Lab 2 test plan:

Test no-parameter constructor:

IntSet a; // no-parameter constructor

Test variable-parameter constructor with some parameters not initialized:

IntSet e(0, 1, 2); // variable-parameter constructor

Test variable-parameter constructor with all parameters initialized:

IntSet f(1, 2, 40, 90, MAXSUPPORTEDINT);

Test variable-parameter constructor with invalid values

IntSet g(-1, MAXSUPPORTEDINT + 1);

Test copy constructor

IntSet b(a); // copy constructor

Test overloaded assignment operator=

d = e; // assignment operator

Test overloaded assignment operator=, self-assignment (no-op)

d = d; // self-assignment; operator implementation must handle this

Test overloaded output operator<<

cout << "Union: d " << d << " + g " << g << endl;

Test overloaded input operator>>

cin >> A;

Test overloaded input operator>> with non-integer input

cin >> A;

Test overloaded union operator+

cout << d + g << endl << endl;

Test overloaded intersection operator\*

cout << d \* g << endl << endl;

Test overloaded difference operator-

cout << d - g << endl << endl;

Test overloaded union assignment operator+=

IntSet added = d;

added += g;

Test overloaded union assignment operator+=, self-assignment (no-op)

added += added;

Test overloaded intersection assignment operator\*=

IntSet intersect(2, 5, 6, 7);

intersect \*= e;

Test overloaded intersection assignment operator\*=, self-assignment (no-op)

intersect \*= intersect;

Test overloaded difference assignment operator-=

IntSet difference(2, 5, 6, 7);

difference -= e;

Test overloaded difference assignment operator-=, self-assignment (null set)

difference -= difference;

Test overloaded equality operator==, different objects, different content

cout << (d == g) << endl << endl;

Test overloaded equality operator==, same object

cout << (d == d) << endl << endl;

Test overloaded equality operator==, different objects, same content

IntSet copyOfD = d;

cout << (copyOfD == d) << endl << endl;

Test overloaded inequality operator!=, different objects, different content

cout << (d != g) << endl << endl;

Test insert() method

int insert = 30;

d.insert(insert);

Test remove() method

d.remove(insert);

Test isInSet() method

cout << d.isInSet(insert) << endl << endl;;

Test isEmpty() method

cout << d.isEmpty() << endl << endl;