Assignment: Create three sets from lists of integers found in files **i071.dat**, **i072.dat**, and **i073.dat**. Find set I the intersection of set 1 and set 2. Find set U the union of set 2 and set 3. Find set D the difference of set U - set I. Print results in file **o07.dat** as shown in Figure 2.

Prohibition: Use of the C++ Standard Template Library is prohibited in the implementation of this project.

Program Files: Project **7** consists of files **p07.cpp**, **List07.h**, **Set07.h**, and **p07make**. Project file names are exactly as given. Failure to employ the foregoing names will result in a score of ***zero*** (***0***) for this project

Project files must be stored in the **root directory of your student account**. Failure to store project files in the root directory of your student account will result in a score of ***zero*** (***0***) for this project.

File	Description
p07.cpp	File **p07.cpp** contains functions that process command line arguments and direct set operations. Implement the finding of the union, intersection, and difference as arithmetic expression rather than calling member functions.
List07.h	File **List07.h** contains all that is required for **class** *List* **{ … };** Class *List* implements a list of integers by dynamically allocating an integer array. Elements are stored in ascending order. Elements of the list are unique. Class *List* is a concrete class and not a template. Overload operator<< to print elements of class List enclosed in curly braces, { }, and separated by commas. Make operator << a friend of class List. Overload operator= and make it the copy constructor for class List. Class *List* is the base class.
Set07.h	File **Set07.h** contains all that is required for **class** *Set* **: public** *List* **{ … };**. Class *set* is derived from class *List*. Set operations include insertion, deletion, union, intersection, and difference, and printing. Class set is a concrete class and is not a template. Overload operator+, operator*, and operator- to be the union, intersection, and difference for class Set: these operators need to friends of the class.
p07make	File **p07make** contains instructions for creating executable file **p07**. Instructions in file **p07make** are accepted by the UNIX utility make.

Command Line: Project **7** can be invoked with up to four program parameters. The first three program parameters are the names of the input files containing sets 1, 2 and 3. The fourth program parameter in the name of the output file where results are recorded. Sample command lines together with corresponding actions by program **p07** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

$ **p07**
Enter the name of input file 1: **i071.dat**
Enter the name of input file 2: **i072.dat**
Enter the name of input file 3: **i073.dat**
Enter the output file name: **o07.dat**

$ **p07 i071.dat**
Enter the name of input file 2: **i072.dat**
Enter the name of input file 3: **i073.dat**
Enter the output file name: **o07.dat**

$ **p07 i071.dat i072.dat**
Enter the name of input file 3: **i073.dat**
Enter the output file name: **o07.dat**

$ **p07 i071.dat i072.dat i073.dat**
Enter the output file name: **o07.dat**

$ **p07 i071.dat i072.dat i073.dat o07.dat**

Input files: Input files contain a list of integers. Unique integers in the list are the elements of a set. Sample data is given in below.

Input file 1: 1 2 3 4

Input file 2: 2 4 6 8

Input file 3: 1 3 5 7

Output files: The format of the output file is shown in Figure 2. Data shown in Figure 2 is produced by Program **p07** given input files shown above.

Figure 2.
Output file format:

set 1={1,2,3,4}
set 2={2,4,6,8}
set 3={1,3,5,7}
set I={2,4}
set U={1,2,3,4,5,6,7,8}
set D={1,3,5,6,7,8}

```
void SetMgr(istream& i1,istream& i2,istream& i3,ostream& o)
{
   Set S1(i1);
   o    << endl << "Set S1=" << S1;
   cout << endl << "Set S1=" << S1;

   Set S2(i2);
   o    << endl << "Set S2=" << S2;
   cout << endl << "Set S2=" << S2;

   Set S3(i3);
   o    << endl << "Set S3=" << S3;
   cout << endl << "Set S3=" << S3;

   Set U;
   U=S1+S2;
   o    << endl << "Set U=" << U;
   cout << endl << "Set U=" << U;

   Set I;
   I=S2*S3;
   o    << endl << "Set I=" << I;
   cout << endl << "Set I=" << I;

   Set D;
   D=U-I;
   o    << endl << "Set D=" << D;
   cout << endl << "Set D=" << D;

   cout << endl;
   o    << endl;

}
```

Figure 3. Function SetMgr(…) in file p07.cpp