```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;

namespace Assign_1
{
    /** Matthew Alunni
     *  5865647
     *  COSC 3P71
     *  Assignment 1 **/

    public partial class Form1 : Form
    {

        List<Solution> solutions = new List<Solution>(); //list of solutions
        int current = -1; //which solution is being displayed

        public Form1()
        {
            InitializeComponent();
        }

        /** this method finds a solution and prints it to the board on click**/
        private void btnDrawBoard_Click(object sender, EventArgs e)
        {
            FindSolutions();

            if (solutions.Count > 0)
            {
                current = 0;
                PrintSolution();
            }
        }

        /** this method finds solutions, if the user inputs a valid number**/
        private void FindSolutions()
        {
            solutions = new List<Solution>();

            current = 0;
            var pos = new Position(0, Int32.MinValue, null);

            if (numericNumberOfQueens.Value >=2 && numericNumberOfQueens.Value < 
              4)
            {
                System.Windows.Forms.MessageBox.Show("Please enter a valid 
                  number.");
            }
            else
            {
                pos.FindSolution(solutions, Convert.ToInt32
                        (numericNumberOfQueens.Value), 0);
            }
        }


        /** this method is used for the bottom scroller to print solutions**/
        public void PrintSolution()
        {
            if (current == 0)
            {
                buttonFirst.Enabled = false;
                buttonPrevious.Enabled = false;
            }
            else
            {
                buttonFirst.Enabled = true;
                buttonPrevious.Enabled = true;
            }

            if (current == solutions.Count - 1)
            {
                buttonNext.Enabled = false;
                buttonLast.Enabled = false;
            }
            else
            {
                buttonNext.Enabled = true;
                buttonLast.Enabled = true;
            }


            labelCost.Text = string.Format("Heuristic cost {0}", solutions
              [current].Cost);
            labelResults.Text = string.Format("Solution {0} of {1}", current + 1,
               solutions.Count);

            Position Node = solutions[current].Position;

            panel1.Visible = false;

            PictureBox[,] theBoard = createBoard(Convert.ToInt32
              (numericNumberOfQueens.Value));

            while (Node.Row >= 0)
            {
                theBoard[Node.Row, Node.Line - 1].Image =
                  Assign_1.Properties.Resources.crown;
                theBoard[Node.Row, Node.Line - 1].SizeMode =
                  PictureBoxSizeMode.StretchImage;
                Node = Node.Parent;
            }
            panel1.Visible = true;
        }

        /**  this method sets up the board**/
        public PictureBox[,] createBoard(int size)
        {
            panel1.Controls.Clear();

            PictureBox[,] board = new PictureBox[size, size];
            int w = 0, h = 0;
            w = panel1.Width;
            h = panel1.Height;
            int horizontal = (int)((double)w / (double)size);
            int vertical = (int)((double)h / (double)size);
            for (int len = 0; len < size; len++)
                for (int wid = 0; wid < size; wid++)
                {
                    board[len, wid] = new PictureBox();
                    board[len, wid].Parent = panel1;
                    board[len, wid].Location = new Point(wid * horizontal + 1,
                     len * vertical + 1);
                    board[len, wid].Size = new Size(horizontal, vertical);
                    if ((len + wid) % 2 == 0)
                        board[len, wid].BackColor = Color.White;
                    else
                        board[len, wid].BackColor = Color.Black;
                }

            return board;
        }

        private void buttonPrevious_Click(object sender, EventArgs e)
        {
            current--;
            PrintSolution();
        }

        private void buttonNext_Click(object sender, EventArgs e)
        {
            current++;
            PrintSolution();

        }

        private void buttonFirst_Click(object sender, EventArgs e)
        {
            current = 0;
            PrintSolution();
        }

        private void buttonLast_Click(object sender, EventArgs e)
        {
            current = solutions.Count -1;
```

```
                PrintSolution();
            }

            private void buttonHeuristic_Click(object sender, EventArgs e)
            {
                FindSolutions();
                //solutions = solutions.Sort(;
                solutions.Sort((x, y) => x.Cost.CompareTo(y.Cost));

                if (solutions.Count > 0)
                {
                    current = 0;
                    PrintSolution();
                }


            }

            private void label1_Click(object sender, EventArgs e)
            {

            }

            private void numericNumberOfQueens_ValueChanged(object sender, EventArgs e)
            {

            }
        }


    }

```