Andrew Nguyen AXN210010
Project 2 Pseudocode

Test Cases
1. Parse system of equations with the variables out of the usual order of x,y,z
2. Parse system of equations with 2 variables instead of 3
3. Multiply row 3 by a negative integer
4. Swap rows 1 and 3 to see if the function works properly
5. See if the function to check if a matrix is in RREF works
6. Add 2 times row 2 to row 3 to see if the function works properly
7. See if the parsing can handle empty lines
8. See if the parsing can handle multiple of the same characters it checks for ('x', 'y', 'z', '=') on a single line
9. Check if the for loops counts each row correctly
10. Multiply a row by a decimal

```
Read file function (fileName)
	Opens an input file stream using the given file name

	Create an array for the matrix to be stored in (size 16 which is 4x4)
	All values in the array by default is zero

	Create a variable to keep track of the current line (row of the matrix)

	Go through each line of the file containing the equations
		Create two string variables (target string and remaining string) for future use

		Loop through four times
			Find indexes of the first occurrences of 'x', 'y', 'z', and '=' in the remaining string
			Find which index is the smallest number
			Split the line into two strings (target string and remaining string)

			Split the non-alphabetical characters and the alphabetical characters in the target string
			Save the digits to the array based on the characters checked for

			The positions for 'x', 'y', 'z', and '=' is at 1, 2, 3, and 4

			Add (the number from the current row variable minus 1) multiplied by 4
			to the position when saving to the array
		end

		If the file is not at the end of file then
			Add one to the current row of the matrix variable
		end
	end

	Close the file

	Calculate correct size by multiplying where the current row variable left off by 4
	Create a dynamic array using the correct size
	Copy the array created at the beginning of the function to the dynamic array

	Returns the dynamic array containing the matrix
end
```

```
Matrix reduced row echelon form (RREF) check function (&matrixArray, matrixSize)
	Create boolean to see if matrix is in RREF
	Create variable to track current row
	Create variables for x, y, and z

	For loop (i = 1 until i <= matrixSize)
		Check if row i, column i is 1
			This would actually check for (1, 6, 11) all subtracted by 1 which would be (0, 5, 10)
			So it would check if ((i - 1) % 5 == 0) and if the value at that spot in the array is 1
		Else
			Set RREF to false
		end

		If (i % 4 == 0) then
			Switch (current row)
				case 1:
					Set x to current value in array
				case 2:
					Set y to current value in array
				case 3:
					Set z to current value in array

			Add 1 to current row
		Else
			If everything else is not 0 then
				Set RREF to false
			end
		end
	end

	If RREF is true then
		Print out x, y, and z values
	end

	Returns boolean value of whether or not it is in RREF
end


Display matrix function (&matrixArray, matrixSize)
	For loop (i = 1 until i <= matrixSize)
		Print each value

		If (i % 4 == 0) then
			Print new line
		end
	end

	Returns nothing
end
```

```
Switch rows function (matrixArray, matrixSize)
	Ask the user which 2 rows they want to switch
	(First row going to be switched and second row going to be switched)

	Create a new dynamic array which will be the result of the operation
	Create a variable for the current row

	For loop (i = 1 until i <= matrixSize)
		If (i % 4 == 0) then
			Add one to the current row variable
		end

		Row offset = (Current row - 1) * 4
		Get the difference of rows between the rows to be switched
		(For example, if rows 1 and 3 are to be switched, the difference is 2)

		Use the row offset multiplied by the difference calculated earlier
		to determine the position of the new row

		If the current row is equal to the first row going to be switched then
			Copy current row to the second row going to be switched in the new array
		end
		If the current row is equal to the second row going to be switched then
			Copy current row to the first row going to be switched in the new array
		end

		If the current row is not equal to the 2 rows going to be switched then
			Copy the row(s) that won't be switched to the new array
		end
	end

	Returns the new array
end


Multiply rows function (matrixArray, matrixSize)
	Ask the user for which row they want to multiplying
	Ask the user for a number to multiply

	Create a new dynamic array
	Create a variable for the current row

	For loop (i = 1 until i <= matrixSize)
		If (i % 4 == 0) then
			Add one to the current row variable
		end

		If the current row is equal to the row going to be multiplied then
			Multiply the value from the current matrix by the multiplier and save it to the new array
		Else
			Copy the row(s) that won't be modified to the new array
		end
	end

	Returns the new array
end
```

```
Add scalar of one row to another row function (matrixArray, matrixSize)
	Ask the user which row they want to use for multiplication
	Ask the user for a number to multiply
	Ask the user which row to be modified

	Create a one dimensional array of size 4 to store one of the rows
	Create a variable for the current row

	For loop (i = 1 until i <= matrixSize)
		If (i % 4 == 0) then
			Add one to the current row variable
		end

		If the current row is equal to the row to be used for multiplication then
			Row offset = (Current row - 1) * 4
			Current position = i - Row offset

			Multiply the row by the multiplier and save it to the array of size 4
		end
	end

	Create a new dynamic array
	Set the current row to 1

	For loop (i = 1 until i <= matrixSize)
		If (i % 4 == 0) then
			Add one to the current row variable
		end

		If the current row is equal to the row going to be modified then
			Add the current row by the multiplied row from earlier and save it to the new array
		Else
			Copy the row(s) that won't be modified to the new array
		end
	end


	Returns the new array
end


Main function
	Ask the user for the file name
	Call the read file function to get the matrix as an array

	Create a variable for the current array (matrixArray)
	Check if the matrix is already in RREF
	Print the matrix that it just read

	Show the user options for
		1 Switching rows
		2 Multiply by non-zero number
		3 Add scalar of one row to another row
		4 Quit

	Call each of the respective functions for each of the options besides Quit
	Set matrixArray to the new array from each function
	Display the matrixArray

	Get the array returned from options 1, 2, and 3 and display them
	Check if the matrix is in RREF and display the value of the variables

end
```