CS246 Winter 2013 Assignment 4   V1.0

Sets of integers as a C++ class. In this assignment you are to write a C++ class named *SetInt* that supports various operations on sets of integers such as + (union), - (set subtraction) and * (intersection). An object of this class represents a set of integers.

You are to represent each object as a C++ vector of integers. The size of the vector gives the cardinality (number of integers) in the set.

The interface to the StrInt class will be given to you as the file setint.h and its implementation must be in the file setint.cpp. You are not to change the setint.h file.

Implementation of a driver for using sets of integers.
Your main program, to be named runset.cpp, is used to manipulate sets. It reads commands from the keyboard and produces output on the terminal. The executable version of your program is to be named *driver*.

The main program maintains a C++ vector of classes, whose two parts are the set name and the set value. The set value is an instance of the SetInt class.

Your driver does not require DBU (declaration before use). Instead, any time a named set is used, it automatically declared (if not already declared) – much like how shell language implicitly declares shell variables. The value of an automatically declared variable is the empty set.

You program should be kept simple by doing only simple checks for the correctness of its input. All error messages should be sent to the standard error stream and should begin with the word ERROR. Generally, your program should keep on executing after outputting an error message.

Strings read by the driver are non-whitespace surrounded by white space. These can be read using cin and >>.

The help command
The driver outputs help information when you give it the '?' command. There is a function in help.cpp that prints out the list of commands. You should study those files as they list commands that the driver is to carry out.   Your main program should #include help.h. You are not to change help.h or help.cpp.

Example driver session. Below is an example interaction between a user and the driver. The parts in **bold** are typed by the user. The other parts are prompts written by the program. Here is a session with the driver, run by the command ./driver

```
Driver for set operations. Type '?' for help.
Command: e
Hello_world!!
Hello_world!!
```

```
Command: i
Init: x = { int1 int2 ... intN }: myvar = { 3 5 2 }
Command: p
Print. Give set name: myvar
2
3
5
Command: =
Assign: x = y: abc = myvar
Command: p
Print. Give set name: abc
2
3
5
Command: i
Init: x = { int1 int2 ... intN }: def = { 1 2 4 }
Command: +
Union. var1 = var2 + var3: u = abc + myvar
Command: p
Print. Give set name: u
2
3
5
Command: +
Union. var1 = var2 + var3: u = myvar + def
Command: p
Print. Give set name: u
1
2
3
4
5
Command: q
```

Sorted sets. When a set is output, the members should be output in numeric order. In your implementation, keep all sets in numeric order. The user does not need to type in elements in order.

Related files. You will be given a compiled (master) solution to this assignment. You can run this under your student account to see if your solution gives the same answers. Your output (except when there are user errors) from your program should be the same, character for character, as the master solution.

You will be given the makefile for your program. You should not change it. It should compile your program.

You will be given the files: setint.h, help.h and help.cpp and you are not to change these files.

These files can be found on the student Linux system in the directory

       ~holt/cp-for-students/cs/246/2013/asgn04

How to Submit.
You will be using Marmoset to submit and test your program. Tests available on Marmoset will account for your correctness marks for the assignment.
Please submit file 'myset.zip' to project 'a4' on Marmoset. Your zipped file should contain the following files:
makefile   (unchanged as given to you)
help.cpp   (unchanged as given to you)
help.h   (unchanged as given to you)
driver.cpp
setint.cpp
setint.h (unchanged as given to you)

Marking scheme.
40 Correctness Marks
10 Style Marks