資料結構與進階程式設計（108-2）
程式作業五

作業設計：孔令傑
國立臺灣大學資訊管理學系

此次作業的兩題請至 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 ***2020 年 4 月 14 日早上八點***。在你開始前，請閱讀課本的第 3 和 4 章[1]。為這份作業設計測試資料並且提供解答的是***盧慶原***。

新類別：背包與道具

在過往的課程影片中老師有提到過 `Character` 的類別，以及繼承 `Character` 的 `Warrior` 和 `Wizard`。在此題中，我們將擴充這個程式（你可以用老師的範例程式碼來進行擴充，也可以從頭開始寫一個新的），給予遊戲中的每個角色一個背包，裡面可以存放他擁有的道具。我們將實作一個背包類別 `ItemBag` 與一個道具類別 `Item`。請利用本週課程影片教的 bag 資料結構概念，案照以下規格來實作。

首先，每一個 `Item` 將包含以下 instance variable：

- `string name`：該道具的名稱，不同的道具其名稱必不同。
- `int type`：該道具的類型。若為 1，表示該道具是所有角色都可以使用的；若為 2，只有戰士可以使用；若為 3，只有巫師可以使用。

接下來，每一個角色都會擁有一個背包，背包中可以裝任何道具，但每一個背包都有給定的容量上限，會在角色加入隊伍時被給定。為了方便使用 `ItemBag`，我們設計以下 instance function：

- `void addItem(Item i)`：傳入一個道具 `i`，若未超過容量上限就將 `i` 放入此背包中，若容量已滿則拋出例外（throw exception）。
- `bool hasItem(Item i)`：針對傳入的道具 `i`，請尋找此背包中是否已經有一個或多個 `i`，若有則回傳 `true`，反之則回傳 `false`。
- `string getItemList()`：請將背包中的所有道具的名稱依照 `string::compare()` 判定的大小順序依序列出來，每個道具名稱間以逗號隔開。舉例來說，假設背包中依序曾被放入 `sword`、`potion`、`herb`、`herb`、`herb`、`potion`，則此函數應回傳一個其值為

  `herb,herb,herb,potion,potion,sword`

  的 C++ string 物件。若該背包中沒有任何道具，則回傳其值為 `empty` 的 C++ string 物件。
- `string getItemSummary()`：請計算並列出背包中的各種道具的個數，將道具名稱與個數間以一個逗號隔開，不同道具間以分號隔開，印出道具的順序也是依據 `string::compare()` 針對道具名稱判定的大小順序。承上例，此函數應回傳一個其值為

  `herb,2;potion,3;sword,1`

[1]課本是 Carrano and Henry 著的 *Data Abstraction and Problem Solving with C++: Walls and Mirrors* 第六版。

的 C++ string 物件。若該背包中沒有任何道具，則回傳其值為 `empty` 的 C++ string 物件。

- `void useItem(Item i)`：表示該角色使用了道具 `i`，因此原則上應將道具 `i` 的個數減一（為了簡單起見，我們不考慮使用道具後的效果，只是單純讓道具變少），但因為道具的類型會限制使用的對象，所以在一個角色使用該道具前，必須判斷他是否有使用的權限。若戰士想要使用類型 3 的道具或巫師想要使用類別 2 的道具 Item，則該道具的數量應維持不變，且函數應拋出例外。

  注意：上述關於一個角色是否可以使用一個物品的判斷，應該要實作在 `Warrior` 和 `Wizard` 裡面，而非在 `ItemBag` 中，這是因為背包本身只是裝道具而已，能否使用一個道具是角色的問題，不是背包的問題。

- `void removeItem(Item i)`：請將 `i` 從背包中移除一份（只移除一份，不是全部），若背包中沒有任何一份 `i`，則拋出例外。

在你的 `ItemBag` 中必須包含以上函式，但除了上述函式之外，你也可以設計任何你覺得需要用到的屬性或函式[2]。當然，基於 ADT 的基本設計原則，你自己設計的屬性或函式應該是 private 的。

事件

你的程式將會處理許多的事件，每個事件會以一列字串的方式輸入，每一列的第一個字元為事件代表字元，表示該事件的類型。以下將利用一些簡單的範例介紹各種事件。在所有的事件字串中，都以一個空白字元隔開相鄰的兩個資訊。

- 事件代表字元為 R 表示新增一個戰士；事件代表字元為 D 表示新增一個巫師。接下來，將依序出現一個字串以及兩個數字。字串表示該角色的名字，第一個數字表示該角色的等級，第二個數字表示該角色的背包的容量。例如

  ```
  R Alice 10 5
  D Bob 8 3
  ```

  以上兩個事件分別新增了一個叫做 Alice 的戰士以及叫做 Bob 的巫師，他們的等級分別是 10 和 8，背包的 capacity 分別是 5 和 3。另外，因為角色不能重複，所以使用者不能新增一個已經存在的角色。因次，若第三個事件為

  ```
  R Alice 8 3
  ```

  像這種嘗試新增已經加入隊伍的相同姓名角色（不論職業），請輸出

  ```
  the character exists
  ```

- 事件代表字元為 A 表示某角色新增了一個道具，後面將依序出現角色的名稱、他新增的道具名稱，以及該道具的類型，且同樣名稱的道具保證會有相同的類型。例如

  ```
  A Alice potion 1
  A Alice potion 1
  A Alice sword 2
  ```

[2]舉例來說，一個 `int capacity` 表示背包容量好像不錯，一個 `bool isEmpty()` 好像也不錯。

表示 Alice 依序撿到了類型 1 的藥水、又一個藥水，以及類型 2 的劍，並且放進背包[3]。若該角色的背包已達上限，則不要新增該道具，此時請輸出

```
no more capacity
```

若該角色不存在，例如

```
A Alison potion 1
```

請輸出

```
no such a character
```

- 事件代表字元為 H 時需在該角色的背包中尋找是否有給定的道具，後面將依序出現一個角色名稱以及一個道具名稱，若有請輸出 yes，沒有則輸出 no。例如若輸入為

```
H Alice potion
```

則輸出為

```
yes
```

若輸入為

```
H Alice herb
```

則輸出為

```
no
```

若該角色不存在，請輸出

```
no such a character
```

- 事件代表字元為 U 時表示某角色使用了某個道具，後面將依序出現一個角色名稱以及一個道具名稱。若該角色擁有該道具，且可以使用該道具，請將該道具減一，不印出任何東西。例如

```
U Alice potion
```

則 Alice 的 potion 擁有數應減一。若該角色不擁有該道具，請輸出

```
does the character own it?
```

若該角色不能使用該道具，請輸出

```
cannot use it
```

若該角色不存在，請輸出

```
no such a character
```

[3]請不要在意為什麼藥水跟劍都是佔去一單位的空間；遊戲中都是這樣的啦。

- 事件代表字元為 L 時，請將某角色的背包中的所有道具依前述 `getItemList()` 的規則列印出來。例如

```
L Alice
```

則輸出

```
potion,sword
```

若該角色不存在，請輸出

```
no such a character
```

- 事件代表字元為 S 時，請依前述 `getItemSummary()` 的規則計算並印出某角色的背包中的所有道具的個數，例如

```
S Alice
```

則輸出

```
potion,1;sword,1
```

若該角色不存在，請輸出

```
no such a character
```

- 事件代表字元為 V 時表示某角色嘗試丟棄某個道具，後面將依序出現角色以及要丟棄的道具名稱，例如

```
V Alice potion
```

若該角色不擁有該道具，請輸出

```
does the character own it?
```

若該角色不存在，請輸出

```
no such a character
```

第一題

（40 分）在這一題中，請用 array-based implementation 的方式，依上述規則實作 `ItemBag`。

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，每一行皆代表一個事件，而每一個事件的格式則如前述。事件共有最少一筆、最多 10000 筆。此外，角色的總數為 1 到 10 的整數；角色的等級為 1 到 50 的整數；背包的容量為 1 到 10 的整數；角色的名字長度介於 1 到 20 個字元；道具的名字長度介於 1 到 20 個字元，且角色及道具的名字都只包含英文字母，且有區分大小寫，不會有空白鍵、標點符號、數字等等。

在依序讀入這些資訊後，請針對每一個事件做出相應的動作、更新系統狀態、印出指定的資訊。舉例來說，如果輸入是

```
R Alice 10 4
D Bob 8 3
R Alice 7 2
A Alice potion 1
A Alice herb 3
A Alice sword 3
A Alice potion 1
A Alice sword 3
A Alison potion 1
H Alice herb
H Alison potion
U Alice potion
U Alice cloak
U Alice herb
L Alice
S Alice
V Alice potion
L Alice
```

則輸出應該是

```
the character exists
no more capacity
no such a character
yes
no such a character
does the character own it?
cannot use it
herb,potion,sword
herb,1;potion,1;sword,1
herb,sword
```

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***使用上課沒教過的方法，也***不可以***使用 C++ 的 `vector` 或 standard library（STL）。

評分原則

這一題的其中 40 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。

另外，助教也會檢查你的程式，確認你是否有使用 array-based implementation、是否有實作上述提到必須包含的函式、是否確實沒有使用規定不能用的功能。如果有違反規定之處，將會酌予扣分，其中若不使用 array-based implementation 將整題 0 分（不論 PDOGS 上得到幾分）。

第二題

（60 分）在這一題中，請用 link-based implementation 的方式，依上述規則實作 `ItemBag`。

輸入輸出格式

同第一題。

你上傳的原始碼裡應該包含什麼

同第一題。

評分原則

- 這一題的其中 40 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。
- 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運算邏輯、可讀性，以及可擴充性（順便檢查你有沒有使用上課沒教過的語法，並且抓抓抄襲）。請寫一個「好」的程式吧！

另外，助教也會檢查你的程式，確認你是否有使用 array-based implementation、是否有實作上述提到必須包含的函式等。如果有違反規定之處，將會酌予扣分，其中若不使用 link-based implementation 將整題 0 分（不論 PDOGS 上得到幾分）。