程式設計（112-1）
作業十

作業設計：孔令傑
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS（`http://pdogs.ntu.im/`）為第一、二、三題上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳），並且到 NTU COOL 以線上輸入的方式輸入第四題的答案。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。這份作業的程式碼和書面報告的截止時間都是 ***11 月 28 日早上八點***。為這份作業設計測試資料並且提供解答的助教是陳廷旭。

在你開始前，請閱讀課本的第 12–13 章[1]。

本次作業滿分為 110 分，得幾分就算幾分。若整學期有 n 份作業，則學期的作業總成績即為 n 份作業的總分除以 n（不論超過 100 與否）。

第一題

（20 分）「東東動物園」全新開幕，園方引進了非常多的奇珍異獸，並且規劃了多樣的展示、表演活動。園方希望把這些繁雜的相關資料，系統性地建置在自動化查詢程式中，方便管理與掌握所有動物的狀況，因此請資訊與管理雙專長的你，幫忙撰寫程式來完成這個任務。

首先，我們會定義一個 `class Animal`，這個 `class` 大概會長得像下面這樣：

```
struct Date {
    int year;
    int month;
    int day;
};

class Animal {
protected:
    int id;
    Date birthday;
    string name;
public:
    Animal(int id, Date b, string n);
    Animal(int id, Date b);
    ~Animal();
    int getAge() const;
    string getName() const;
    void setName(string n);
    void print() const;
};
```

[1]課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

根據以上程式碼，`class Animal` 包含以下內容，

- `int id`：動物的唯一識別編號。
- `Date birthday`：動物的生日。
- `string name`：動物的名稱。
- `Animal(int id, Date b, string n)`：帶有編號、生日和名字的 constructor。
- `Animal(int id, Date b)`：只有編號和生日的 constructor，有一些動物可能剛出生還沒有取名。
- `~Animal()`：destructor。
- `int getAge()`：計算並回傳動物的年齡。年齡從 0 歲開始計算，並且在生日當天，即算做進入下一歲。舉例來說，如果於 2021/07/27 出生，那麼在 2022/07/26 及其之前都算做 0 歲，在 2022/07/27 時即算做 1 歲。
- `string getName()`：回傳動物的名字。
- `void setName(string n)`：設定（或更改）動物的名字。
- `void print()`：印出動物的基本資訊。請依序印出名字、生日（`YYYY/MM/DD`）以及年齡，兩兩之間以一個逗點隔開。

該園中許多的動物會在固定時間參與展示（例如無尾熊每天下午 3:00 到 5:00 間開放參觀，之類的），針對這些要被展示的動物們，我們進一步定義一個 `class DisplayAnimal`，它繼承自 `class Animal`。這個子類別用於特別表示那些參與展示的動物：

```
struct Time {
    int hour;
    int minute;
    int second;
};

class DisplayAnimal : public Animal {
private:
    Time start;
    Time end;
public:
    DisplayAnimal(int id, Date b, string n);
    DisplayAnimal(int id, Date b);
    ~DisplayAnimal();
    void setDisplayTime(const Time& start, const Time& end);
    void getDisplayTime(Time& start, Time& end) const;
    void print() const;
};
```

根據以上程式碼，`class DisplayAnimal` 包含以下內容：

- `Time start` 和 `Time end`：分別表示展示的開始和結束時間。
- `DisplayAnimal(int id, Date b, string n)`：帶有編號、生日和名字的 constructor。
- `DisplayAnimal(int id, Date b)`：只有編號和生日的 constructor，有一些動物可能剛出生還沒有取名。
- `~DisplayAnimal()`：destructor。
- `void setDisplayTime(const Time& start, const Time& end)`：設定動物展示的開始和結束時間。
- `void getDisplayTime(Time& start, Time& end)`：透過 call by reference 回傳動物展示的開始和結束時間。
- `void print()`：印出該展示動物的詳細資訊，依序為名字、生日（`YYYY/MM/DD`）、年齡、展示開始時間（`HH:mm:ss`）、展示結束時間（`HH:mm:ss`），兩兩之間以一個逗點隔開。如果該展示動物尚未被設定展示時間，則在年齡後方再印出一個逗點和「none」這個字串即可。

園方的管理人員會時不時地對你的程式發動多次行動，包含創建動物資料、設定（更改）動物資料等，且以上行動可能會交錯出現，最後則查詢動物資料。在每次遇到創建資料任務時，請根據給定的動物編號、動物生日、以及動物名稱（若有動物出生還沒有取名，會以「Unknown」表示），創建對應的 `DisplayAnimal` 物件。在每次遇到設定或更改資料任務時，請根據給定的動物名稱或編號，更改或設定展示時間、名字等等。在每次遇到查詢任務時，請先印出所有 `DisplayAnimal` 的數量，再根據給定的動物名稱，呼叫該指定動物物件的 `print()` 函數，印出對應的資訊。

此外，請注意也有可能會出現不合理的行動，像是如果某次行動針對叫做「Spotty」的動物，設定其展示時間為「14:00:00–15:00:00」，可是叫做「Spotty」的動物並不存在，那麼請直接認定這筆行動無效，忽略這一筆行動，換而言之，我們只需要執行所有「有效」的行動就可以了。另外，創建動物資料時指定的動物編號也有可能不按照大小順序排列，也就是有可能先創建了編號為 8 的動物資料，再創建編號為 2 的動物資料。最後，每個動物都可以被改名多次，而每個動物在任何時候名稱都不會重複。

最後，雖然這一題不寫 `class` 也寫得出來，但後面還有其他題目是從此題做延伸，屆時若沒有 `Animal` 類別、`DisplayAnimal` 類別，寫起來會更痛苦，所以我們建議在這一題就花點功夫熟悉語法，並且幫你的程式建立良好的結構。上方的 `struct` 和 `class` 純屬範例，你可以照著延伸，也可以自行修改（例如把 `struct` 改成 `class`、幫 `struct` 寫成員函數、幫 `class` 新增成員函數等等都可以），不用一定要跟範例一模一樣。

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $t+2$ 行：

- 第一行包含一個整數 t，代表總共對於這個系統執行了多少次行動（包含創建、設定等等）。
- 第二行起的 t 行中，包含三種資訊。若該行代表一筆創建紀錄，則該行會依序包含一個大寫字元「C」、一個空白字元、一個整數 i 代表該動物的編號（id）、一個空白字元、一個字串代表動物的名稱（尚未取名的動物一律以「Unknown」表示）、一個空白字元、一個「`YYYY/MM/DD`」格式的字串代表動物的生日日期。若該行代表一次設定紀錄，則該行會依序包含一個大寫字元「S」、一個

空白字元、一個字串代表動物名稱、一個空白字元、一個「HH:mm:ss」格式的字串代表欲設定的展示開始時間、一個空白字元、一個「HH:mm:ss」格式的字串代表欲設定的展示結束時間。若該行代表一次命名紀錄，則該行會依序包含一個大寫字元「N」、一個空白字元、一個字串代表該動物的編號、一個空白字元、一個字串代表欲更改的名稱，且已知該編號一定存在。

- 最後一行代表一筆查詢，該行會依序包含一個大寫字元「Q」、一個空白字元、一個字串代表欲查詢的動物的名稱，且已知該動物名稱一定存在。

已知所有 DisplayAnimal 的數量為 n，且 $1 \leq n \leq 100$；姓名字串只包含大小寫英文字元，且長度不超過 20 個字元，也不會有任何動物被命名為「Unknown」；$1 \leq t \leq 10000$；給定的動物編號不會重複且介於 1 至 100 之間。在各種活動的資料正確性方面，可能的錯誤有三種：(1) 設置展示時間時對象是不存在的動物；(2) 展示開始時間或結束時間不在 00:00:00 和 23:59:59 之間（例如 28:00:00 或 16:98:23）；(3) 結束時間早於開始時間。遇到這三種情況時，請直接跳過該項活動。

讀入這些資訊後，請執行 t 次行動，在每次行動裡依照指定的任務執行，最後依序在第一行印出所有 DisplayAnimal 的數量，接著執行最後一行查詢行動中，指定的動物的 print() 函數，並將結果印出。計算年齡時，請設定程式執行的日期為 2023/11/21，且已知所有動物的生日都不晚於 2023/11/21。舉例來說，如果輸入是

```
8
C 1 Unknown 2023/05/08
C 2 Cutie 2023/09/15
S Cutie 14:00:00 16:00:00
S John 14:00:00 15:00:00
S Cutie 14:00:00 15:00:00
S John 14:00:00 15:00:00
N 1 John
S John 11:00:00 14:00:00
Q Cutie
```

則輸出應該是

```
2
Cutie,2023/09/15,0,14:00:00,15:00:00
```

如果輸入是

```
2
C 5 Cutie 2022/11/21
S Cutie 14:00:00 16:99:99
Q Cutie
```

則輸出應該是

```
1
Cutie,2022/11/21,1,none
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。

第二題

（20 分）承上題，動物園中，除了會參與展示的動物（也就是 `DisplayAnimal`），也有會參與表演的動物。目前為了動物的身心健康，每一種動物都只會參與展示或參與表演，並不會有動物同時參與展示與表演的過勞情況。

針對會參與表演的動物，我們進一步定義一個 `class ShowAnimal`，它繼承自 `class Animal`。這個子類別用於特別表示那些參與表演的動物：

```
class ShowAnimal : public Animal {
private:
    Date** showDates;
    int showCnt;
public:
    ShowAnimal(int id, Date b, string n);
    ShowAnimal(int id, Date b);
    ShowAnimal(const ShowAnimal& sa);
    ~ShowAnimal();
    void addShowDate(const Date& d);
    int getShowCnt(const Date& start, const Date& end) const;
    void print(const Date& start, const Date& end) const;
    void print() const;
};
```

根據以上程式碼，`class ShowAnimal` 包含以下內容：

- `Date** showDates`：動態儲存過往表演的日期，其中 `showDates` 指向一個動態陣列，而陣列中的每個元素都是一個指向 `Date` 物件的指標。
- `int showCnt`：記錄動物參與表演的總次數。
- `ShowAnimal(int id, Date b, string n)`：帶有編號、生日和名字的 constructor，其中表演次數應被初始化為 0。
- `ShowAnimal(int id, Date b)`：只有編號和生日的 constructor，有一些動物可能剛出生還沒有取名，其中表演次數應被初始化為 0。

- `ShowAnimal(const ShowAnimal& sa)`：copy constructor。
- `~ShowAnimal()`：destructor。
- `void addShowDate(const Date& d)`：用於添加一個新的表演日期，具體實作方式沒有限制。
- `int getShowCnt(const Date& start, const Date& end)`：用於計算並回傳指定起訖日期（包含這兩個日期）間的表演次數。
- `void print(const Date& start, const Date& end)`：用於印出表演動物的詳細資訊，依序為名字、生日（`YYYY/MM/DD`）、年齡、以及在指定起訖日期（包含這兩個日期）間的表演次數，兩兩之間以一個逗點隔開。
- `void print()`：用於印出表演動物的詳細資訊，依序為名字、生日（`YYYY/MM/DD`）、年齡、以及有史以來的所有表演次數，兩兩之間以一個逗點隔開。

跟第一題一樣，園方的管理人員會時不時地對你的程式發動多次行動，包含創建動物資料、設定（更改）動物資料等，且以上行動可能會交錯出現，最後則查詢動物資料。在每次遇到創建資料任務時，請根據給定的動物編號、動物生日、以及動物名稱（若有動物出生還沒有取名，會以「Unknown」表示），創建對應的 `DisplayAnimal` 或 `ShowAnimal` 物件。在每次遇到設定或更改資料任務時，請根據給定的動物名稱或編號，更改或設定展示時間、名字等等。在每次遇到查詢任務時，請根據給定的表演動物名稱，先印出所有 `DisplayAnimal` 的數量以及所有 `ShowAnimal` 的數量，再呼叫該指定動物物件的 `print(const Date& start, const Date& end)` 函數，印出對應的資訊。

此外，請注意也有可能會出現不合理的行動，像是某次行動針對叫做「Spotty」的動物，設定其表演日期為「2023/11/21」，可是叫做「Spotty」的動物並不存在，那麼請直接認定這筆行動無效，忽略這一筆行動，換而言之，我們只需要執行所有「有效」的行動就可以了。另外，創建動物資料時指定的動物編號也有可能不按照大小順序排列，並且在表演及展示的動物之間，編號也不會重複，也就是有可能先創建了編號為 8 的動物資料，再創建編號為 2 的動物資料。最後，每個動物都可以被改名多次，而每個動物在任何時候名稱都不會重複。

跟第一題不一樣的是，創建資料任務會指定創建的動物類別是 Display 還是 Show。此外，因應 `ShowAnimal` 的類別，我們多新增了一個添加的行動，在每次遇到添加資料任務時，請根據給定的動物名稱、表演日期資料，添加對應的表演日期資料。

最後，雖然這一題一樣不寫 `class` 也寫得出來，不過大家還是盡力試試看吧！

輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $t+2$ 行：

- 第一行包含一個整數 t，代表總共對於這個系統執行了多少次行動（包含創建、設定、添加等等）。
- 第二行起的 t 行中，包含三種資訊。若該行代表一筆創建紀錄，則該行會依序包含一個大寫字元「C」、一個空白字元、要創建的動物類別、一個空白字元、一個整數 i 代表該動物的編號（id）、一個空白字元、一個字串代表動物的名稱（尚未取名的動物一律以「Unknown」表示）、一個空白字元、一個字串代表動物的生日日期，並以「`YYYY/MM/DD`」表示。若該行代表一次展示動物的設定紀錄，則該行會依序包含一個大寫字元「S」、一個空白字元、一個字串代表展示動物名稱、

一個空白字元、一個「HH:mm:ss」格式的字串代表欲設定的展示開始時間、一個空白字元、一個「HH:mm:ss」格式的字串代表欲設定的展示結束時間。若該行代表一次命名紀錄，則該行會依序包含一個大寫字元「N」、一個空白字元、一個字串代表該動物的編號、一個空白字元、一個字串代表欲更改的名稱，且已知該編號一定存在。若該行代表一次表演動物的添加紀錄，則該行會依序包含一個大寫字元「A」、一個空白字元、一個字串代表表演動物名稱、一個空白字元、一個字串代表欲添加的表演日期，並以「YYYY/MM/DD」表示。

- 最後一行代表一筆針對表演動物的查詢，該行會依序包含一個大寫字元「Q」、一個空白字元、一個字串代表欲查詢的表演動物的名稱、一個「YYYY/MM/DD」格式的字串代表起始日期、一個「YYYY/MM/DD」的字串代表結束日期（結束日期一定晚於起始日期），且已知該表演動物名稱一定存在。

已知所有 DisplayAnimal 的數量為 n，且 $1 \leq n \leq 100$；所有 ShowAnimal 的數量為 m，且 $1 \leq m \leq 100$；姓名字串只包含大小寫英文字元，且長度不超過 20 個字元，也不會有任何動物被命名為「Unknown」；$1 \leq t \leq 10000$；給定的動物編號不會重複且介於 1 至 200 之間；給定的表演日期一定是合理的日期（例如不會出現 2023/2/29 或 2023/14/5 這種日期）。在各種活動的資料正確性方面，可能的錯誤有五種：(1) 設置展示時間時對象非展示動物或不存在；(2) 添加表演日期時對象非表演動物或不存在；(3) 展示時間不在 00:00:00 和 23:59:59 之間（例如 28:00:00 或 16:98:23）；(4) 展示結束時間早於開始時間；(5) 要幫一個動物新增的表演日期和這個動物既有的表演日期重複（一個動物一天只能表演一次）或早於其出生日期。遇到這五種情況時，請直接跳過該項活動。

讀入這些資訊後，請執行 t 次行動，在每次行動裡依照指定的任務執行，最後依序在第一行依序印出所有 DisplayAnimal 的數量以及所有 ShowAnimal 的數量，兩兩之間以一個逗點隔開。接著根據指定的年份，執行最後一行查詢行動中指定的動物的 print(const Date& start, const Date& end) 函數，並將結果印出。計算年齡時，請設定程式執行的日期為 2023/11/21，且已知所有動物的生日都不晚於 2023/11/21。舉例來說，如果輸入是

```
9
C Show 3 John 2022/05/08
C Show 1 Unknown 2022/05/08
C Display 8 Cutie 2023/09/15
S Cutie 14:00:00 16:00:00
N 1 Spotty
A Spotty 2023/01/05
A Spotty 2023/05/09
A Spotty 2022/11/05
S Spotty 14:00:00 16:00:00
Q Spotty 2023/01/01 2023/12/31
```

則輸出應該是

```
1,2
Spotty,2022/05/08,1,2
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你可以使用任何方法。

評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。

第三題

（50 分）承上題，為了提高這套動物園管理系統的可維護性，我們將實作一個類別 `Zoo`，大概長得像下面這樣：

```
const int ZOO_CAP = 100;
class Zoo {
private:
    int displayAnimalCnt;
    int showAnimalCnt;
    Animal* animals[ZOO_CAP];
public:
    Zoo();
    ~Zoo();
    bool addAnimal(const Animal* a, bool isShowAnimal);
    void print() const;
    Animal* findMostBusyShowAnimal();
};
```

根據以上程式碼，`class Zoo` 包含以下內容：

- `int displayAnimalCnt` 和 `int showAnimalCnt`：分別用於儲存展示和表演的動物數量。
- `Animal* animals[ZOO_CAP]`：一個 `Animal` 類別物件的指標陣列，用於儲存動物園內的所有動物的資訊。陣列大小由 `ZOO_CAP`（動物園容量）定義。
- `Zoo()`：constructor，其中 `animals` 的所有元素都應該被初始化為 `nullptr`。
- `~Zoo()`：destructor，透過 `animals` 裡的指標去 `delete` 動物物件所佔的記憶體空間。
- `bool addAnimal(const Animal* a, bool isShowAnimal)`：用於向動物園中添加一個新的動物，並傳入 `isShowAnimal` 來表示添加的動物是否為表演動物。如果動物園容量已經滿了，則不新增並且回傳 false；反之則新增此動物的指標到 `animals` 中還是 `nullptr` 且索引值最小的元素，並且回傳 true。

- `void print()`：用於印出動物園中所有動物的資訊。請依序先在第一行依序印出展示動物和表演動物的總數量；接著在第二行起的 `displayAnimalCnt + showAnimalCnt` 行依照動物被新增的順序，呼叫每一個動物的 `print()` 函數並印出相關資訊。

- `Animal* findMostBusyShowAnimal()`：找出在 `ShowAnimal` 類別中總表演次數最多的表演動物。檢查 `animals` 陣列中的每一個指標指到的 `ShowAnimal` 物件（如果是 `DisplayAnimal` 就跳過），並回傳表演次數最多的表演動物[2]。如果平手，就回傳編號較小的。

請完整實作以上 `class Zoo`，並在讀取並執行所有輸入資料後，最後執行 `class Zoo` 的 `print()` 函數，接著再執行 `Animal* findMostBusyShowAnimal()`，獲得回傳結果後，再執行該動物的 `print()` 函數並印出結果。雖然照理說應該要實作 copy constructor，但為了減輕大家負擔，這題就不要求了。

特別提醒：在本題的測試資料中，如果使用兩個陣列來分別儲存展示動物和表演動物，會在程式碼品質方面被扣分。

輸入輸出格式

系統會提供一共 15 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有 $t+1$ 行，其格式與第二題的前 $t+1$ 行完全相同，但 $1 \le t \le 20000$。

讀入這些資訊後，請執行 t 次行動，在每次行動裡依照指定的任務執行，最後依序在第一行執行 `Zoo` 的 `print()` 函數，接著再執行 `Animal* findMostBusyShowAnimal()`，獲得回傳結果後，再執行該動物的 `print()` 函數並印出結果。計算年齡時，請設定程式執行的日期為 2023/11/21，且已知所有動物的生日都不晚於 2023/11/21。舉例來說，如果輸入是

```
8
C Show 1 Unknown 2022/05/08
C Display 8 Cutie 2023/09/15
S Cutie 14:00:00 16:00:00
N 1 Spotty
A Spotty 2023/01/05
A Spotty 2023/05/09
A Spotty 2022/11/05
C Display 3 John 2023/09/15
```

則輸出應該是

```
3
Spotty,2022/05/08,1,3
Cutie,2023/09/15,0,14:00:00,16:00:00
John,2023/09/15,0,none
Spotty,2022/05/08,1,3
```

如果輸入是

[2]如果需要判斷 `animals` 陣列裡的指標指向的物件類別，一個作法是擴充 `Animal` 去增加一個 instance variable 來記錄是否為表演動物。

```
14
C Show 1 Ted 2020/02/02
C Show 10 Unknown 2021/02/01
C Display 3 Coco 2023/06/21
C Show 6 Kiki 2022/03/02
A Kiki 2023/01/05
A Ted 2021/11/05
N 10 Peter
N 10 Flora
S Coco 13:00:00 14:00:00
A Ted 2023/05/09
A Kiki 2023/01/05
A Flora 2023/01/07
A Ted 1998/11/05
A Kiki 2023/01/06
```

則輸出應該是

```
4
Ted,2020/02/02,3,2
Flora,2021/02/01,2,1
Coco,2023/06/21,0,13:00:00,14:00:00
Kiki,2022/03/02,1,2
Ted,2020/02/02,3,2
```

你上傳的原始碼裡應該包含什麼

你的 .cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你***不可以***使用上課沒有教過的方法：

- 確定可以使用的語法包含繼承、多型，以及在之前的作業中正面表列過的語法。
- 確定不可以使用的語法包含 `printf`、`scanf`、template、`<vector>`、例外處理等等。

請注意正面表列的固然是都確定可以用，但沒有被負面表列的不表示可以用喔！

評分原則

- 這一題的其中 30 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。
- 這一題的其中 20 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的結構、運算邏輯、可讀性（包含排版、變數命名、註解等等）、可擴充性、模組化程度，以及是否使用了還沒教過的語法。請寫一個「好」的程式吧！

第四題

特別說明：

- 本題為作文題而非程式題，請不要上傳東西到 PDOGS。由於班上同學很多，但助教人力有限，所以原則上我們會從所有繳交中隨機批改約 50%。沒被抽到的同學在本題就會得到滿分，但不會得到直接的助教回饋；有被抽到的同學們可能會被扣分，但相對應地也會得到助教的回饋。
- 本題請大家在 NTU COOL 上直接輸入答案，而非繳交 PDF 檔。由於直接輸入的是純文字，大家可以不用費心排版、調整格式等等。但如果你的敘述會有數個段落，請用空行區隔段落，而各段開頭不用（不要）手動縮排。

（20 分；每小題 10 分）請回答以下兩題。在回答時，都不用寫程式碼，只要文字敘述即可。

1. 如果在第三題的程式中，你要允許系統中出現展示動物和表演動物，且每個動物都要是展示動物和表演動物其中之一，不能有未定義其類型的動物，你應該怎樣修改 `Animal` 這個類別？
2. 如果在第三題的程式中，你要允許系統中出現既要展示又要表演的動物，並且不能使用多重繼承（multiple inheritance），你該怎麼修改你的程式？你要新增一個類別，還是修改既有類別？請列出幾種作法，從中選擇一種，並且說明你為什麼覺得你的作法是好的。