Source File: `~/2336/11/lab11.(C|CPP|cpp|c++|cc|cxx|cp)`
Input: Under control of `main` function
Output: Under control of `main` function
Value: 1

The purpose of this assignment is to become more familiar with the process of providing overloaded operators for a class. The `IntegerSet` class from Labs 04, 06, 08, and 10 will be modified to provide:

- overloaded operators to implement equality, subset, and proper subset.

Some useful definitions:

- Two sets A and B are said to be equal if every element of A is an element of B and every element of B is an element of A.
- Set A is said to be a subset of set B if and only if each element of A is an element of B.
- Set A is said to be a proper subset of set B if A is a subset of B and at least one member of B is not a member of A.

A header file is shown in Figure 1, a sample `main` function for testing your implementation is shown in Figure 2, and a sample execution sequence is shown in Figure 3. To use the `Makefile` as distributed in class, add a target of `lab11` to `targets2srcfileswithlibrary`.

```
1  #ifndef LAB11_H
2  #define LAB11_H
3
4  #include <iostream>
5  #include <bits.h>
6
7  using namespace std;
8
9  const uint N = 40;
10
11 class IntegerSet
12 {
13   // overloaded output operator for printing IntegerSet set to
14   // output stream out
15   friend ostream& operator<<(ostream& out, const IntegerSet& set);
16  public:
17   IntegerSet();                            // initializes the set to the empty
18                                            //   set
19   IntegerSet(const IntegerSet& otherSet); // copy constructor
20   ~IntegerSet();                           // destructor
21   bool isMember(uint e) const;             // returns true if e is a member of
22                                            //   the set and false otherwise
23   uint cardinality() const;                // cardinality of a set
24   IntegerSet operator+(uint e) const;      // if e is valid and not a member of
25                                            //   the set, insert e into set
26   IntegerSet operator-(uint e) const;      // if e is valid and a member of
27                                            //   the set, delete e from set
28   IntegerSet operator-() const;            // complement of a Set
29   IntegerSet& operator=(const IntegerSet& rhs); // *this = rhs
30
```

Figure 1. `/usr/local/2336/include/lab11.h` (Part 1 of 2)

```
31    IntegerSet operator+(const IntegerSet& rhs) const; // union
32    IntegerSet operator*(const IntegerSet& rhs) const; // intersection
33    IntegerSet operator-(const IntegerSet& rhs) const; // difference
34    IntegerSet operator/(const IntegerSet& rhs) const; // symmetric diff
35
36    bool operator==(const IntegerSet& rhs) const; // Test for equality
37    bool operator<=(const IntegerSet& rhs) const; // Subset
38    bool operator< (const IntegerSet& rhs) const; // Proper Subset
39
40   private:
41    uint *bitVector;                          // Pointer to dynamically
42                                              //   allocated memory
43    bool isValid(uint e) const;               // 0 <= e < N
44    uint word(uint n) const;                  // Determine index within
45                                              //   bitVector where n is located
46    uint bit(uint n) const;                   // Determine position within
47                                              //   bitVector[word(n)]
48                                              //   for element n
49    void allocateStorage();                   // Calculate # of elements
50                                              //   in bitVector to represent
51                                              //   elements 0..(N-1) & then
52                                              //   allocate storage
53  };
54
55  #endif
```

Figure 1. `/usr/local/2336/include/lab11.h` (Part 2 of 2)

```
#include <lab11.h>
#include <iomanip>

using namespace std;

int main()
{
  uint e, j, n;
  IntegerSet s, t, c, u, i, d;

  while (cin >> n)
  {
    for (j = 0; j < n; ++j)
    {
      cin >> e;
      s = s + e;
    }
    cout << " s = " << s;
    cout << "s.cardinality() = " << s.cardinality() << endl;

    cin >> n;
    for (j = 0; j < n; ++j)
    {
      cin >> e;
      t = t + e;
    }
    cout << " t = " << t;
    cout << "t.cardinality() = " << t.cardinality() << endl;

    cout << boolalpha << "s == t = " << (s == t) << endl;
    cout << "s <= t = " << (s <= t) << endl;
    cout << "s <  t = " << (s <  t) << endl;

    // clear sets s & t
    for (e = 0; e < N; ++e)
    {
      if (s.isMember(e))
        s = s - e;
      if (t.isMember(e))
        t = t - e;
    }
  }

  return 0;
}
```

Figure 2. `/usr/local/2336/src/lab11main.C`

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 11
  * Checking to see if a folder exists for Lab 11. . .No
  * Creating a folder for Lab 11
  * Checking to see if Lab 11 has sample input and output files. . .Yes
  * Copying input and output files for Lab 11
    from folder /usr/local/2336/data/11 to folder ./11
  * Checking to see if /usr/local/2336/src/lab11main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab11main.C to folder ./11
  * Checking to see if /usr/local/2336/include/lab11.h exists. . .Yes
  * Copying file /usr/local/2336/include/lab11.h to folder ./11
  * Copying file /usr/local/2336/src/Makefile to folder ./11
  * Adding a target of lab11 to targets2srcfileswithlibrary
  * Touching file ./11/lab11.cpp
  * Edit file ./11/lab11.cpp in Notepad++
newuser@csunix ~/2336> cd 11
newuser@csunix ~/2336/11> ls
01.dat       01.out       Makefile     lab11.cpp    lab11.h      lab11main.C
newuser@csunix ~/2336/11> make lab11
g++ -g -Wall -std=c++11 -c lab11main.C -I/usr/local/2336/include -I.
g++ -g -Wall -std=c++11 -c lab11.cpp -I/usr/local/2336/include -I.
g++ -o lab11 lab11main.o lab11.o -L/usr/local/2336/lib \
-Wl,-whole-archive -llab11 -Wl,-no-whole-archive -lm -lbits
newuser@csunix ~/2336/11> cat 01.dat
4
1 2 3 4
3
1 4 5
6
1 2 4 8 16 32
10
3 6 9 12 15 3 6 9 12 15
13
4 8 12 16 20 24 28 32 36 40 44 48 52
48
 0  1  2  3  4  5  6  7  8  9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47
0
0
```

Figure 3. Commands to Compile, Link, & Run Lab 11 (Part 1 of 2)

```
43 newuser@csunix ~/2336/11> cat 01.dat | ./lab11
44  s = {1,2,3,4}
45 s.cardinality() = 4
46  t = {1,4,5}
47 t.cardinality() = 3
48 s == t = false
49 s <= t = false
50 s <  t = false
51  s = {1,2,4,8,16,32}
52 s.cardinality() = 6
53  t = {3,6,9,12,15}
54 t.cardinality() = 5
55 s == t = false
56 s <= t = false
57 s <  t = false
58  s = {4,8,12,16,20,24,28,32,36}
59 s.cardinality() = 9
60  t = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39}
61 t.cardinality() = 40
62 s == t = false
63 s <= t = true
64 s <  t = true
65  s = ∅
66 s.cardinality() = 0
67  t = ∅
68 t.cardinality() = 0
69 s == t = true
70 s <= t = true
71 s <  t = false
72 newuser@csunix ~/2336/11> cat 01.dat | ./lab11 > my.out
73 newuser@csunix ~/2336/11> diff 01.out my.out
74 newuser@csunix ~/2336/11>
```

Figure 3. Commands to Compile, Link, & Run Lab 11 (Part 2 of 2)