Andrew Nguyen AXN210010
Project 3 Pseudocode

Andrew Nguyen AXN210010
Project 3 Pseudocode

Test Cases
1. Draw a horizontal line out of bounds
2. Draw a vertical line out of bounds
3. Draw a bold line (#) over a non-bolded line (*)
4. Move the pointer around the grid with the pen up
5. Test the print function after drawing multiple times
6. Try printing the grid after deleting it
7. Move the pointer to the corner of the grid
8. Draw a line across the entire grid
9. Use several commands to create a diagonal line
10. Move the pointer to the center of the grid and draw a single character

```
Creating the struct node with pointers pointing to other nodes
        char value (Would hold '*', '#', or ' ')
        node *N (North/up)
        node *S (South/down)
        node *W (West/left)
        node *E (East/right)
end

node* CreateRow ()
    Create two pointers that point to two dynamically allocated nodes (previous and next)
    Create a head pointer that points to the previous pointer that was just created (the beginning of the row)

    for (int i = 1; i < 50; i++)
              Set next node's left pointer to the previous node

        If the current node is not the last node of the row (i != 50)
                    Set previous node's right pointer to the next node (previous node acts as the current node)

            Set the previous node equal to the next node
                    Make a new dynamically allocated node and set next node equal to that
        end
    end

    Return the head pointer
end

void CreateGrid (node *&head)
   Create node pointers that point to the beginning of two rows
    node *previousRow = CreateRow()
    node *nextRow = CreateRow()

    Set head equal to the previous row (beginning of the grid)

    for (int i = 1; i < 50; i++)
              Make two traversal pointers that begin at each of the 2 rows
        node *prevTraversal = previousRow
        node *nextTraversal = nextRow

        for (int i = 1; i < 50; i++)
                    Link the up and down pointers of the two rows
            nextTraversal->N = prevTraversal
            prevTraversal->S = nextTraversal

            if the current node is not the last node (i != 50)
                            Move the traversal pointers to the right one node
            end
        end

        if (i != 50)
            Set the previous or current row to the next row
            Set next row to a new row created by the CreateRow() function
        end
    end
end
```

```
void PrintGrid (node *head)
	linePtr = node pointer that points to head (the first node of the first row)
	nextPtr = node pointer that points to the row underneath head (first node of the second row)

	Loop until linePtr == nullptr
		Print the value at the current node that linePtr is pointing at
		Set linePtr to the node to the right of the node it is pointing at
	end

	Print a newline after printing a row

	PrintGrid (nextPtr)
end

void DeleteGrid (node *&head)
	linePtr = node pointer that points to head (the first node of the first row)
	nextPtr = node pointer that points to the row underneath head (first node of the second row)

	Loop until linePtr == nullptr
		Create a hold pointer to hold the node to the right of the current node
		Delete the node at the linePtr
		Set linePtr to the hold pointer
	end

	DeleteGrid (nextPtr)
end
```