

# Java

巢狀選擇



比較裡面  
還有比較

## 問題

寫一個程式，判斷一個數，  
是不是正偶數

問題



# 兩種寫法

當一個 if 需要使用兩個以上的條件判斷運算式時，可以改寫為使用多個 if，每個 if 裡面都使用一個運算式。

## 一個 if 搭配 && 運算子

```
if (num > 0 && num % 2 == 0) {  
    System.out.printf("%d是正偶數\n", num);  
}
```

## 兩個 if

```
if (num > 0) {  
    if (num % 2 == 0) {  
        System.out.printf("%d是正偶數\n", num);  
    }  
}
```

一個結構裡面還有同樣結構的寫法，稱為『**巢狀**』結構！

if 裡面還有 if 的寫法，就稱為『**巢狀**』選擇！



## 問題

寫一個程式讓使用者輸入分數，印出分數等級。

100到90之間是等級 A，其餘類推如下：

89-80 B

79-70 C

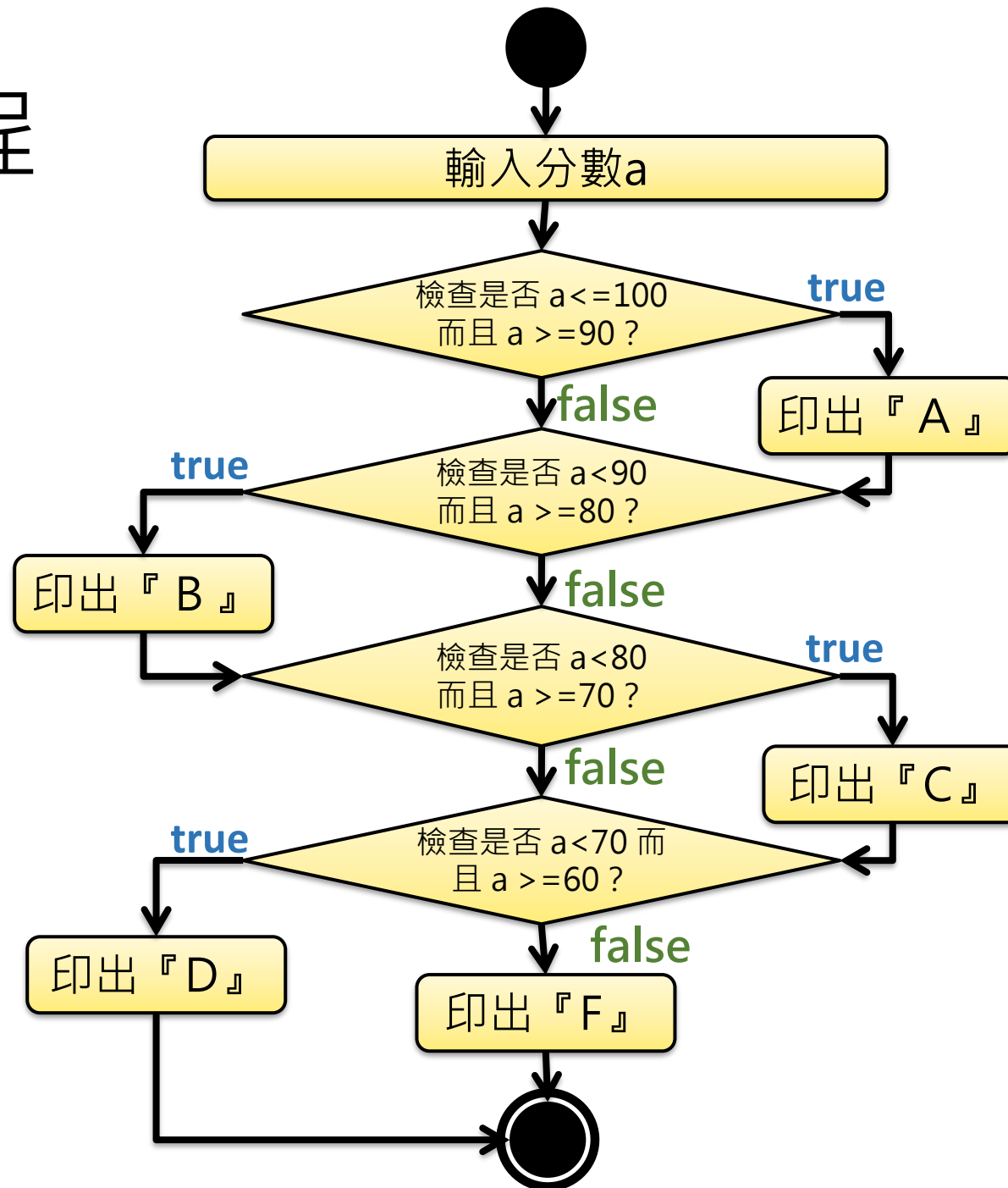
69-60 D

59-0 F

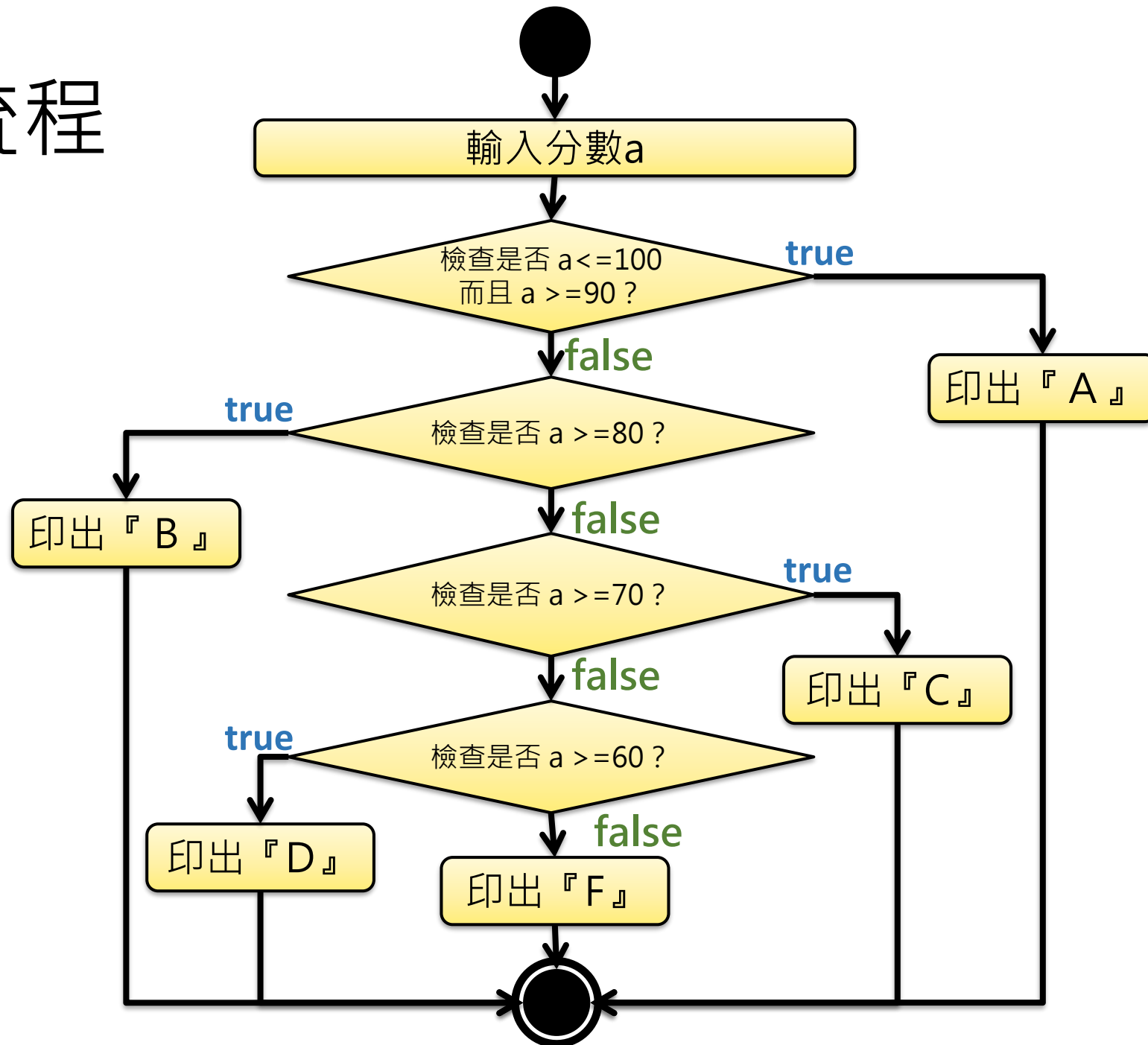
問題

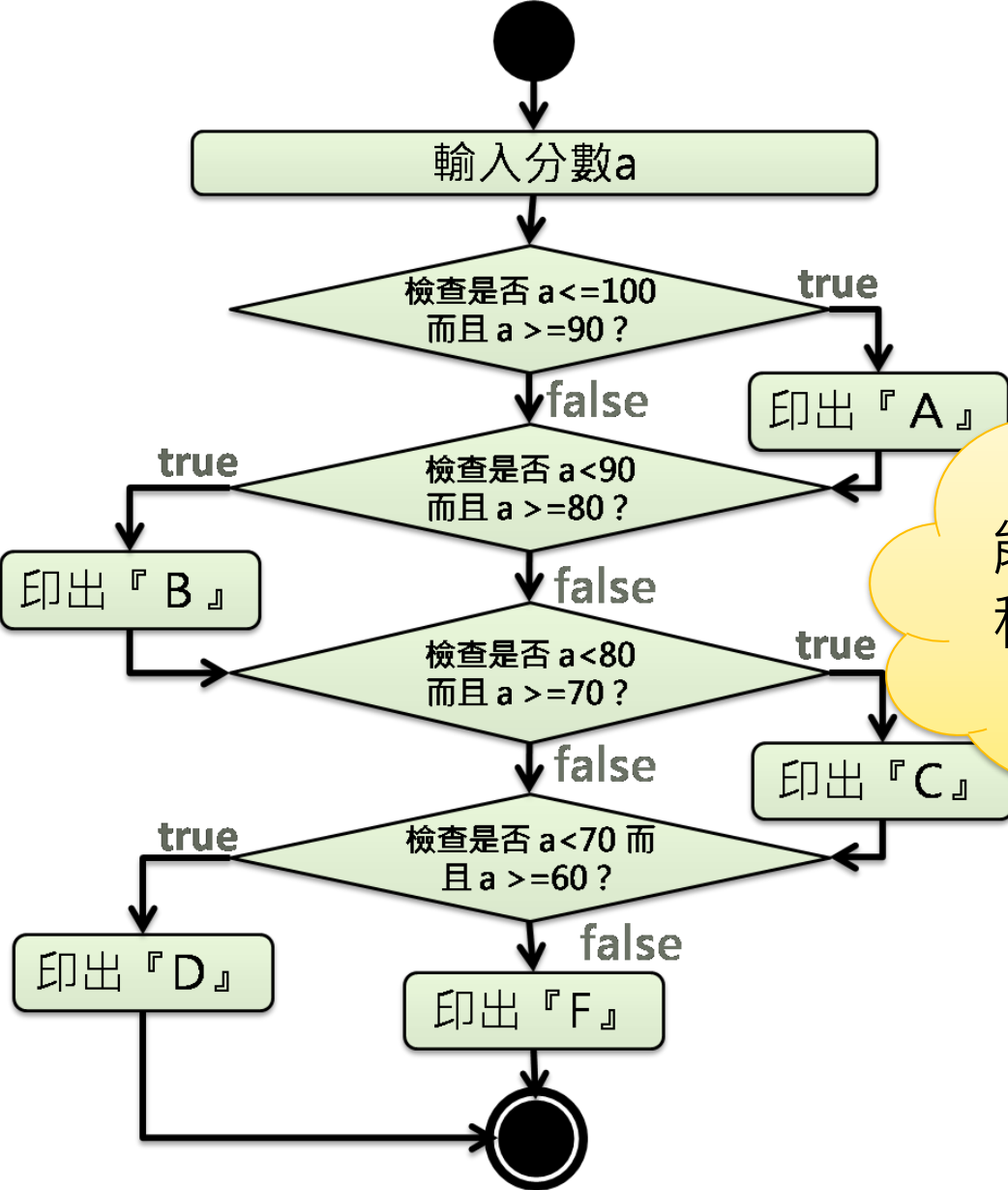


# 第一種流程

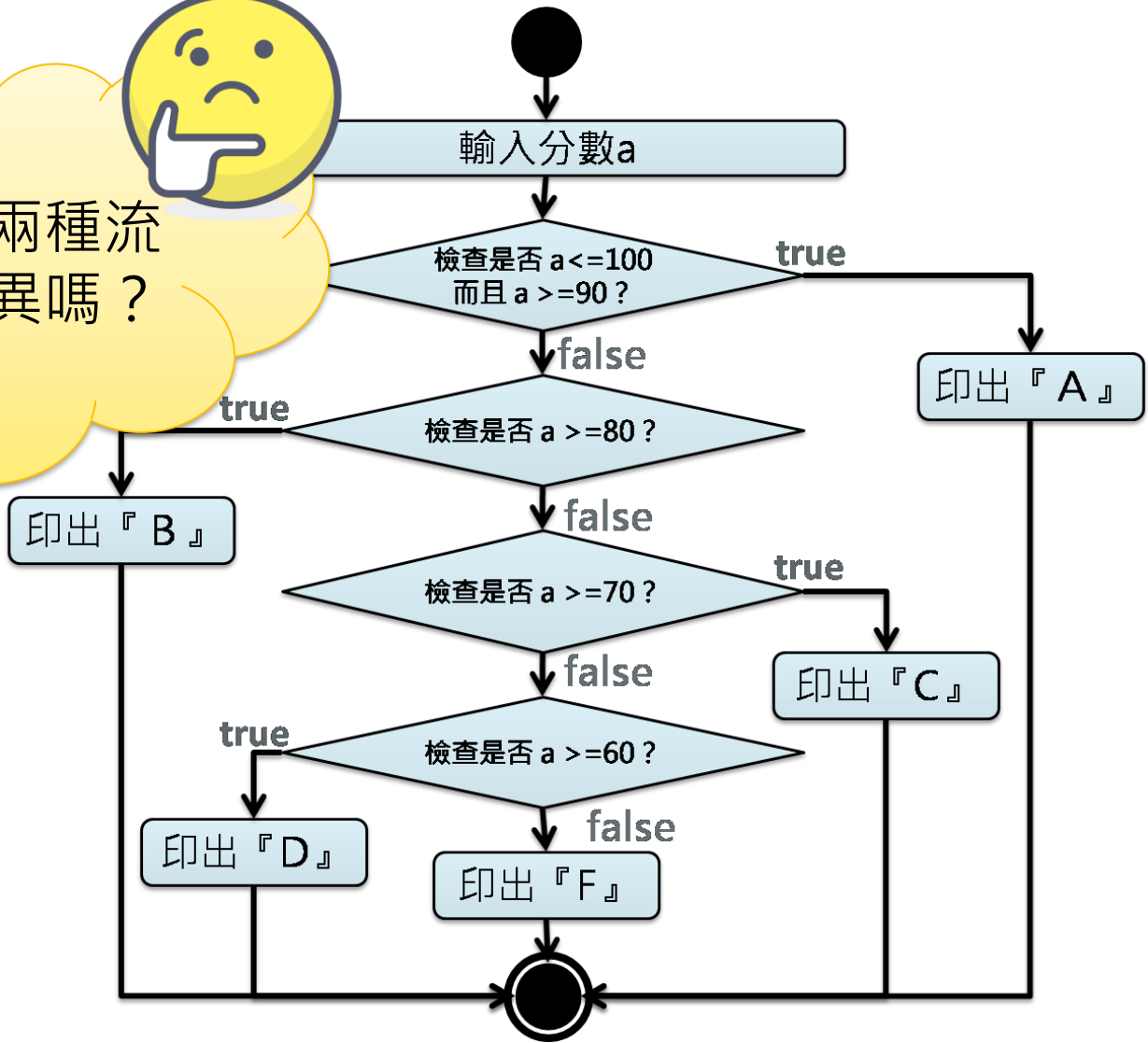


# 第二種流程

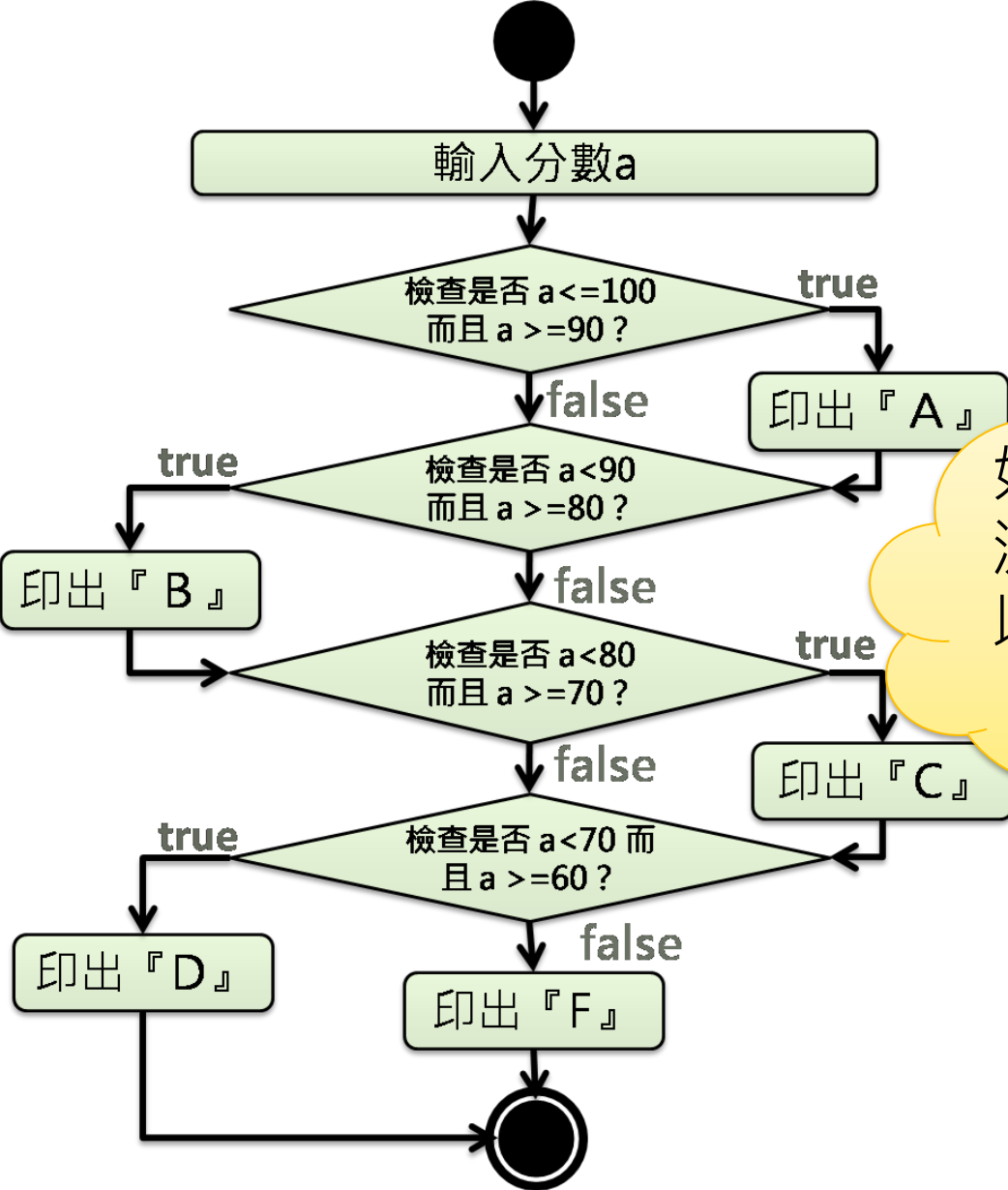




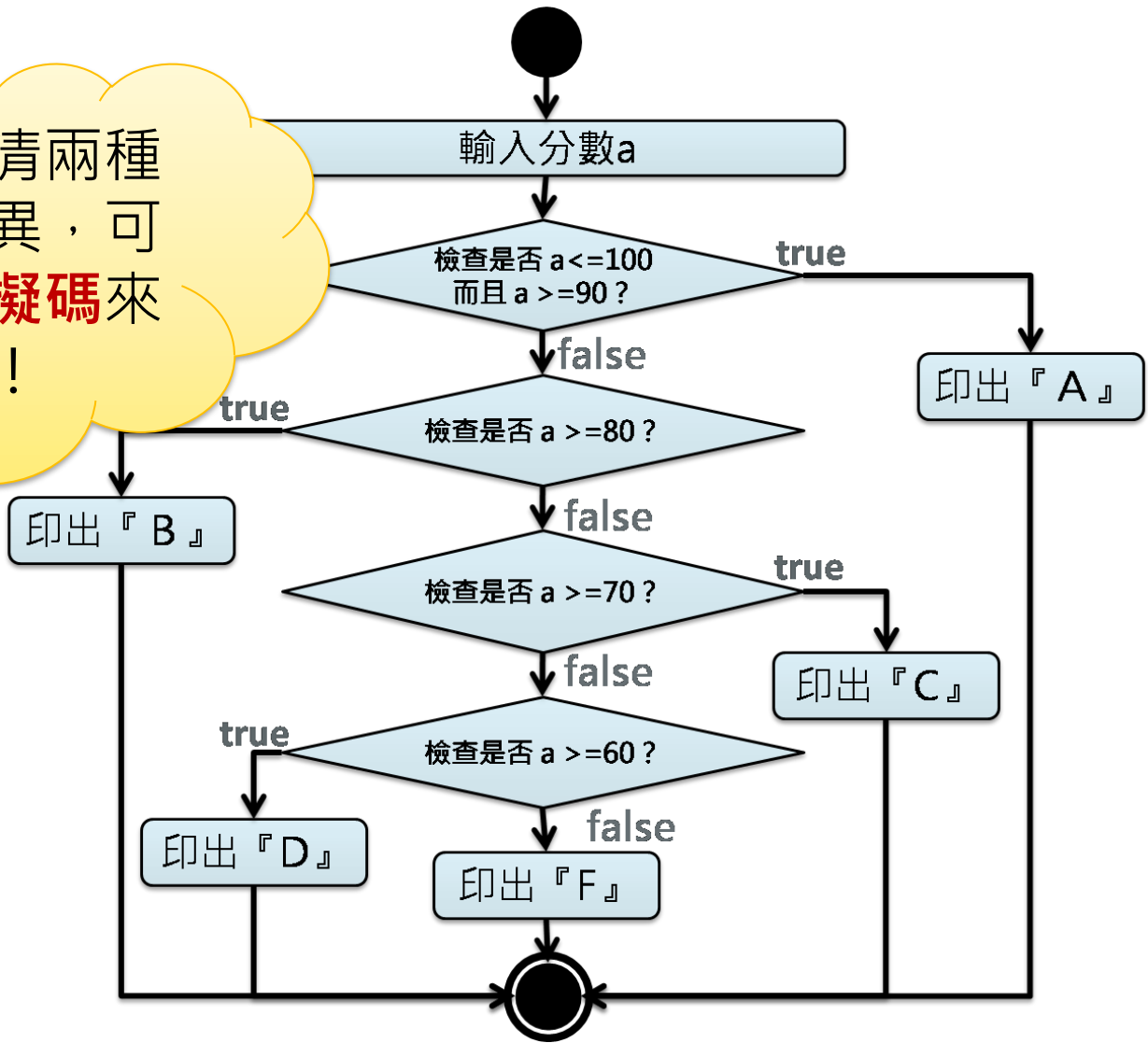
能比較出兩種流  
程圖的差異嗎？





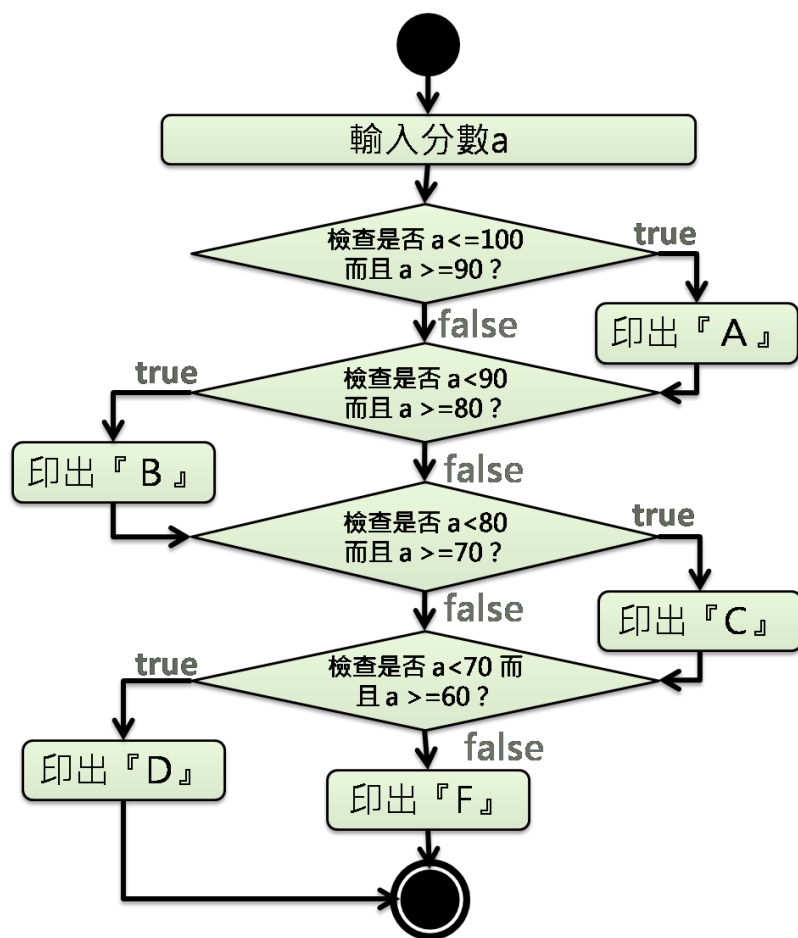


如果要釐清兩種  
流程的差異，可以  
使用**虛擬碼**來  
輔助！



## 第一種流程

# 用虛擬碼來分辨



如果 成績  $\leq 100$  而且  $\geq 90$

輸出 『A』

如果 成績  $< 90$  而且  $\geq 80$

輸出 『B』

如果 成績  $< 80$  而且  $\geq 70$

輸出 『C』

如果 成績  $< 70$  而且  $\geq 60$

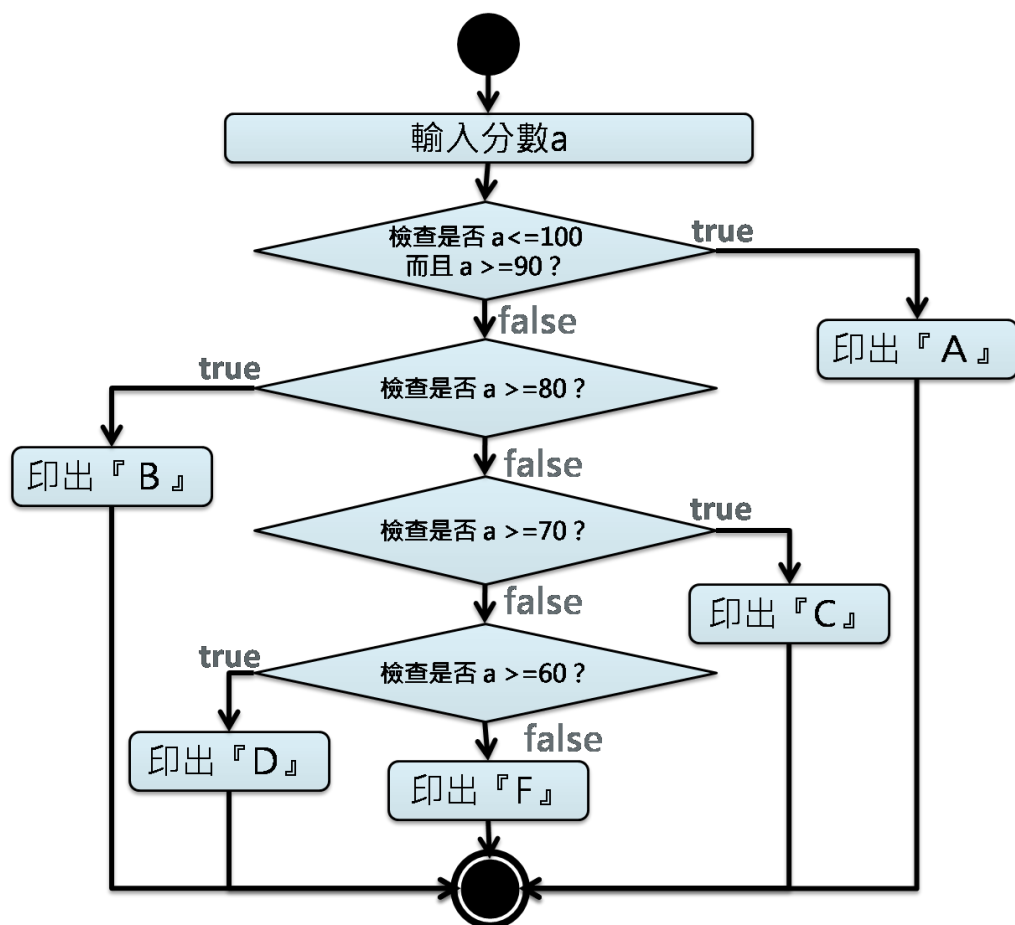
輸出 『D』

否則

輸出 『F』

## 第二種流程

# 用虛擬碼來分辨



如果 成績  $\leq 100$  而且  $\geq 90$

輸出 『A』

否則

如果 成績  $\geq 80$

輸出 『B』

否則

如果 成績  $\geq 70$

輸出 『C』

否則

如果 成績  $\geq 60$

輸出 『D』

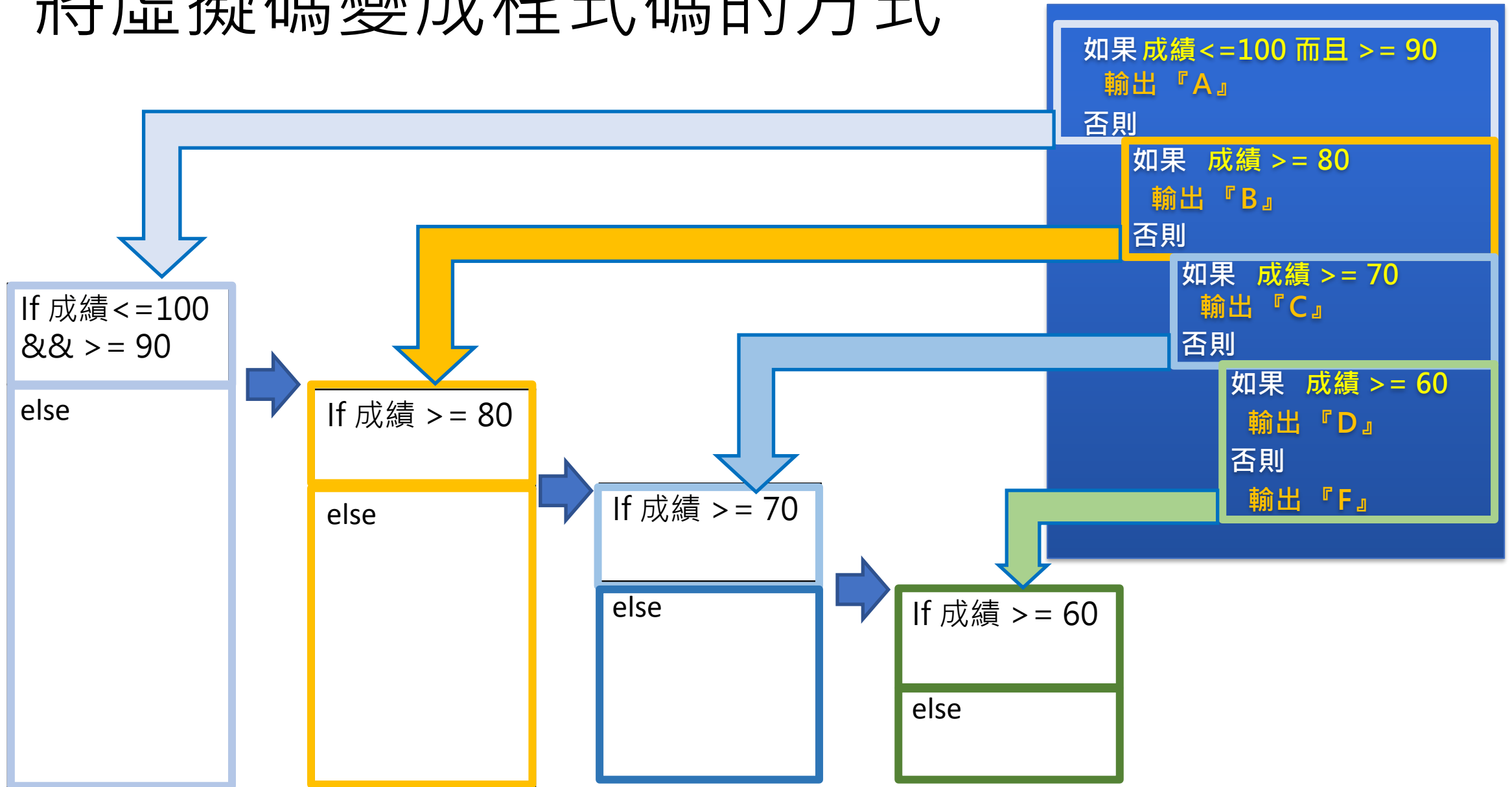
否則

輸出 『F』

# 甚麼是虛擬碼？

- 虛擬程式碼是給人看的非正規語言，用來幫助程式設計師撰寫程式前，思考這個程式該如何撰寫。
- 當程式碼基礎知識越來越豐富之後，以後的描述將會漸漸少給完整的程式碼，改以虛擬程式碼的方式出現

# 將虛擬碼變成程式碼的方式



## 第一種流程

# 虛擬碼 → 程式碼

如果 成績  $\leq 100$  而且  $\geq 90$

輸出 『A』

如果 成績  $< 90$  而且  $\geq 80$

輸出 『B』

如果 成績  $< 80$  而且  $\geq 70$

輸出 『C』

如果 成績  $< 70$  而且  $\geq 60$

輸出 『D』

否則

輸出 『F』

```
import java.util.Scanner;
class web{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        }
        if (num < 90 && num >= 80) {
            System.out.printf("%d是B\n", num);
        }
        if (num < 80 && num >= 70) {
            System.out.printf("%d是C\n", num);
        }
        if (num < 70 && num >= 60) {
            System.out.printf("%d是D\n", num);
        } else {
            System.out.printf("%d是F\n", num);
        }
    }
}
```



## 第二種流程

# 虛擬碼 → 程式碼

如果 成績  $\leq 100$  而且  $\geq 90$   
輸出 『A』

否則

如果 成績  $\geq 80$   
輸出 『B』

否則

如果 成績  $\geq 70$   
輸出 『C』

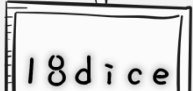
否則

如果 成績  $\geq 60$   
輸出 『D』

否則

輸出 『F』

```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        } else {
            if (num >= 80) {
                System.out.printf("%d是B\n", num);
            } else {
                if (num >= 70) {
                    System.out.printf("%d是C\n", num);
                } else {
                    if (num >= 60) {
                        System.out.printf("%d是D\n", num);
                    } else {
                        System.out.printf("%d是F\n", num);
                    }
                }
            }
        }
    }
}
```



```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        }
        if (num < 90 && num >= 80) {
            System.out.printf("%d是B\n", num);
        }
        if (num < 80 && num >= 70) {
            System.out.printf("%d是C\n", num);
        }
        if (num < 70 && num >= 60) {
            System.out.printf("%d是D\n", num);
        } else {
            System.out.printf("%d是F\n", num);
        }
    }
}
```

當兩種流程都轉成  
程式碼之後，是否  
可以分辨出兩種寫  
法的差異呢？

```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        } else {
            if (num >= 80) {
                System.out.printf("%d是B\n", num);
            } else {
                if (num >= 70) {
                    System.out.printf("%d是C\n", num);
                } else {
                    if (num >= 60) {
                        System.out.printf("%d是D\n", num);
                    } else {
                        System.out.printf("%d是F\n", num);
                    }
                }
            }
        }
    }
}
```



```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        }
        if (num < 90 && num >= 80) {
            System.out.printf("%d是B\n", num);
        }
        if (num < 80 && num >= 70) {
            System.out.printf("%d是C\n", num);
        }
        if (num < 70 && num >= 60) {
            System.out.printf("%d是D\n", num);
        } else {
            System.out.printf("%d是F\n", num);
        }
    }
}
```

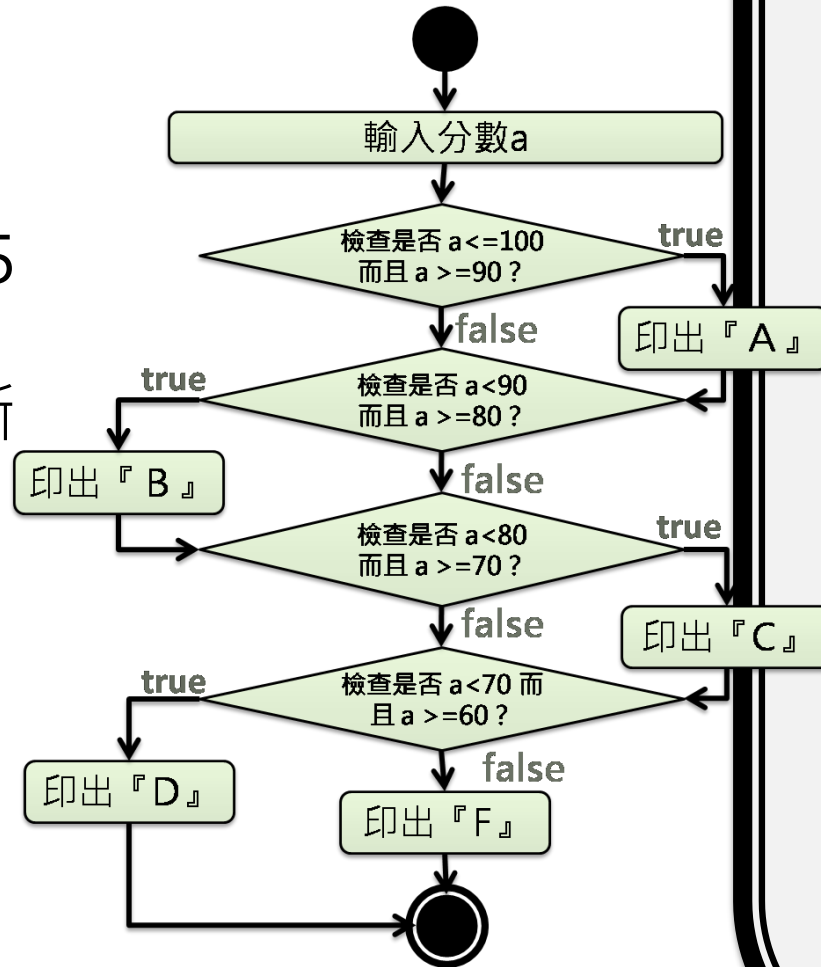
第二種寫法就是  
『巢狀選擇』。但  
兩種寫法有什麼優  
缺點呢？

```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        } else {
            if (num >= 80) {
                System.out.printf("%d是B\n", num);
            } else {
                if (num >= 70) {
                    System.out.printf("%d是C\n", num);
                } else {
                    if (num >= 60) {
                        System.out.printf("%d是D\n", num);
                    } else {
                        System.out.printf("%d是F\n", num);
                    }
                }
            }
        }
    }
}
```

『巢狀選擇』

# 第一種寫法

- 只有使用 if 的寫法
- 缺點：當分數是 95 時，在第一個 if 就會得到 true 的判斷結果，也就會印出『A』，但即使已經輸出資料了，仍會接著執行下面的三種 if 判斷。



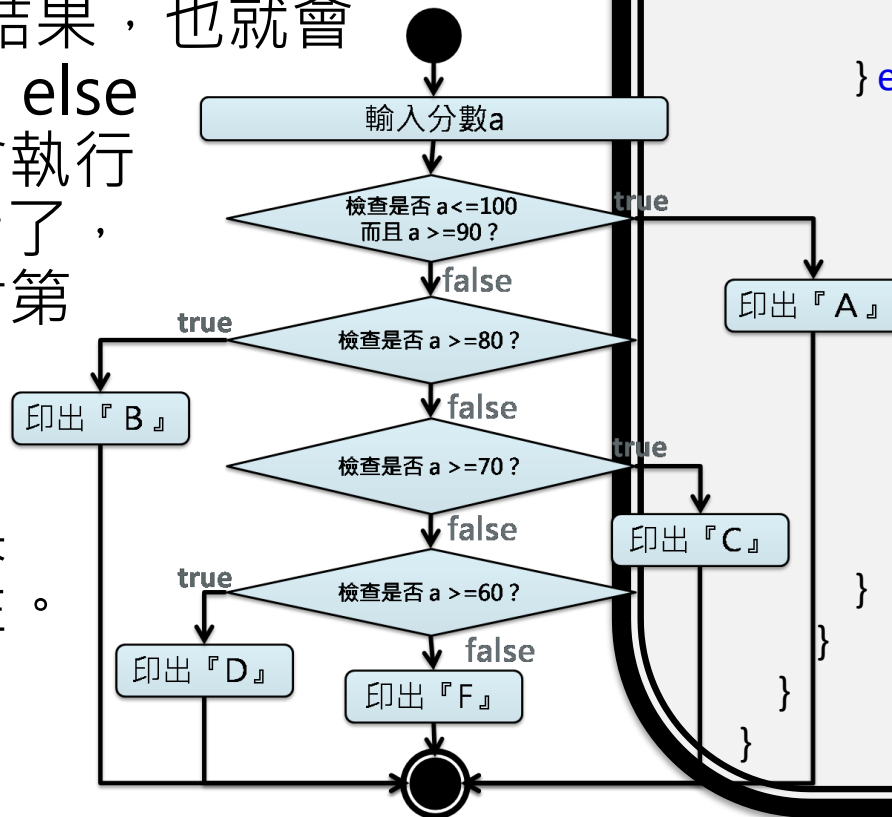
```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        }
        if (num < 90 && num >= 80) {
            System.out.printf("%d是B\n", num);
        }
        if (num < 80 && num >= 70) {
            System.out.printf("%d是C\n", num);
        }
        if (num < 70 && num >= 60) {
            System.out.printf("%d是D\n", num);
        }
        else {
            System.out.printf("%d是F\n", num);
        }
    }
}
```

# 第二種寫法

- if...else... 搭配巢狀選擇

- 優點：當分數是 95 時，在第一個 if 就會得到 true 的判斷結果，也就會印出『A』，下面的 else 就不會進入，更不會執行 else 內的其他if 判斷了，程式執行的效率相對第一種寫法會比較好。

- 缺點：太多層的結果會降低程式的可讀性。



```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        } else {
            if (num >= 80) {
                System.out.printf("%d是B\n", num);
            } else {
                if (num >= 70) {
                    System.out.printf("%d是C\n", num);
                } else {
                    if (num >= 60) {
                        System.out.printf("%d是D\n", num);
                    } else {
                        System.out.printf("%d是F\n", num);
                    }
                }
            }
        }
    }
}
```

```
import java.util.Scanner;
class web {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num;
        num = sc.nextInt();
        if (num <= 100 && num >= 90) {
            System.out.printf("%d是A\n", num);
        } else {
            if (num >= 80) {
                System.out.printf("%d是B\n", num);
            } else {
                if (num >= 70) {
                    System.out.printf("%d是C\n", num);
                } else {
                    if (num >= 60) {
                        System.out.printf("%d是D\n", num);
                    } else {
                        System.out.printf("%d是F\n", num);
                    }
                }
            }
        }
    }
}
```

if...else是一種『二分法』

採取二分再二分的方法

