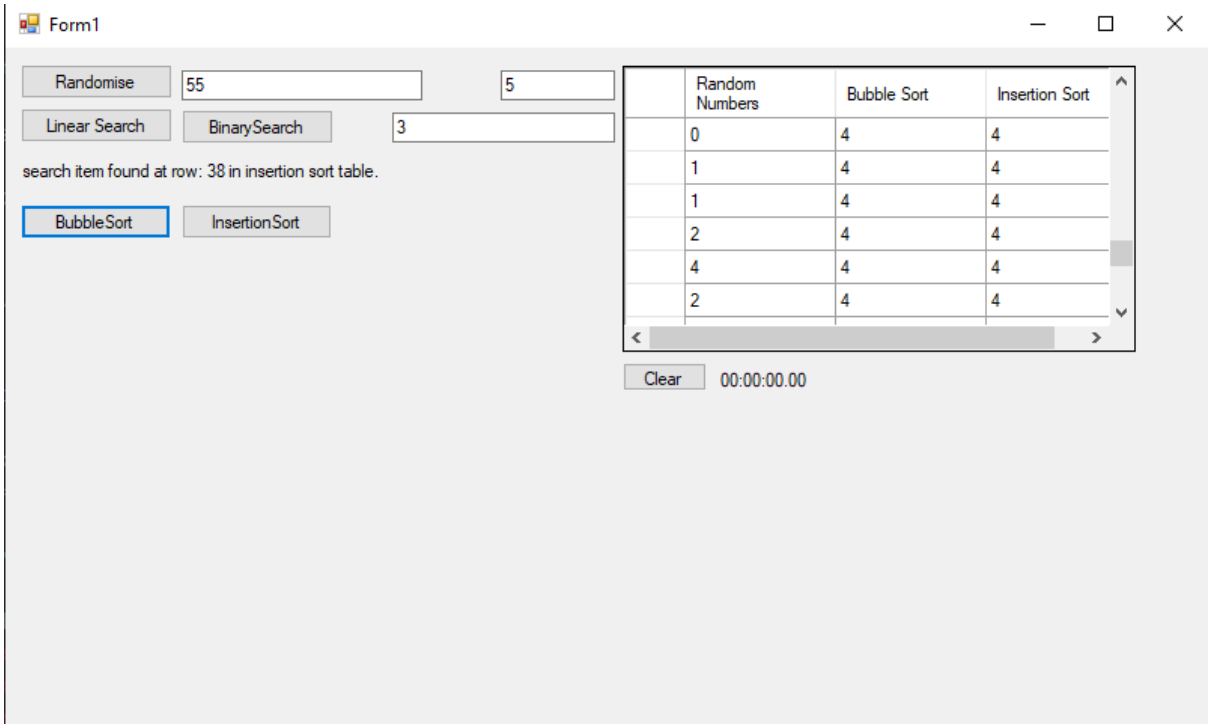

Code file:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;
using System.Diagnostics;

namespace LinearSearchUnsorted
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void BTNRandomise_Click(object sender, EventArgs e)
        {
            Random randVars = new Random();//creates var that when queried will create
a random num in a given range
            for (int randNumAmount = Convert.ToInt32(TBRandomNum.Text);randNumAmount >
1; randNumAmount--)//fills dgv with random nums
            {
                DGVOutput[0, 0].Value = randVars.Next(0,
Convert.ToInt32(TBrandLimit.Text)+1);
                DGVOutput.Rows.Insert(0);//always inserts random num in first row
            }
            DGVOutput[0, 0].Value = randVars.Next(0,
Convert.ToInt32(TBrandLimit.Text)+1);//needs extra fill to fill initial row.
```

```
        }

        private void BTNSort_Click(object sender, EventArgs e)
        {
            Stopwatch stopWatch = new Stopwatch();
            stopWatch.Start();
            bool bubbleSortComplete = false;
            for (int i = 0; i < (Convert.ToInt32(DGVOutput.RowCount)); i++)//copies
all randomly generated to the bubble sorted column. then sorts.
            {
                DGVOutput[1, i].Value = Convert.ToInt32(DGVOutput[0, i].Value);
            }

            while (bubbleSortComplete == false)
            {
                bubbleSortComplete = true;//flagged as sorted until a swap occurs
                for (int i = 0; i < (Convert.ToInt32(DGVOutput.RowCount)-1); i++)
                {
                    if (Convert.ToInt32(DGVOutput[1,i].Value) >
Convert.ToInt32(DGVOutput[1,i+1].Value))//swaps 2 vars without third using a=a+b, b=a-
b, a=a-b. not significantly more efficient but is more memory efficient
                    {
                        bubbleSortComplete = false;
                        DGVOutput[1,i].Value = Convert.ToInt32(DGVOutput[1,i].Value) +
Convert.ToInt32(DGVOutput[1,i + 1].Value);
                        DGVOutput[1,i + 1].Value =
Convert.ToInt32(DGVOutput[1,i].Value) - Convert.ToInt32(DGVOutput[1,i + 1].Value);
                        DGVOutput[1,i].Value = Convert.ToInt32(DGVOutput[1,i].Value) -
Convert.ToInt32(DGVOutput[1,i+1].Value);
                    }

                }
            }
            TimeSpan ts = stopWatch.Elapsed;
            LBLDebug.Text = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours,
ts.Minutes, ts.Seconds, ts.Milliseconds / 10);
        }

        private void BTNLinearSearch_Click(object sender, EventArgs e)
        {
            Stopwatch stopWatch = new Stopwatch();
            stopWatch.Start();
            LBLSearchOutput.Visible = true;
            for (int i = 0; i < (Convert.ToInt32(DGVOutput.RowCount)); i++)
            {
                if (Convert.ToInt32(DGVOutput[0, i].Value) ==
Convert.ToInt32(TBsearchNum.Text)) //if the item on row i in randomised column =
search time then state that.
                {
                    LBLSearchOutput.Text = "Search item found at row: " + i + " in
randomised table";
                    break;
                }
                else
                {
                    LBLSearchOutput.Text = "Search item not found.";
                }
                TimeSpan ts = stopWatch.Elapsed;
                LBLDebug.Text = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours,
ts.Minutes, ts.Seconds, ts.Milliseconds / 10);
            }
        }
```

```
        private void BTNinsertionSort_Click(object sender, EventArgs e)//create list
add 0 at start, copy templist[i] to templist[0], loop up through list checking if
templist[0] > templist[listpos]
        {
            Stopwatch stopWatch = new Stopwatch();
            stopWatch.Start();
            List<int> tempList = new List<int>();
            for (int i = 0; i < (Convert.ToInt32(DGVOutput.RowCount));i++)//creates a
temp list from the random data
            {
                tempList.Add(Convert.ToInt32(DGVOutput[0, i].Value));
            }

            for (int sortItem = 0;sortItem < tempList.Count; sortItem++)//loops
through list to select item to compare against others.
            {
                for (int compareValue = 0;compareValue <
sortItem;compareValue++)//loops up through already sorted items to find place for new
item.
                {
                  if (tempList[sortItem] <= tempList[compareValue])//checks value of
next index
                  {
                    tempList.Insert(compareValue, tempList[sortItem]);
                    tempList.RemoveAt(sortItem+1);
                    break;
                  }
                }
            }

            for (int tableInsertionIndex = 0; tableInsertionIndex < tempList.Count;
tableInsertionIndex++)//enters data from temp list into insertion sort column of dgv
                    {
                DGVOutput[2, tableInsertionIndex].Value =
tempList[tableInsertionIndex];
                    }
            TimeSpan ts = stopWatch.Elapsed;
            LBLDebug.Text = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours,
ts.Minutes, ts.Seconds, ts.Milliseconds / 10);
        }

        private void BTNgridClear_Click(object sender, EventArgs e)//clears all rows
in dgv
        {
            DGVOutput.Rows.Clear();
        }

        private void TBsearchNum_Click(object sender, EventArgs e)//clears search num
tb when clicked, needed to display text and use input conveniently
        {
            TBsearchNum.Text = "";
        }
        private void TBrandLimit_Click(object sender, EventArgs e)//clears randlimit
tb when clicked
        {
            TBrandLimit.Text = "";
        }
        private void TBRandomNum_Click(object sender, EventArgs e)//clears randomNum
tb when clicked
        {
```

```
            TBRandomNum.Text = "";
        }

        private void BTNBinarySearch_Click(object sender, EventArgs e)
        {
            Stopwatch stopWatch = new Stopwatch();
            stopWatch.Start();
            BTNinsertionSort_Click(sender, e);//creates sorted list using insertion
sort for binary search to use.
            int searchNum = Convert.ToInt32(TBsearchNum.Text);
            int startPoint = 0;//defines start and end pointers to control range of
search
            int endPoint = DGVOutput.RowCount;
            bool searchItemFound = false;
            while (startPoint!=endPoint & searchItemFound == false)//runs until the
pointers values are the same (searched whole list) or the item is stated to be found.
            {
                if (searchNum ==
Convert.ToInt32(DGVOutput[2,(startPoint+endPoint)/2].Value))//if the item is found
then exits the loop and provides an output.
                {
                    searchItemFound = true;
                    int foundRow = ((startPoint + endPoint) / 2) + 1;
                    LBLSearchOutput.Text = "search item found at row: " + foundRow + "
in insertion sort table.";
                    break;
                }
                else if (searchNum > Convert.ToInt32(DGVOutput[2, (startPoint +
endPoint) / 2].Value))//if the search item is greater than the midpoint of the
pointers then the items to the left of that midpoint are discarded
                {
                    startPoint = (startPoint + endPoint) / 2;
                }
                else if (searchNum < Convert.ToInt32(DGVOutput[2, (startPoint +
endPoint) / 2].Value))//if the search item is less than the midpoint of the pointers
then the items to the right of that midpoint are discarded.
                {
                    endPoint = (startPoint + endPoint) / 2;
                }
            }
            TimeSpan ts = stopWatch.Elapsed;//used to time searches/sorts/etc and
provides output in debut label
            LBLDebug.Text = String.Format("{0:00}:{1:00}:{2:00}.{3:00}", ts.Hours,
ts.Minutes, ts.Seconds, ts.Milliseconds / 10);
        }
    }
}
```

Designer file:

```
namespace LinearSearchUnsorted

{

  partial class Form1

  {

    /// <summary>
```

```
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.DGVOutput = new System.Windows.Forms.DataGridView();
            this.RandomNumbers = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.bubbleSort = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.InsertionSortColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.BTNRandomise = new System.Windows.Forms.Button();
```

```
        this.TBRandomNum = new System.Windows.Forms.TextBox();
        this.BTNBubbleSort = new System.Windows.Forms.Button();
        this.LBLDebug = new System.Windows.Forms.Label();
        this.BTNLinearSearch = new System.Windows.Forms.Button();
        this.TBsearchNum = new System.Windows.Forms.TextBox();
        this.LBLSearchOutput = new System.Windows.Forms.Label();
        this.TBrandLimit = new System.Windows.Forms.TextBox();
        this.BTNgridClear = new System.Windows.Forms.Button();
        this.BTNinsertionSort = new System.Windows.Forms.Button();
        this.BTNBinarySearch = new System.Windows.Forms.Button();
        ((System.ComponentModel.ISupportInitialize)(this.DGVOutput)).BeginInit();
        this.SuspendLayout();
        //
        // DGVOutput
        //
        this.DGVOutput.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.DGVOutput.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
        this.RandomNumbers,
        this.bubbleSort,
        this.InsertionSortColumn});
        this.DGVOutput.Location = new System.Drawing.Point(410, 12);
        this.DGVOutput.Name = "DGVOutput";
        this.DGVOutput.Size = new System.Drawing.Size(341, 190);
        this.DGVOutput.TabIndex = 3;
        //
        // RandomNumbers
        //
        this.RandomNumbers.HeaderText = "Random Numbers";
        this.RandomNumbers.Name = "RandomNumbers";
        //
```

```
// bubbleSort
//
this.bubbleSort.HeaderText = "Bubble Sort";
this.bubbleSort.Name = "bubbleSort";
//
// InsertionSortColumn
//
this.InsertionSortColumn.HeaderText = "Insertion Sort";
this.InsertionSortColumn.Name = "InsertionSortColumn";
//
// BTNRandomise
//
this.BTNRandomise.Location = new System.Drawing.Point(10, 11);
this.BTNRandomise.Name = "BTNRandomise";
this.BTNRandomise.Size = new System.Drawing.Size(101, 23);
this.BTNRandomise.TabIndex = 4;
this.BTNRandomise.Text = "Randomise";
this.BTNRandomise.UseVisualStyleBackColor = true;
this.BTNRandomise.Click += new System.EventHandler(this.BTNRandomise_Click);
//
// TBRandomNum
//
this.TBRandomNum.Location = new System.Drawing.Point(117, 15);
this.TBRandomNum.Name = "TBRandomNum";
this.TBRandomNum.Size = new System.Drawing.Size(160, 20);
this.TBRandomNum.TabIndex = 5;
this.TBRandomNum.Text = "Num Of Randoms To Generate";
this.TBRandomNum.Click += new System.EventHandler(this.TBRandomNum_Click);
//
// BTNBubbleSort
//
```

```
this.BTNBubbleSort.Location = new System.Drawing.Point(10, 104);
this.BTNBubbleSort.Name = "BTNBubbleSort";
this.BTNBubbleSort.Size = new System.Drawing.Size(100, 23);
this.BTNBubbleSort.TabIndex = 6;
this.BTNBubbleSort.Text = "BubbleSort";
this.BTNBubbleSort.UseVisualStyleBackColor = true;
this.BTNBubbleSort.Click += new System.EventHandler(this.BTNSort_Click);
//
// LBLDebug
//
this.LBLDebug.AutoSize = true;
this.LBLDebug.Location = new System.Drawing.Point(472, 214);
this.LBLDebug.Name = "LBLDebug";
this.LBLDebug.Size = new System.Drawing.Size(39, 13);
this.LBLDebug.TabIndex = 7;
this.LBLDebug.Text = "Debug";
//
// BTNLinearSearch
//
this.BTNLinearSearch.Location = new System.Drawing.Point(10, 40);
this.BTNLinearSearch.Name = "BTNLinearSearch";
this.BTNLinearSearch.Size = new System.Drawing.Size(101, 23);
this.BTNLinearSearch.TabIndex = 8;
this.BTNLinearSearch.Text = "Linear Search";
this.BTNLinearSearch.UseVisualStyleBackColor = true;
this.BTNLinearSearch.Click += new System.EventHandler(this.BTNLinearSearch_Click);
//
// TBsearchNum
//
this.TBsearchNum.Location = new System.Drawing.Point(257, 43);
this.TBsearchNum.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
```

```
this.TBsearchNum.Name = "TBsearchNum";
this.TBsearchNum.Size = new System.Drawing.Size(148, 20);
this.TBsearchNum.TabIndex = 9;
this.TBsearchNum.Text = "Search Number";
this.TBsearchNum.Click += new System.EventHandler(this.TBsearchNum_Click);
//
// LBLSearchOutput
//
this.LBLSearchOutput.AutoSize = true;
this.LBLSearchOutput.Location = new System.Drawing.Point(9, 75);
this.LBLSearchOutput.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0);
this.LBLSearchOutput.Name = "LBLSearchOutput";
this.LBLSearchOutput.Size = new System.Drawing.Size(73, 13);
this.LBLSearchOutput.TabIndex = 10;
this.LBLSearchOutput.Text = "SearchOutput";
this.LBLSearchOutput.Visible = false;
//
// TBrandLimit
//
this.TBrandLimit.Location = new System.Drawing.Point(329, 15);
this.TBrandLimit.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
this.TBrandLimit.Name = "TBrandLimit";
this.TBrandLimit.Size = new System.Drawing.Size(76, 20);
this.TBrandLimit.TabIndex = 11;
this.TBrandLimit.Text = "Random Limit";
this.TBrandLimit.Click += new System.EventHandler(this.TBrandLimit_Click);
//
// BTNgridClear
//
this.BTNgridClear.Location = new System.Drawing.Point(410, 209);
this.BTNgridClear.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
```

```
this.BTNgridClear.Name = "BTNgridClear";
this.BTNgridClear.Size = new System.Drawing.Size(56, 19);
this.BTNgridClear.TabIndex = 12;
this.BTNgridClear.Text = "Clear";
this.BTNgridClear.UseVisualStyleBackColor = true;
this.BTNgridClear.Click += new System.EventHandler(this.BTNgridClear_Click);
//
// BTNinsertionSort
//
this.BTNinsertionSort.Location = new System.Drawing.Point(117, 104);
this.BTNinsertionSort.Name = "BTNinsertionSort";
this.BTNinsertionSort.Size = new System.Drawing.Size(100, 23);
this.BTNinsertionSort.TabIndex = 13;
this.BTNinsertionSort.Text = "InsertionSort";
this.BTNinsertionSort.UseVisualStyleBackColor = true;
this.BTNinsertionSort.Click += new System.EventHandler(this.BTNinsertionSort_Click);
//
// BTNBinarySearch
//
this.BTNBinarySearch.Location = new System.Drawing.Point(117, 41);
this.BTNBinarySearch.Name = "BTNBinarySearch";
this.BTNBinarySearch.Size = new System.Drawing.Size(101, 23);
this.BTNBinarySearch.TabIndex = 14;
this.BTNBinarySearch.Text = "BinarySearch";
this.BTNBinarySearch.UseVisualStyleBackColor = true;
this.BTNBinarySearch.Click += new System.EventHandler(this.BTNBinarySearch_Click);
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
```

```
    this.ClientSize = new System.Drawing.Size(800, 450);
    this.Controls.Add(this.BTNBinarySearch);
    this.Controls.Add(this.BTNinsertionSort);
    this.Controls.Add(this.BTNgridClear);
    this.Controls.Add(this.TBrandLimit);
    this.Controls.Add(this.LBLSearchOutput);
    this.Controls.Add(this.TBsearchNum);
    this.Controls.Add(this.BTNLinearSearch);
    this.Controls.Add(this.LBLDebug);
    this.Controls.Add(this.BTNBubbleSort);
    this.Controls.Add(this.TBRandomNum);
    this.Controls.Add(this.BTNRandomise);
    this.Controls.Add(this.DGVOutput);
    this.Name = "Form1";
    this.Text = "Form1";
    ((System.ComponentModel.ISupportInitialize)(this.DGVOutput)).EndInit();
    this.ResumeLayout(false);
    this.PerformLayout();

}

#endregion
private System.Windows.Forms.DataGridView DGVOutput;
private System.Windows.Forms.Button BTNRandomise;
private System.Windows.Forms.TextBox TBRandomNum;
private System.Windows.Forms.Button BTNBubbleSort;
private System.Windows.Forms.Label LBLDebug;
private System.Windows.Forms.Button BTNLinearSearch;
private System.Windows.Forms.TextBox TBsearchNum;
private System.Windows.Forms.Label LBLSearchOutput;
private System.Windows.Forms.TextBox TBrandLimit;
```

```
        private System.Windows.Forms.Button BTNgridClear;
        private System.Windows.Forms.Button BTNinsertionSort;
        private System.Windows.Forms.DataGridViewTextBoxColumn RandomNumbers;
        private System.Windows.Forms.DataGridViewTextBoxColumn bubbleSort;
        private System.Windows.Forms.DataGridViewTextBoxColumn InsertionSortColumn;
        private System.Windows.Forms.Button BTNBinarySearch;
    }
}
```