```
using System;
using System.Collections.Generic;

namespace EQL
{
    public class Solver
    {
        public Solver()
        {
        }

        public List<Tuple<int,int>> Solve()
        {
            int[,] b = new int[8, 8];
            for (int i = 0; i < 8; i++)
            {
                for (int j = 0; j < 8; j++)
                {
                    b[i, j] = 0;
                }
            }
            var starter = 0;
            for (int i = 0; i < 8; i++)
            {
                var f = false;
                for (int j = starter; j < 8; j++)
                {
                    if (b[i, j] == 0)
                    {
                        f = true;
                        b[i, j] = 1;
                        for (int k = 0; k < 8; k++)
                        {
                            if (b[i, k] != 1)
                            {
                                b[i, k] = 2;
                            }
                        }
                        for (int l = 0; l < 8; l++)
                        {
                            if (b[l, j] != 1)
                            {
                                b[l, j] = 2;
                            }
                        }
                        for (int k = j, l = i; k < 8 && l < 8; k++, l++)
                        {
                            if (b[l, k] != 1)
                            {
                                b[l, k] = 2;
                            }
```

```
                for (int k = j, l = i; k < 8 && l >= 0; k++, l--)
                {
                    if (b[l, k] != 1)
                    {
                        b[l, k] = 2;
                    }

                }
                for (int k = j, l = i; k >= 0 && l < 8; k--, l++)
                {
                    if (b[l, k] != 1)
                    {
                        b[l, k] = 2;
                    }
                }
                for (int k = j, l = i; k >= 0 && l >= 0; k--, l--)
                {
                    if (b[l, k] != 1)
                    {
                        b[l, k] = 2;
                    }
                }
                break;
            }
        }
        if (!f)
        {
            i = i - 1;
            for (int j = 0; j < 8; j++)
            {
                if (b[i, j] == 1)
                {
                    b[i, j] = 0;
                    starter = j + 1;
                }
            }
            i = i - 1;
            for (int k = 0; k < 8; k++)
            {
                for (int j = 0; j < 8; j++)
                {
                    if (b[k, j] != 1)
                    {
                        b[k, j] = 0;
                    }
                }
            }
            for (int x = 0; x < 8; x++)
            {
                for (int y = 0; y < 8; y++)
                {
                    if (b[x, y] == 1)
```

```
                    for (int k = 0; k < 8; k++)
                    {
                        if (b[x, k] != 1)
                        {
                            b[x, k] = 2;
                        }
                    }
                    for (int l = 0; l < 8; l++)
                    {
                        if (b[l, y] != 1)
                        {
                            b[l, y] = 2;
                        }
                    }
                    for (int k = y, l = x; k < 8 && l < 8; k++, l++)
                    {
                        if (b[l, k] != 1)
                        {
                            b[l, k] = 2;
                        }
                    }
                    for (int k = y, l = x; k < 8 && l >= 0; k++, l--
                    {
                        if (b[l, k] != 1)
                        {
                            b[l, k] = 2;
                        }
                    }
                    for (int k = y, l = x; k >= 0 && l < 8; k--, l++
                    {
                        if (b[l, k] != 1)
                        {
                            b[l, k] = 2;
                        }
                    }
                    for (int k = y, l = x; k >= 0 && l >= 0; k--, l-
                    {
                        if (b[l, k] != 1)
                        {
                            b[l, k] = 2;
                        }
                    }
                }
            }
        }



    } else
    {
```

```
            }
        }









        // The below code gets all the found cells
        // in b to return as the solution to the problem
        List<Tuple<int,int>> result = new List<Tuple<int,int>>();
        for (int i = 0; i < 8; i++)
        {
            for (int j = 0; j < 8; j++)
            {
                if (b[i, j] == 1)
                {
                    result.Add(new Tuple<int, int>(i, j));
                }
            }
        }

        return result;
    }
  }
}
```