

Feladat:

Írj egy C# konzolos alkalmazást, amely tartalmaz négy különböző függvényt, és teszteld őket az NUnit keretrendszer segítségével a következő Assert metódusokkal: `IsTrue`, `IsFalse`, `AreEqual`, `Greater`, `Less`.

Függvények:

1. Létezik-e páros szám a listában?

- Függvény: `bool ContainsEvenNumber(List<int> numbers)`
- Tétel: Keresés
- Assert: `Assert.IsTrue` (ha létezik páros szám).

2. Minden szám pozitív-e?

- Függvény: `bool AreAllPositive(List<int> numbers)`
- Tétel: Kiválogatás
- Assert: `Assert.IsFalse` (ha van negatív szám).

3. Legnagyobb szám a listában.

- Függvény: `int FindMax(List<int> numbers)`
- Tétel: Maximumkeresés
- Assert: `Assert.Greater` (legyen nagyobb, mint egy adott érték, pl. 10).

4. Legkisebb szám helye a listában.

- Függvény: `int FindMinIndex(List<int> numbers)`
- Tétel: Minimumkeresés
- Assert: `Assert.Less` (ellenőrizd, hogy a legkisebb szám indexe kisebb, mint a lista hossza).

Kód:

Fő Program:

```
using System;

using System.Collections.Generic;

public class ListProcessor
{
```

```
public bool ContainsEvenNumber(List<int> numbers)
{
    foreach (var number in numbers)
    {
        if (number % 2 == 0)
            return true;
    }
    return false;
}
```

```
public bool AreAllPositive(List<int> numbers)
{
    foreach (var number in numbers)
    {
        if (number < 0)
            return false;
    }
    return true;
}
```

```
public int FindMax(List<int> numbers)
{
    if (numbers.Count == 0) throw new ArgumentException("List is empty");
    int max = numbers[0];
    foreach (var number in numbers)
    {
        if (number > max)
            max = number;
    }
}
```

```

        return max;
    }

    public int FindMinIndex(List<int> numbers)
    {
        if (numbers.Count == 0) throw new ArgumentException("List is empty");
        int minIndex = 0;
        for (int i = 1; i < numbers.Count; i++)
        {
            if (numbers[i] < numbers[minIndex])
                minIndex = i;
        }
        return minIndex;
    }
}

```

Tesztosztály:

```

using NUnit.Framework;
using System.Collections.Generic;

[TestFixture]
public class ListProcessorTests
{
    private ListProcessor processor;

    [Setup]
    public void Setup()
    {
        processor = new ListProcessor();
    }
}

```

[Test]

public void TestContainsEvenNumber()

{

var numbers = new List<int> { 1, 3, 5, 8 };

Assert.IsTrue(processor.ContainsEvenNumber(numbers));

}

[Test]

public void TestAreAllPositive()

{

var numbers = new List<int> { 1, -3, 5, 8 };

Assert.IsFalse(processor.AreAllPositive(numbers));

}

[Test]

public void TestFindMax()

{

var numbers = new List<int> { 1, 3, 5, 8 };

Assert.Greater(processor.FindMax(numbers), 5);

}

[Test]

public void TestFindMinIndex()

{

var numbers = new List<int> { 10, 3, 5, 8 };

Assert.Less(processor.FindMinIndex(numbers), numbers.Count);

}

}

Magyarázat:

1. A **ContainsEvenNumber** függvény keresést valósít meg, és azt vizsgálja, hogy van-e páros szám a listában. Az `Assert.IsTrue` teszteli, hogy ez az állítás igaz.
2. A **AreAllPositive** függvény kiválogatást valósít meg, és megállapítja, hogy minden szám pozitív-e. Az `Assert.IsFalse` ellenőrzi, hogy van-e negatív szám.
3. A **FindMax** függvény maximumkeresést hajt végre, és megkeresi a lista legnagyobb elemét. Az `Assert.Greater` ellenőrzi, hogy ez az érték meghalad egy küszöböt.
4. A **FindMinIndex** függvény minimumkeresést végez, és megadja a legkisebb szám indexét. Az `Assert.Less` garantálja, hogy az index a lista érvényes tartományába esik.