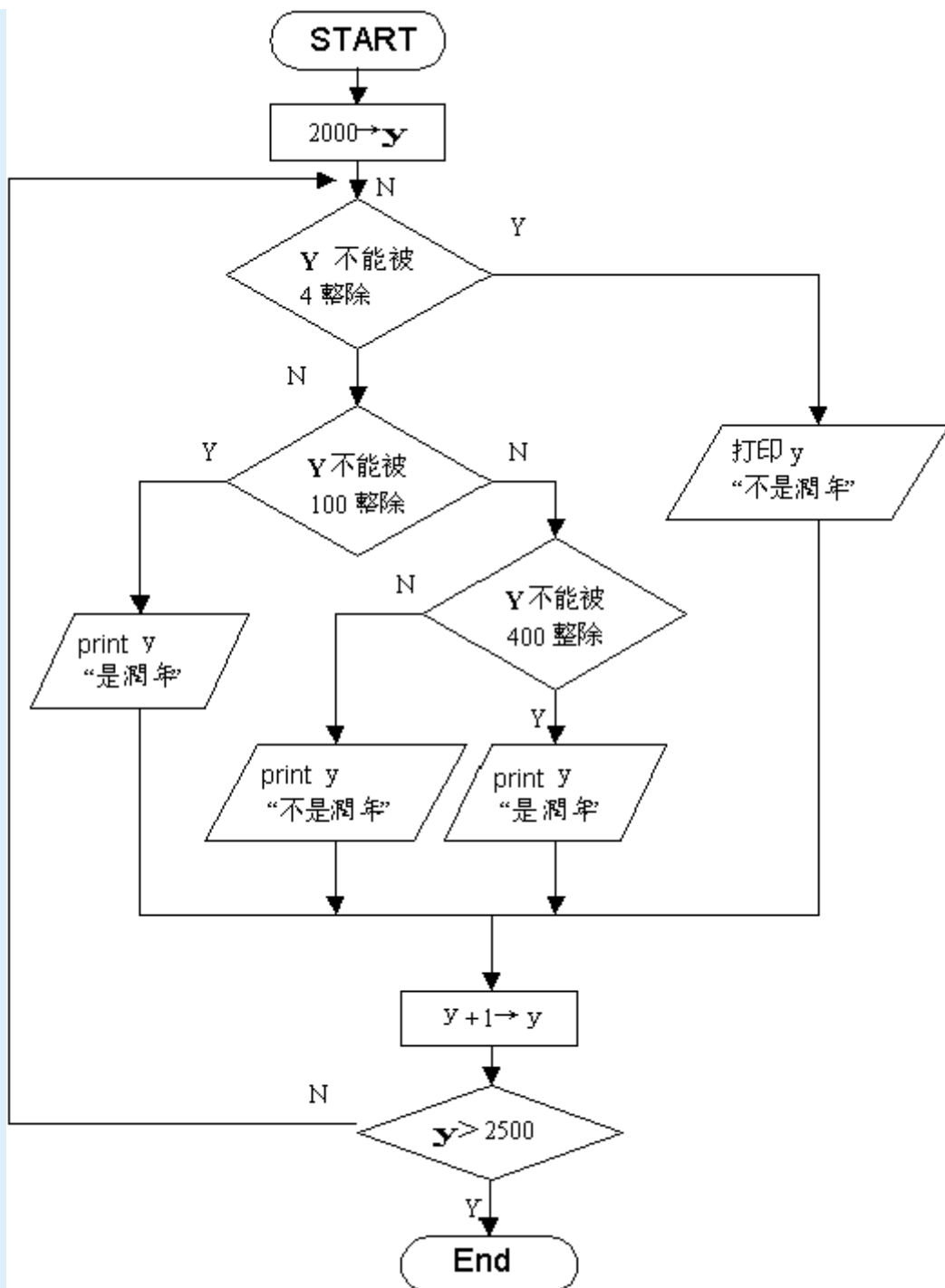
【例 2.6】 5! 的計算用流程圖表示。



【例 2.7】用流程圖表示。



【例 2.8】判定閏年的算用流程圖表示。



【例 2.9】將例 2.4 求 $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{1}{99} - \frac{1}{100}$ 的算用流程圖表示。

【例 2.10】

請設計一程式，輸入一學生成績，由程式判斷若成績在60分以上，請輸出“及格”，反之則為“不及格”。

1、開始

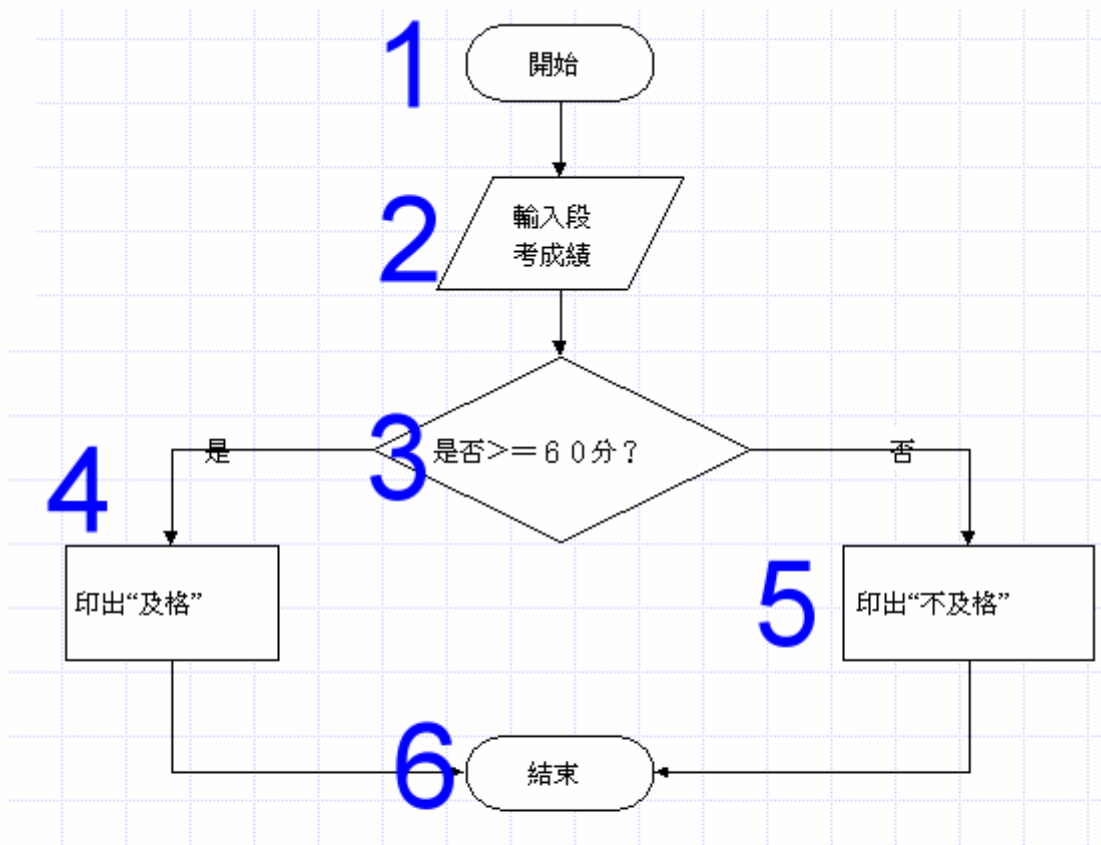
2、取得某科的段
考成績

3、判斷是否 ≥ 60

4、若是，則印出“及格”

5、若否，則印出“不及格”

6、結束



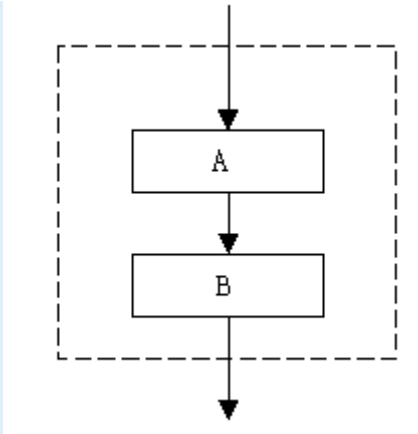
小結論：

一個流程圖包括：

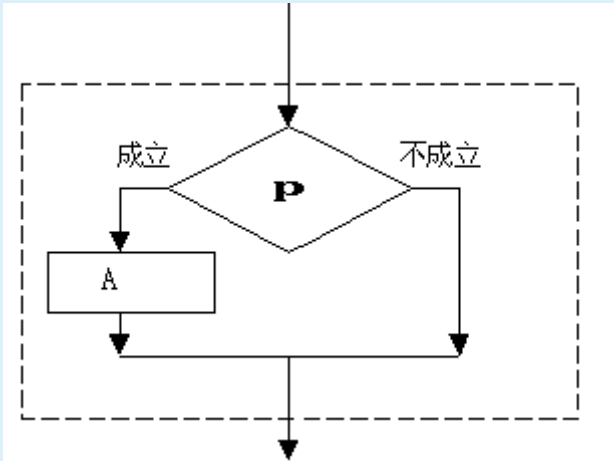
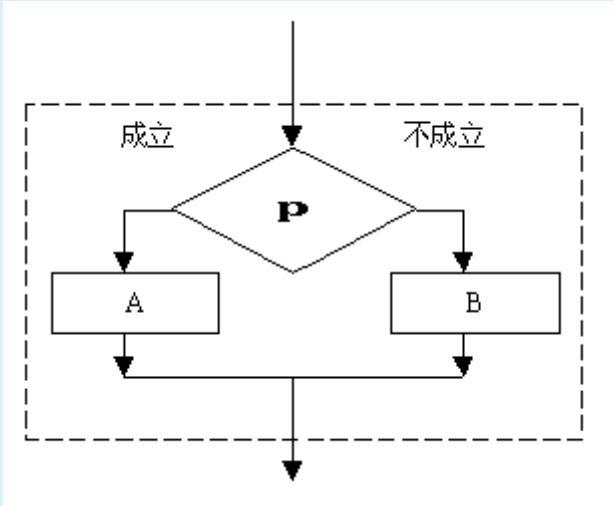
1. 表示相應操作的框；
2. 帶箭頭的流程線；
3. 框內外必要的文字說明。

三、三種基本結構和改進的流程圖

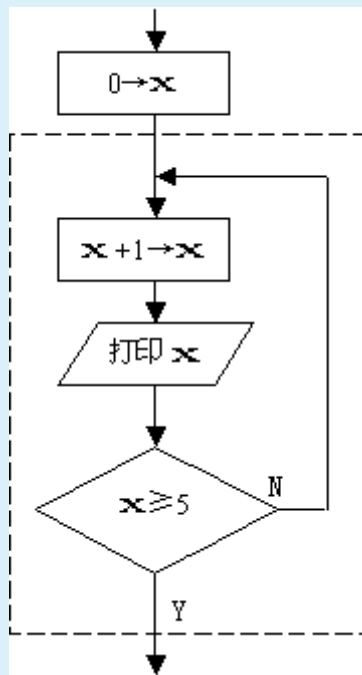
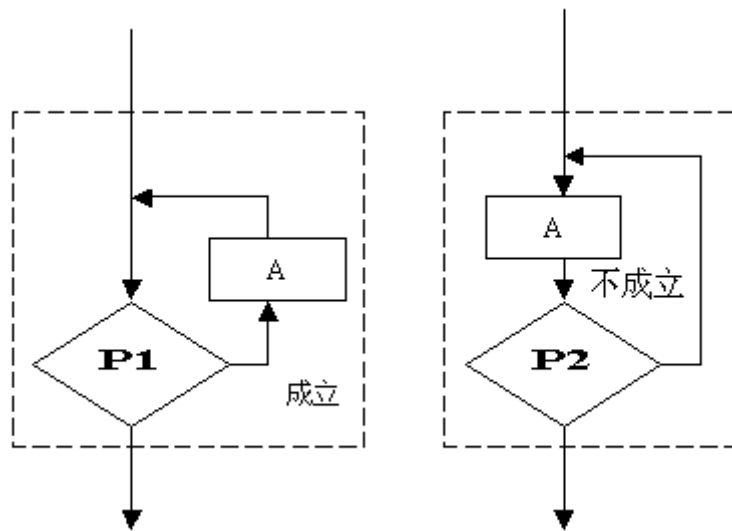
1. 順序結構：



2. 選擇結構：



3. 迴圈結構



三種基本結構的共同特點：

- 只有一個入口；
- 只有一個出口；
- 結構內的每一部分都有機會被執行到；
- 結構內不存在“閉環”。

四、用電腦語言表示演算法

- 我們的任務是用電腦解題，就是用電腦實現演算法；
- 用電腦語言表示演算法必須嚴格遵循所用語言的語法規則。

【例 2.20】求 $1 \times 2 \times 3 \times 4 \times 5$ 用 C 語言表示。

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

```
main()
```

```
{
```

```
int i,t ;
```

```
t=1 ;
```

```
i=2 ;
```

```
while(i<=5)
```

```
{t=t*i ;
```

```
i=i+1 ;
```

```
}
```

```
printf(“%d”,t) ;
```

```
}
```

【例2.21】求級數的值。

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

```
main()
```

```
{
```

```
int sigh=1 ;
```

```
float deno=2.0,sum=1.0,term ;
```

```
while(deno<=100)
```

```
{ sigh= -sigh ;
```

```
term= sigh/ deno ;
```

```
sum=sum+term ;
```

```
deno=deno+1 ;
```

```
}
```

```
printf(“%f”,sum) ;
```

```
}
```

五、 結構化程式設計方法

- 自頂向下；
- 逐步細化；
- 模組化設計；
- 結構化編碼。