```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Win32;


namespace WpfApplication1
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    ///
    // LIST FUNCTIONALITY

    public class GenericList<Tip>
    {
        public class Node
        {
            public Node Prev;
            public Tip Data;
            public int Index;
        }

        public Node head = null;
        public Node current = null;


        // -

        public void Reset()
        {
            current = head;
            if (head != null)
            {
```

```
            AscendingByIndex();
        }
    }

    // -

    public void AddNode(Tip t, int sortIndex) //, int index)
    {
        Node newNode = new Node();

        newNode.Prev = head;
        newNode.Data = t;
        newNode.Index = sortIndex;
        head = newNode;

        AscendingByIndex();

        if (current == null)
        {
            current = head;
        }
    }

    // -

    public Tip GetFirst()
    {
        Tip temp = default(Tip);

        Node current = head;
        while (current != null)
        {
            temp = current.Data;
            current = current.Prev;
        }
        return temp;
    }

    // -

    public Tip GetNode()
    {
        if (current == null)
        {
            return default(Tip);
            }
            else
            {
                return current.Data;
            }
        }

        // -

        public bool MoveNext()
        {
            if (current == null)
            {
                return false;
            }
            if (current.Prev == null)
            {
                return false;
            }
            current = current.Prev;
            return true;
        }

        // -

        public void AscendingByIndex()
        {
            Node init = null;
            Node temp = new Node();
            bool k = true;

            // Sort ascending by list index ('ListIndex' field exist in every struct)
            while (k)
            {

                k = false;
                init = head;

                while (init.Prev != null)
                {
                    if (init.Index > init.Prev.Index)
                    {
                        // The ole switch-a-roo
                        temp.Data = init.Data;
                        temp.Index = init.Index;
                    init.Data = init.Prev.Data;
                    init.Index = init.Prev.Index;

                    init.Prev.Data = temp.Data;
                    init.Prev.Index = temp.Index;

                    k = true;
                }
                else
                {
                    init = init.Prev;
                }
            }
        }
    }

    // -

    public bool notNull()
    {
        if (head == null)
        {
            return false;
        }
        else
        {
            return true;
        }
    }

    // -

    public int Count()
    {
        int count = 0;
        Node temp = head;

        while (temp != null)
        {
            count++;
            temp = temp.Prev;
        }

        return count;
    }

    // -

    public bool RemoveNode(int index)
    {
        bool found = false;
        Node init = head;
        int i = 0;

        while (init.Prev != null && !found)
        {
            if (index == i + 1)
            {
                found = true;
                init.Prev = init.Prev.Prev;
            }
            else
            {
                init = init.Prev;
                i++;
            }
        }

        return true;
    }

    // -

    public void ClearList()
    {
        head = null;
        current = null;
    }

} // SFARSIT LISTA
```

```
// HEROES

struct Hero
{
    public int heroID;
    public string heroName;
    // public System.Drawing.Image heroImage;
}


class Heroes
{
    public GenericList<Hero> heroes;
    public Hero hero;

    public Heroes()
    {
        heroes = new GenericList<Hero>();
        //
    }

    public void AddHero(int ID, string nume) //, System.Drawing.Image avatar)
    {
        hero = new Hero();

        hero.heroID = ID;
        hero.heroName = nume;
        // hero.heroImage = avatar;

        heroes.AddNode(hero, hero.heroID);
    }
}



// MAPS

struct Map
{
    public int[,] Synergy;
    public int[,] Counter;
}
```

```
class Maps
{
    public GenericList<Map> maps;
    public Map map;

    public Maps(int mapID = 0)
    {
        // maps.head.Data.Counter = null;
    }


    public int[] getSuggestions()
    {
        int Synergy = 0, Counter = 0;
        int[] H = { 0 };

        // Analize ranks

        return H;
    }

    public int[] markFriendly()
    {

        return getSuggestions();
    }

    public int[] markEnemy()
    {
        return getSuggestions();
    }
}



public partial class MainWindow : Window
{

    Map[] maps = new Map[9];
    Hero[] heroes = new Hero[5];

    Maps mapsX = new Maps(0);
    Heroes heroesX = new Heroes();

    public int[] Friendlys;
public int[] Enemies;

string newHeroName = null;
System.Drawing.Image newImageAvatar = null;



public MainWindow()
{
    InitializeComponent();
}



private void button1_Click(object sender, RoutedEventArgs e)
{
    int[,] M = new int[52, 52];

    // Create an instance of the open file dialog box.
    OpenFileDialog openFileDialog1 = new OpenFileDialog();

    // Set filter options and filter index.
    openFileDialog1.Filter = "Text Files (.txt)|*.txt|All Files (*.*)|*.*";
    openFileDialog1.FilterIndex = 1;

    openFileDialog1.Multiselect = true;

    // Call the ShowDialog method to show the dialog box.
    bool? userClickedOK = openFileDialog1.ShowDialog();

    // Process input if the user clicked OK.
    if (userClickedOK == true)
    {
        // Open the selected file to read.
        //System.IO.Stream fileStream = openFileDialog1.File.OpenRead();
        string[] lines = System.IO.File.ReadAllLines(openFileDialog1.FileName);

        for(int i = 0; i < lines.Length; i++)
        {
            string[] line = lines[i].Split(' ');
            for(int j = 0; j < line.Length; j++)
            {
                M[i, j] = Convert.ToInt16(line[j]);
            }
        }
```

```
            // maps[0] = new Map(52, M);
            // maps[0] = new Map(52, M);
            //fileStream.Close();

        }
    }



    private void buttonLoadHeroes_Click(object sender, RoutedEventArgs e)
    {

        Image imageX = new Image();
        System.Drawing.Image image;
        BitmapImage imageBMP = new BitmapImage();

        string[] lines = null, line = null;
        Window1 popUp = new Window1();

        OpenFileDialog fileDialog = new OpenFileDialog();
        bool? userClickedOK = fileDialog.ShowDialog();
        if (userClickedOK == true)
        {
            lines = System.IO.File.ReadAllLines(fileDialog.FileName);
        }

        for(int i = 0; i < lines.Count(); i++)
        {
            line = lines[i].Split(' ');
            // System.IO.MemoryStream ms = new System.IO.MemoryStream(Convert.FromBase64String(line[2]));
            // image = System.Drawing.Image.FromStream(ms);

            heroesX.AddHero(Convert.ToInt16(line[0]), line[1]);
            popUp.listBoxListaEroi.Items.Add(line[1]);
        }
        popUp.Show();
    }



    private void buttonLoadAvatar_Click(object sender, RoutedEventArgs e)
    {
        // newImageAvatar
        Image showImage = new Image();
```

```
            BitmapImage showImageBMP = new BitmapImage();
            System.IO.MemoryStream ms = new System.IO.MemoryStream();

            OpenFileDialog fileDialog = new OpenFileDialog();
            bool? userClickedOK = fileDialog.ShowDialog();
            if(userClickedOK == true)
            {
                newImageAvatar = System.Drawing.Image.FromFile(fileDialog.FileName);
                showImageBMP.BeginInit();

                // showImageBMP.BaseUri = new Uri(fileDialog.FileName, UriKind.Absolute);
                showImageBMP.UriSource = new Uri(fileDialog.FileName, UriKind.Absolute);
                showImageBMP.EndInit();

                showImage.Source = showImageBMP;
                showImage.Height = 89;
                showImage.Width = 89;
                canvasAvatar.Children.Clear();
                canvasAvatar.Children.Add(showImage);
            }
        }



        private void buttonSaveNewHero_Click(object sender, RoutedEventArgs e)
        {
            string[] line = new string[1];
            System.IO.MemoryStream ms = new System.IO.MemoryStream();

            OpenFileDialog fileDialog = new OpenFileDialog();
            bool? userClickedOK = fileDialog.ShowDialog();
            if(userClickedOK == true)
            {
                line[0] = "0" + " " + textBoxNewHeroName.Text + " " +
System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes(newImageAvatar.ToString()));
                System.IO.File.AppendAllLines(fileDialog.FileName, line);
            }

        }


    }

}
```