**物件導向程式設計作業**

|  |
| --- |
| **命題者︰Hong-Yang Shih** |
| **題目名稱(中文/英文)：Student** **Flea market** |
| **主要測試觀念：**   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  □ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  ■ ARRAYS  ■ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES  □ STRINGS  ■ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **題目說明：**  Design a class **Student**, which can trade items with shop and other students.  There are **N** students and **M** items can be traded. Each student has their own money, and each item has its own value. Shop has unlimited items. Students only have money and no item at the beginning. The item’s value will decrease $10 in each transaction, and the minimum of item’s value is $0.  **輸入說明：**  The first line contains two integers ***N***and ***M***, which are the number of students and items, respectively.  The second line contains ***N*** integers for every students’ money.  The third line contains ***M*** integers for every item’s value.  The following lines are input commands.  **Input command:**   1. buy <student\_id> <item\_id> Student buy an item from the shop. For example, if input is **buy 0 1**, then student 0 will buy an item 1 from the shop. 2. sell < student\_id> <item\_id> <item\_value> Student sells an item to the shop, and the item value must be <item\_value>.  For example, if input is **sell 0 1 100**, then student 0 will sell an item 1 which value is $100 to the shop. 3. trade <student1\_id> < student2\_id> <item\_id> <item\_value\_min> <item\_value\_max> Student1 trade an item with Student2, and the item value range is between <item\_value\_min> and <item\_value\_max>. Assume there is an item id N. If some student have more than one item N which has the value between the range, **trade with the item N having the highest value.** For example, if input is **trade 0 1 1 100 200**, then student 0 will **buy** an item 1 which value is between $100 and $200 from student 1. 4. show <student\_id> Print money and items owned by the student. Items need to be sorted by id and its own value. 5. end Close the program.   **Exception case**  If student doesn’t have enough money, print “Student <student\_id> doesn't have enough money.”  If there is no matching item, print “Student <student\_id> doesn’t have this item.”  **輸出說明：**  See the Sample Output.  **IO範例 :**   |  |  |  | | --- | --- | --- | |  | **Sample Input** | **Sample Output** | | 第一組 | 2 5  1000 2000  50 100 150 200 250  buy 0 1  show 0  sell 0 0 50  buy 1 4  show 1  trade 0 1 4 100 250  show 0  show 1  trade 0 1 4 100 250  buy 0 4  buy 0 4  buy 0 4  show 0  end | Student 0 has $900.  ID 1 $90  Student 0 doesn't have this item.  Student 1 has $1750.  ID 4 $240  Student 0 has $660.  ID 1 $90  ID 4 $230  Student 1 has $1990.  Student 1 doesn't have this item.  Student 0 doesn't have enough money.  Student 0 has $160.  ID 1 $90  ID 4 $230  ID 4 $240  ID 4 $240 | | 第二組 | 3 5  1000 2000 3000  50 100 150 200 250  buy 0 0  buy 0 1  buy 0 2  buy 0 3  buy 0 4  show 0  trade 2 1 0 100 250  trade 2 0 0 100 250  trade 2 0 0 40 50  show 0  show 2  trade 1 2 0 20 30  show 1  show 2  sell 0 0 50  sell 0 0 20  sell 1 0 20  show 1  end | Student 0 has $250.  ID 0 $40  ID 1 $90  ID 2 $140  ID 3 $190  ID 4 $240  Student 1 doesn't have this item.  Student 0 doesn't have this item.  Student 0 has $290.  ID 1 $90  ID 2 $140  ID 3 $190  ID 4 $240  Student 2 has $2960.  ID 0 $30  Student 1 has $1970.  ID 0 $20  Student 2 has $2990.  Student 0 doesn't have this item.  Student 0 doesn't have this item.  Student 1 has $1990. | |
| **附屬資料︰**  🗹解答程式：Item.h, Item.cpp, Student.h, Student.cpp, Trade.cpp  🗹測試資料：input1.txt, output1.txt, input2.txt, output2.txt, input3.txt, output3.txt |
| * + 易，僅需用到基礎程式設計語法與結構   ■ 中，需用到多項程式設計語法與結構   * + 難，需用到多項程式結構或較為複雜之資料型態或結構 |
| **解題時間：60**分鐘。 |
| **其他註記：** |