Source File: `~/2336/06/lab06.(C|CPP|cpp|c++|cc|cxx|cp)`
Input: Under control of `main` function
Output: Under control of `main` function
Value: 2

Extend the `IntegerSet` class from Lab 04 to provide the following additional member functions:

- The union of sets A and B is the set of all elements that are in A or in B or in both A and B.
- The intersection of sets A and B is the set of all elements that are in both A and B.
- The difference of sets A and B is the set of all elements in A that are not in B.
- The symmetric difference of two sets is the union of two sets minus the intersection.

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 `lab06` to `targets2srcfileswithlibrary`.

```
1  #ifndef LAB06_H
2  #define LAB06_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  public:
14   IntegerSet();                            // initializes the set to the empty
15                                            //   set
16   IntegerSet(const IntegerSet& otherSet); // copy constructor
17   ~IntegerSet();                           // destructor
18   bool isMember(uint e) const;             // returns true if e is a member of
19                                            //   the set and false otherwise
20   uint cardinality() const;                // cardinality of a set
21   void insertElement(uint e);              // if e is valid and not a member of
22                                            //   the set, insert e into set
23   void deleteElement(uint e);              // if e is valid and a member of
24                                            //   the set, delete e from set
25   IntegerSet complement() const;           // complement of a Set
26
27   ostream& print(ostream& os) const;
28   IntegerSet Union(const IntegerSet& otherSet) const;
29   IntegerSet intersection(const IntegerSet& otherSet) const;
30   IntegerSet difference(const IntegerSet& otherSet) const;
31   IntegerSet symmetricDifference(const IntegerSet& otherSet) const;
```

Figure 1. `/usr/local/2336/include/lab06.h` (Part 1 of 2)

```
32  private:
33   uint *bitVector;                        // Pointer to dynamically
34                                           //   allocated memory
35   bool isValid(uint e) const;             // 0 <= e < N
36   uint word(uint n) const;                // Determine index within
37                                           //   bitVector where n is located
38   uint bit(uint n) const;                 // Determine position within
39                                           //   bitVector[word(n)]
40                                           //   for element n
41   void allocateStorage();                 // Calculate # of elements
42                                           //   in bitVector to represent
43                                           //   elements 0..(N-1) & then
44                                           //   allocate storage
45 };
46
47 #endif
```

Figure 1. `/usr/local/2336/include/lab06.h` (Part 2 of 2)

```
1  #include <lab06.h>
2  #include <iomanip>
3
4  using namespace std;
5
6  int main()
7  {
8    uint e, j, n;
9    IntegerSet s, t;
10
11   while (cin >> n)
12   {
13     for (j = 0; j < n; ++j)
14     {
15       cin >> e;
16       s.insertElement(e);
17     }
18     cout << " s = ";
19     s.print(cout);
20     cout << "s.cardinality() = " << s.cardinality() << endl;
21
22     cin >> n;
23     for (j = 0; j < n; ++j)
24     {
25       cin >> e;
26       t.insertElement(e);
27     }
```

Figure 2. `/usr/local/2336/src/lab06main.C` (Part 1 of 2)

```
28      cout << " t = ";
29      t.print(cout);
30      cout << "t.cardinality() = " << t.cardinality() << endl;
31
32      // Use the copy constructor to initialize u with the union of s & t
33      IntegerSet u(s.Union(t));
34      cout << " u = ";
35      u.print(cout);
36
37      // Use the copy constructor to initialize i with the intersection of s & t
38      IntegerSet i(s.intersection(t));
39      cout << " i = ";
40      i.print(cout);
41
42      // Use the copy constructor to initialize d with the difference of s & t
43      IntegerSet d(s.difference(t));
44      cout << " d = ";
45      d.print(cout);
46
47      // Use the copy constructor to initialize sd with the symmetric
48      // difference of s & t
49      IntegerSet sd(s.symmetricDifference(t));
50      cout << "sd = ";
51      sd.print(cout);
52      cout << endl;
53
54      // clear sets s & t
55      for (e = 0; e < N; ++e)
56      {
57        if (s.isMember(e))
58          s.deleteElement(e);
59        if (t.isMember(e))
60          t.deleteElement(e);
61      }
62    }
63
64    return 0;
65  }
```

Figure 2. `/usr/local/2336/src/lab06main.C` (Part 2 of 2)

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 06
  * Checking to see if a folder exists for Lab 06. . .No
  * Creating a folder for Lab 06
  * Checking to see if Lab 06 has sample input and output files. . .Yes
  * Copying input and output files for Lab 06
    from folder /usr/local/2336/data/06 to folder ./06
  * Checking to see if /usr/local/2336/src/lab06main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab06main.C to folder ./06
  * Checking to see if /usr/local/2336/include/lab06.h exists. . .Yes
  * Copying file /usr/local/2336/include/lab06.h to folder ./06
  * Copying file /usr/local/2336/src/Makefile to folder ./06
  * Adding a target of lab06 to targets2srcfileswithlibrary
  * Touching file ./06/lab06.cpp
  * Edit file ./06/lab06.cpp in Notepad++
newuser@csunix ~/2336> cd 06
newuser@csunix ~/2336/06> ls
01.dat       01.out       Makefile     lab06.cpp    lab06.h      lab06main.C
newuser@csunix ~/2336/06> make lab06
g++ -g -Wall -std=c++11 -c lab06main.C -I/usr/local/2336/include -I.
g++ -g -Wall -std=c++11 -c lab06.cpp -I/usr/local/2336/include -I.
g++ -o lab06 lab06main.o lab06.o -L/usr/local/2336/lib \
-Wl,-whole-archive -llab06 -Wl,-no-whole-archive -lm -lbits
newuser@csunix ~/2336/06> 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 06 (Part 1 of 2)

```
newuser@csunix ~/2336/06> cat 01.dat | ./lab06
 s = {1,2,3,4}
s.cardinality() = 4
 t = {1,4,5}
t.cardinality() = 3
 u = {1,2,3,4,5}
 i = {1,4}
 d = {2,3}
sd = {2,3,5}

 s = {1,2,4,8,16,32}
s.cardinality() = 6
 t = {3,6,9,12,15}
t.cardinality() = 5
 u = {1,2,3,4,6,8,9,12,15,16,32}
 i = ∅
 d = {1,2,4,8,16,32}
sd = {1,2,3,4,6,8,9,12,15,16,32}

 s = {4,8,12,16,20,24,28,32,36}
s.cardinality() = 9
 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}
t.cardinality() = 40
 u = {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}
 i = {4,8,12,16,20,24,28,32,36}
 d = ∅
sd = {0,1,2,3,5,6,7,9,10,11,13,14,15,17,18,19,21,22,23,25,26,27,29,30,31,33,34,35,37,38,39}

 s = ∅
s.cardinality() = 0
 t = ∅
t.cardinality() = 0
 u = ∅
 i = ∅
 d = ∅
sd = ∅

newuser@csunix ~/2336/06> cat 01.dat | ./lab06 > my.out
newuser@csunix ~/2336/06> diff 01.out my.out
newuser@csunix ~/2336/06>
```

Figure 3. Commands to Compile, Link, & Run Lab 06 (Part 2 of 2)