

Άσκηση 1: Απλή Ταξινόμηση

Περιγραφή: Δημιουργήστε ένα πρόγραμμα που ταξινομεί έναν πίνακα ακεραίων χρησιμοποιώντας την ταξινόμηση με εισαγωγή.

Λύση:

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 5, 2, 9, 1, 5, 6 };
        InsertionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void InsertionSort(int[] array)
    {
        for (int i = 1; i < array.Length; i++)
        {
            int key = array[i];
            int j = i - 1;

            while (j >= 0 && array[j] > key)
            {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = key;
        }
    }
}
```

Άσκηση 2: Ταξινόμηση με Εισαγωγή σε Φθίνουσα Σειρά

Περιγραφή: Τροποποιήστε τον αλγόριθμο για να ταξινομήσετε τον πίνακα σε φθίνουσα σειρά.

Λύση:

```
using System;

class Program
```

```

{
    static void Main()
    {
        int[] array = { 5, 2, 9, 1, 5, 6 };
        InsertionSortDescending(array);
        Console.WriteLine("Φθινομενικά ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void InsertionSortDescending(int[] array)
    {
        for (int i = 1; i < array.Length; i++)
        {
            int key = array[i];
            int j = i - 1;

            while (j >= 0 && array[j] < key)
            {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = key;
        }
    }
}

```

Άσκηση 3: Ταξινόμηση Λίστας Ακεραίων

Περιγραφή: Δημιουργήστε μια μέθοδο που δέχεται μια λίστα ακεραίων και επιστρέφει τη ταξινομημένη λίστα χρησιμοποιώντας την ταξινόμηση με εισαγωγή.

Λύση:

```

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<int> numbers = new List<int> { 5, 2, 9, 1, 5, 6 };
        List<int> sortedNumbers = InsertionSortList(numbers);
        Console.WriteLine("Ταξινομημένη λίστα: " + string.Join(", ", sortedNumbers));
    }

    static List<int> InsertionSortList(List<int> list)
    {

```

```

for (int i = 1; i < list.Count; i++)
{
    int key = list[i];
    int j = i - 1;

    while (j >= 0 && list[j] > key)
    {
        list[j + 1] = list[j];
        j--;
    }
    list[j + 1] = key;
}
return list;
}
}

```

Άσκηση 4: Ταξινόμηση Μη Ακεραίων (Πραγματικοί Αριθμοί)

Περιγραφή: Εφαρμόστε την ταξινόμηση με εισαγωγή σε έναν πίνακα πραγματικών αριθμών.

Λύση:

```

using System;

class Program
{
    static void Main()
    {
        double[] array = { 5.2, 2.3, 9.1, 1.5, 5.6, 6.4 };
        InsertionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void InsertionSort(double[] array)
    {
        for (int i = 1; i < array.Length; i++)
        {
            double key = array[i];
            int j = i - 1;

            while (j >= 0 && array[j] > key)
            {

```

```

        array[j + 1] = array[j];
        j--;
    }
    array[j + 1] = key;
}
}
}

```

Άσκηση 5: Ταξινόμηση Χαρακτήρων

Περιγραφή: Εφαρμόστε την ταξινόμηση με εισαγωγή σε έναν πίνακα χαρακτήρων.

Λύση:

```
using System;
```

```

class Program
{
    static void Main()
    {
        char[] array = { 'd', 'a', 'c', 'b', 'e' };
        InsertionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(" ", array));
    }

    static void InsertionSort(char[] array)
    {
        for (int i = 1; i < array.Length; i++)
        {
            char key = array[i];
            int j = i - 1;

            while (j >= 0 && array[j] > key)
            {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = key;
        }
    }
}

```

Άσκηση 6: Ταξινόμηση και Εκτύπωση Χρονικών Δεδομένων

Περιγραφή: Δημιουργήστε έναν πίνακα με ημερομηνίες και ταξινομήστε τον.

Λύση:

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        DateTime[] dates = {
```

```
            new DateTime(2023, 5, 1),
```

```
            new DateTime(2022, 1, 12),
```

```
            new DateTime(2023, 4, 5),
```

```
            new DateTime(2022, 6, 25)
```

```
        };
```

```
        InsertionSort(dates);
```

```
        Console.WriteLine("Ταξινομημένες ημερομηνίες: " + string.Join(", ", dates));
```

```
    }
```

```
    static void InsertionSort(DateTime[] array)
```

```
    {
```

```
        for (int i = 1; i < array.Length; i++)
```

```
        {
```

```
            DateTime key = array[i];
```

```
            int j = i - 1;
```

```
            while (j >= 0 && array[j] > key)
```

```
            {
```

```
                array[j + 1] = array[j];
```

```
                j--;
```

```
            }
```

```
            array[j + 1] = key;
```

```
        }
```

```
    }
```

```
}
```