

資料結構與進階程式設計 (107-2)

程式作業一

作業設計：孔令傑

國立臺灣大學資訊管理學系

繳交作業時，前兩題請以英文或中文於紙本作答。不接受遲交。請將答案以 PDF 檔上傳到 NTU COOL；**不接受紙本繳交**；第三題請至 PDOGS (<http://pdogs.ntu.im/judge/>) 上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。這份作業的截止時間是 **2019 年 3 月 11 日凌晨 1:00**。在你開始前，請閱讀課本的第 8 和 18 章¹。

由於本課程修課同學眾多，而助教人力有限，為了避免過度壓榨助教，我們可能會對手寫題進行隨機批改，亦即助教會隨機選擇一份作業的若干題進行批改和給分。助教會對全班同學都批改同樣的題目。若有批改之題目的總分為 x 而一位同學得到 y 分，且手寫題總分為 z 分，則其在該份作業會得到 $z(\frac{y}{x})$ 分。公佈的參考解答中會包含所有題目（包含沒被批改的）。

第一題

（20 分）請簡要地回答以下各題：

- (a)（5 分）請說明 `<string>` 函式庫中的 `size_t` 是什麼，以及不使用它的話會有什麼缺點。
- (b)（5 分）請說明 `<fstream>` 函式庫中的 `EOF` 是什麼，以及如何利用它來協助我們讀取檔案。
- (c)（10 分）講 C++ string 的那次課堂練習，我們做了 Problem 3，是關於 Dollar Format 的，題目的第一句是「When we have an integer value, we print it out directly. However, we should “decorate” it if it is representing an amount of money.」。針對此題，請利用 C++ string 的 `insert()` 函數來實做一個能完成任務的程式（包含合適的註解），並貼上你的程式碼做為答案。

第二題

（20 分；每小題 10 分）請就以下各小題，做簡要的回答。你當然可以一次回答這兩題，也就是在 (b) 小題寫「承上題...」；你也可以兩題完全分開。

- (a) 請說明 C++ 中「繼承」(inheritance) 的概念、其如何運作，並且自行舉一個適合利用繼承來實現的類別間關係。你應該先說明一個程式任務（像是本次作業第三題的船舶進出港口排班），接著說明該程式中應該出現的類別有哪些、各類別應該包含什麼屬性、你建議的繼承關係為何，以及最重要的，如果不使用繼承關係會有什麼程式開發上的壞處，以及使用繼承關係會有什麼程式開發上的好處。
- (b) 請說明 C++ 中「多型」(polymorphism) 的概念、其如何運作，並且自行舉一個適合利用多型來實現的功能。你應該先說明一個程式任務（像是本次作業第三題的船舶進出港口排班），接著說明該程式中應該出現的類別有哪些、各類別應該包含什麼屬性、你建議的繼承關係為何，以及你建

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

議利用多型實現的功能及其實作方式（可能是一個函數、一個陣列，或其他東西）。同樣地，你得要說明不用多型的壞處，以及使用多型的好處。

第三題

（60 分）在此題，我們想要來實作船舶進出港口的排班規劃演算法。已知某港乃國際大港，共有兩個港口（port）可以讓行駛船隻做進出港使用，分別為一號港與二號港。此外，共有 m 個碼頭（pier）提供船舶得以在港內停靠。在該港口有登記且可能進出的共有 s 條船舶（ship），一天的船舶移動工作（job）有 n 個。

在給定船舶資訊與每日工作資訊後，港口需要安排船舶進出港口以及在港內移動的時間。我們將撰寫排班程式，分別實踐以下兩種排班演算法：Earliest Ready Time First（ERT）和 Shortest Processing Time First（SPT），並計算出在兩種演算法下所有工作的延遲時間總和。

針對每條船舶，我們記載以下資訊：

- 船舶編號（最長六個英數字元，皆大寫）。
- 船舶重量（單位為噸，是介於 0 和 100000 之間的整數）。
- 國籍（最長五十個英文字元，可能含有空白，皆大寫）。
- 船長姓名（最長五十個英文字元，可能含有空白，皆大寫）。
- 是否裝載危險貨品（Y 或 N，Y 表示有，N 表示沒有）。

船舶資訊被以純文字檔記錄，記錄時每一列儲存一條船舶的資訊，將上述五個欄位用四個逗點隔開之後依序儲存。舉例來說，以下是一批船舶的資訊：

AFECXS,25500,TAIWAN,LING KUNG,Y
AFEUTI,20000,JAPAN,HIRONA OISHI,N
TRUCXS,13700,THAILAND,SAWADIKA,Y
THUEDC,18600,TAIWAN,YENYENER,N
KCUHIC,78600,THAILAND,KAPUNKA,Y

其中第一條是編號為 AFECXS、重 25500 噸、國籍為 TAIWAN、船長姓名為 LING KUNG、有載危險貨品的船舶。所有船舶的編號都不會一樣。

針對每個工作，我們記載以下資訊：

- 船舶編號（符合上述船舶編號的格式）。
- 可出發時間（格式為 HH:MM 的字串，HH 表示幾點，MM 表示幾分）。
- 作業時間（單位為分鐘，為介於 0 和 200 之間的整數）。
- 類型（為 I、T 或 O 三個字元其中之一，分別表示進港、移泊、出港）。
- 港口（為 1 或 2 兩個數字其中之一，表示進出的港口，只有進港或出港工作才有這個資訊）。

- 起點碼頭（為介於 1 和 m 之間的整數，表示從哪個碼頭出發，只有出港和移泊的工作才有）。
- 終點碼頭（為介於 1 和 m 之間的整數，表示要前往哪個碼頭，只有進港和移泊的工作才有）。

工作資訊被以純文字檔記錄，記錄時每一列儲存一個工作的資訊，將上述七個欄位（的其中六個）用五個逗點隔開之後依序儲存，其中船舶編號一定會對應到恰好一條記載在船舶資訊中的船舶。舉例來說，以下是一批工作的資訊：

```
AFECXS,14:25,46,I,1,120
AFEUTI,15:25,12,T,106,120
TRUCXS,12:25,87,0,2,17
```

其中第一列表示剛剛 AFECXS 那條船舶，將在 14:25 的時候做好進港的準備，其進港作業從起錨到通過一號港到在 120 碼頭停好，共需 46 分鐘。實務上一條船舶一天可能移動數次，但為了簡單起見，本題中我們假設每條船舶在一天內只會移動一次。

跟世界上大部份的港口一樣，一個港口（一號港或二號港）只有一條供船舶進出的水道，水道不會開太寬（不然風浪就進來了），所以一次只能通過一條船舶。雖然實務上船舶們可以魚貫而入（或出），但為了這題簡單起見，我們假設每一條通過某水道船舶都要完整地停好，才能換下一條船舶開始使用該水道。港內的水道則是很寬，所以包括所有移泊工作在內的港內船舶移動都不需要互相等待。從一港口進出的船舶之間不能互相衝突，從二港口進出的船舶之間也不能互相衝突，而一條從一港口進或出的船舶當然不會跟另一條從二港口進或出的船舶衝突。最後，一個船舶移動的工作一旦開始就不能被中斷或切割，亦即如果要安排上述 AFECXS 在 12:25 開始作業，則從 12:25 到 13:11 之間都不能有任何要使用一號港的船舶開始作業。

在滿足以上假設的前提下，我們將針對給定的船舶與工作資訊，為每個工作安排其「開始時間」。如果可以，我們當然希望所有工作都一準備好就出發，亦即其「延誤時間」為零。如果有工作的開始時間晚於其可開始時間，則其延誤時間即為其開始時間減可開始時間。我們想要安排工作，以最小化所有工作的總延誤時間。

在本題中，我們將請你實做指定的兩個演算法。兩個演算法都是先用各自的規則將所有工作排序，然後依序幫每個工作排定開始時間在「不早於其可出發時間」的「可行的最早時間」。這兩演算法的差異之處僅在於工作優先權排序條件的不同。演算法一是 **Earliest Ready Time First (ERT)** 演算法。針對要通過某一港口（一號港或二號港）的所有工作，我們將工作依照可出發時間由早至晚排序（亦即誰先準備好誰優先），若平手則將作業時間較短者排在前面，再平手則將船舶編號在前者排在前面。演算法二是 **Shortest Processing Time First (SPT)** 演算法。針對要通過某一港口（一號港或二號港）的所有工作，我們將工作依照作業時間由小到大排序（亦即誰佔用進出水道的時間短就誰優先），若平手則將可出發時間較早的排在前面，再平手則將船舶編號在前者排在前面。用船舶編號排序時，請以 C++ 標準函式庫的 `<string>` 的 `string::compare()` 排的先後順序為準。

如上所述，工作不可切割，且應被安排在可行的最早時間。舉例來說，根據演算法優先權排班法則，若已有兩艘船分別排定在 13:00 到 13:45 之間與 14:00 到 14:45 之間來使用一號港水道，若下一艘被安排的工作的作業時間為 30 分鐘，那麼即使其可出發時間為 12:31，其被排定的開始時間也會是 14:45，因為 12:31 到 13:00 的 29 分鐘，以及 13:45 到 14:00 的 15 分鐘，都不足以完成該工作。

給定船舶與工作資訊後，請利用兩個演算法各做一次工作排程，並計算個別演算法得到的總延誤時間。

輸入輸出格式

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中會有若干行，每一行會有一個整數 m ，表示該港的碼頭數量，第二行會有一個檔案路徑的字串，表示船舶資訊檔案的存放位置，第三行也會有一個檔案路徑的字串，表示工作資訊檔案的存放位置。在船舶資訊檔案中總共會有 s 行，在工作資訊檔案中總共會有 n 行，兩個檔案儲存資訊的方式則如上述。

在依序讀入這些資訊後，分別用兩個演算法（ERT、SPT）各自計算所有工作之延遲時間的總和，再依序列出演算法一（ERT）的總延遲時間與演算法二（SPT）的總延遲時間，以分鐘為單位，中間以一個空格作分開。

舉例來說，假設 ShipSample.txt 為船舶資訊檔案，其內容為

```
AFECXS,25500,TAIWAN,LING KUNG,Y
AFEUTI,20000,JAPAN,HIRONA OISHI,N
TRUCXS,13700,THAILAND,SAWADIKA,Y
THUEDC,18600,TAIWAN,YENYENER,N
KCUHIC,78600,THAILAND,KAPUNKA,Y
ITCUOW,12000,TAIWAN,JASON WANG,Y
ZCHCBE,88888,JAPAN,KANAKO,N
```

而 WorkSample.txt 為工作資訊檔案，其內容為

```
AFECXS,14:25,46,I,1,120
AFEUTI,15:25,12,T,106,120
TRUCXS,12:25,50,0,2,17
KCUHIC,12:25,75,I,2,150
ZCHCBE,13:00,50,0,2,35
```

如果輸入是

```
200
ShipSample.txt
WorkSample.txt
```

則輸出應該是

```
140 115
```

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**使用作業截止前在本課程中沒有教過的方法。

針對本題，我們要強迫你撰寫 class。很自然的一種設計是幫船舶與工作各設計一個 class，然後在程式中生成船舶與工作物件去記錄相關資訊。雖然在本題中，每個船舶最多出現在一個工作中，但在未來的作業中，一個船舶可能會出現在多個工作中，但你顯然不會想要在牽涉到同一船舶的數個工作中都

重複地產生一樣的船舶物件。相較之下，在程式中建立單一的船舶物件，並讓工作物件中有一個指向船舶物件的指標，或許是個值得你認真考慮的好主意（但你若真的覺得其他寫法更好，也不用勉強）。在你學了繼承與多型之後，你顯然也應該考慮設計一個 `parent class` 是工作，然後寫三個 `child class` 是進港、出港、移泊工作，這將讓你不用在一個單一的工作類別中有用不到的欄位，也會讓你可以寫一個以抽象工作為參數型態的函數，就能接受傳入三種具體工作物件。總之，想想看怎樣利用上課教的技術，讓你開發程式的過程更有效率，也讓你開發出來的系統更模組化、更有擴充性吧！

評分原則

- 這一題的其中 50 分會根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、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 10 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，而評分重點放在你的 `class` 設計（其 `member function` 是否合理有效地保護了 `member variable`）以及 `class` 之間的關係（特別是如何有效率地讓工作可以參照到船舶）。