* Notable Obstacles:
  + In the class Cell, for the function isValueOK( int thisvalue, NumberSet box, NumberSet row, NumberSet column ), I had forgotten to scope the box, column and row to NumberSet::notFound, which caused a few errors.
  + For Cell::isImmutable(), I didn’t return mImmutable and returned the values true or false which caused a few problems.
  + I did not know how to loop through a multidimensional array.
  + For Board::completed(), I tried to use the value mCells without using ‘.getValue()’. Also, I forgot to add a case in which the result is false and break out of the loop which caused many errors.
  + For Board::evaluate(), my condition for when the board was complete was in the for loop, which led to many errors.
* Test data:

|  |  |  |
| --- | --- | --- |
| **DATA** | **FUNCTION BEING TESTED** | **RESULT** |
| NumberSet ns(1,2,3,4,5,6,7,8);  ns.notFound(0); | NumberSet::notFound(int value) const | Evaluates to true as 0 is not in the provided number set. |
| ns.notFound(1); | NumberSet::notFound(int value) const | Returns false as 1 is in the provided number set. |
| ns.notFound(2); | NumberSet::notFound(int value) const | Returns false as 2 is in the provided number set. |
| ns.notFound(3); | NumberSet::notFound(int value) const | Returns false as 3 is in the provided number set. |
| ns.notFound(4); | NumberSet::notFound(int value) const | Returns false as 4 is in the provided number set. |
| ns.notFound(5); | NumberSet::notFound(int value) const | Returns false as 5 is in the provided number set. |
| ns.notFound(6); | NumberSet::notFound(int value) const | Returns false as 6 is in the provided number set. |
| ns.notFound(7); | NumberSet::notFound(int value) const | Returns false as 7 is in the provided number set. |
| ns.notFound(8); | NumberSet::notFound(int value) const | Returns false as 8 is in the provided number set. |
| Cell c;  c.setValue(1);  c.getValue(); | Cell::setValue(int value);  Cell::getValue() const; | Checks if the value was set to 1 and if the getValue() function is returning 1. |
| c.setImmutable(true);  c.isImmutable(); | Cell::setImmutable(bool i);  Cell:isImmutable() const; | Returns true as mImmutable is set to true |
| c.isValueOK(0, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns true as 0 is not found in the number sets inputted as box, row & column so it is legal |
| c.isValueOK(1, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 1 is found in the number sets inputted as box, row & column |
| c.isValueOK(2, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 2 is found in the number sets inputted as box, row & column |
| c.isValueOK(3, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 3 is found in the number sets inputted as box, row & column |
| c.isValueOK(4, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 4 is found in the number sets inputted as box, row & column |
| c.isValueOK(5, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 5 is found in the number sets inputted as box, row & column |
| c.isValueOK(6, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 6 is found in the number sets inputted as box, row & column |
| c.isValueOK(7, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 7 is found in the number sets inputted as box, row & column |
| c.isValueOK(8, ns, ns, ns); | Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 8 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(0, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as although 0 is not found in the number sets inputted as box, row & column, it is not a legal value to input into an immutable cell |
| c.isOKToSetValue(1, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 1 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(2, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 2 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(3, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 3 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(4, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 4 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(5, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 5 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(6, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 6 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(7, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 7 is found in the number sets inputted as box, row & column |
| c.isOKToSetValue(8, ns, ns, ns); | Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns false as 8 is found in the number sets inputted as box, row & column |
| Game g;  g.cheat("123456789|.........|987321456|456987123|.........|312645978|.........|........  .|.........");  g.evaluate(); | Board::evaluate()  Board::completed()  Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns 36 as those are the total number of cells filled with legal values |
| g.completed(); | Board::completed() | Returns false as there are still cells with the value 0 |
| g.acceptValue(0,1,1); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns false as the particular cell is immutable |
| Game g1;  g1.cheat("53..7....|6..195...|.98....6.|8...6...3|4..8.3..1|7...2...6|.6....28.|...419..5|....8..79")  g1.evaluate();  g1.completed(); | Board::evaluate()  Board::completed() | The board has 30 legal cells filled and it is not completed (returns false) |
| g1.acceptValue(4, 1, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(6, 1, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(8, 1, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(9, 1, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 1, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 1, 9); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 2, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 2, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 2, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 2, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(8, 2, 9); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 3, 1); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 3, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 3, 5); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 3, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 3, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 3, 9); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 4, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(9, 4, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 4, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 4, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 4, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 4, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 5, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(6, 5, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 5, 5); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 5, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(9, 5, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 6, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 6, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(9, 6, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 6, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(8, 6, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 6, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(9, 7, 1); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 7, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 7, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 7, 5); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 7, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 7, 9); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 8, 1); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(8, 8, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(7, 8, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(6, 8, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 8, 8); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(3, 9, 1); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(4, 9, 2); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(5, 9, 3); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(2, 9, 4); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(6, 9, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(1, 9, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns true as the particular cell is mutable and the value being inputted is legal |
| g1.acceptValue(6, 9, 6); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns false as the particular cell is immutable |
| g1.acceptValue(1, 9, 7); | Board::acceptValue( int value, int row, int column, bool immutable)  Cell::isOKToSetValue(int thisvalue, NumberSet box, NumberSet row, NumberSet column)  Cell::isImmutable() const | Returns false as the particular cell is immutable |
| g1.evaluate(); | Board::evaluate()  Board::completed()  Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | The board has a legal value in each cell, so the evaluate function returns 100 |
| g1.completed(); | Board::completed()  Cell::isValueOK(int thisvalue, NumberSet box, NumberSet row, NumberSet column) | Returns true as the board has a legal value in each cell |