Programming Design, Spring 2015
Homework 12A

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

To submit your work, please upload two CPP files for Problems 1 and 2 (optional) to PDOGS at http://pdogs.ntu.im/judge/. Each student must submit her/his individual work. No hard copy. No late submission. The due time of this homework is ***8:00am, June 8, 2014***. Please answer in either English or Chinese.

This is Homework 12A; Homework 12B will be given in the next week. They together count for 100 basic points and 40 bonus points. Please note that they have separated submission deadlines.

Before you start, please read Chapters 12 of the textbook.[1] The TA who will prepare the solution for this homework is ***Willy Liao***.

Problem 1

(50 points) In this problem, let's practice on inheritance by somewhat repeating what we did in Problem 2 of Homework 11. Basically, you will still be given past transactions, among which you should find the list of items that are purchased for the most times: all male consumers, all female consumers, all members, all non-members, and all consumers. Below we indicate the differences between Problem 2 of Homework 11 and this problem.

Input/output formats

In each transaction, there are fields for a date, a member ID, a gender, an age, and a list of items purchased. The formats and availability of these fields are the same as in Problem 2 of Homework 11.

You have upgraded your POS system twice. Therefore, the system records the set of items purchased in three different ways. While in the project you will work with all three formats and in Homework 11 you only work with the first one, in this problem we work on the first two. The first format has been described in Problem 2 of Homework 11. In the second format, the date, member ID, gender, and age are recorded in the same way. However, now only the IDs of purchased items are recorded. Below are some examples (corresponding to those provided in Problem 2 of Homework 11):

```
20130516,abcd1234,M,,1,4,6,7
20130516,,F,20,3,4,9,10
20130624,,,,5,6,7,10
```

After the field of age, the list of purchased items is given. Two consecutive values are separated with a comma. These transactions are provided to you in 24 text files, one for each month. The transaction format is consistent within a file. In each file, the number of transactions is no greater than 1000. Though some values may be missing as described above, you may assume that all the information are recorded correctly.

In each of the 15 input files that your program will read, there are $k+1$ lines. The first line contains a positive integer k that is no greater than 24. Each of the following line then contains an integer $t \in \{1, 2\}$ and then a file name (including ".txt" as the extension file name) as a string with no white spaces. The integer t tells you whether that file is in format 1 or 2; the file contains the sales data in one month. For example, your input file may look like

```
2
```

[1] The textbook is *C++ How to Program: Late Objects Version* by Deitel and Deitel, seventh edition.

```
1 D:/file1.txt
2 D:/file8.txt
```

The task of your program, as well as all other details not mentioned here, are the same as those in Problem 2 of Homework 11.

What should be in your source file

Let `const int MAX_ITEM_COUNT = 10;` declares a global variable. You are required to implement three classes with inheritance. The first one

```
class AbstractTSN
{
protected:
  string memberID;
  char gender;
  int age;
  int itemCount;
public:
  // some functions
};
```

is the base class of two derived classes

```
class FirstTSN : public AbstractTSN
{
private:
  bool items[MAX_ITEM_COUNT];
public:
  // some functions
};
```

and

```
class SecondTSN : public AbstractTSN
{
private:
  int* items;
public:
  // some functions
};
```

The class `AbstractTSN` defines an abstract transaction with the information of purchased items missing. The two classes `FirstTSN` and `SecondTSN` provides two different ways to store purchased items. If you see a transaction recorded in format 1, you should create a `FirstTSN` object to store this transaction; for one in format 2, use `SecondTSN`.[2] These objects should be stored somehow in your program. One natural option, of course, is two arrays of the two classes. You may even consider using just static arrays, as you know the maximum number of transactions in each file.

Your .cpp source file should contain C++ codes that will both read testing data and complete the above task. For this problem, you are NOT allowed to use techniques not covered in lectures. You should write relevant comments for your codes.

[2] It is true that the two formats need not to be bound with the two classes. We adopt this setting for pedagogic reasons.

Grading criteria

If you do not use the three classes to process the data, you get 0 points. If you do, 30 points for this program will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. For each set of input data, if your program outputs correctly without violating the space limit, you get 2 points.

20 points for this program will be based on how you write your program, including the logic and format. Please try to write a robust, efficient, and easy-to-read program.

Problem 2 (bonus)

(20 points) Continue from Problem 1. Redo the task given in Problem 3 of Homework 11. There will be 10 input files, one for 2 points. For this problem, you may use any technique you like.