

Άσκηση 1: Βασική Υλοποίηση

Περιγραφή: Υλοποιήστε τη διαδικασία ταξινόμησης με επιλογή και ελέγξτε την με έναν πίνακα ακεραίων.

Λύση:

```
using System;
```

```
class Program
```

```
{  
    static void Main()  
    {  
        int[] array = { 64, 25, 12, 22, 11 };  
        SelectionSort(array);  
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));  
    }  
}
```

```
static void SelectionSort(int[] array)
```

```
{  
    for (int i = 0; i < array.Length - 1; i++)  
    {  
        int minIndex = i;  
        for (int j = i + 1; j < array.Length; j++)  
        {  
            if (array[j] < array[minIndex])  
            {  
                minIndex = j;  
            }  
        }  
        if (minIndex != i)  
        {  
            int temp = array[i];  
            array[i] = array[minIndex];  
            array[minIndex] = temp;  
        }  
    }  
}
```

Άσκηση 2: Ταξινόμηση Πίνακα με Θέσεις

Περιγραφή: Δημιουργήστε έναν πίνακα τυχαίων αριθμών και ταξινομήστε τον, εκτυπώνοντας τις θέσεις των στοιχείων.

Λύση:

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 20, 5, 30, 15, 10 };
        Console.WriteLine("Αρχικός πίνακας: " + string.Join(", ", array));
        SelectionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void SelectionSort(int[] array)
    {
        for (int i = 0; i < array.Length - 1; i++)
        {
            int minIndex = i;
            for (int j = i + 1; j < array.Length; j++)
            {
                if (array[j] < array[minIndex])
                {
                    minIndex = j;
                }
            }
            if (minIndex != i)
            {
                Swap(ref array[i], ref array[minIndex]);
                Console.WriteLine($"Αντάλλαξα {array[minIndex]} με {array[i]}");
            }
        }
    }

    static void Swap(ref int a, ref int b)
    {
        int temp = a;
        a = b;
        b = temp;
    }
}
```

Άσκηση 3: Ταξινόμηση με Δεύτερο Πίνακα

Περιγραφή: Δημιουργήστε μια δεύτερη μέθοδο που επιστρέφει έναν νέο ταξινομημένο πίνακα αντί να ταξινομεί τον αρχικό.

Λύση:

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 5, 3, 8, 6, 2 };
        int[] sortedArray = SelectionSort(array);
        Console.WriteLine("Αρχικός πίνακας: " + string.Join(" ", array));
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(" ", sortedArray));
    }

    static int[] SelectionSort(int[] array)
    {
        int[] sortedArray = (int[])array.Clone();
        for (int i = 0; i < sortedArray.Length - 1; i++)
        {
            int minIndex = i;
            for (int j = i + 1; j < sortedArray.Length; j++)
            {
                if (sortedArray[j] < sortedArray[minIndex])
                {
                    minIndex = j;
                }
            }
            if (minIndex != i)
            {
                Swap(ref sortedArray[i], ref sortedArray[minIndex]);
            }
        }
        return sortedArray;
    }

    static void Swap(ref int a, ref int b)
    {
        int temp = a;
        a = b;
        b = temp;
    }
}
```

Άσκηση 4: Ταξινόμηση Δεκαδικών Αριθμών

Περιγραφή: Επεκτείνετε την ταξινόμηση με επιλογή ώστε να ταξινομείτε και δεκαδικούς αριθμούς (float).

Λύση:

```
using System;

class Program
{
    static void Main()
    {
        float[] array = { 64.5f, 25.3f, 12.1f, 22.4f, 11.7f };
        SelectionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(", ", array));
    }

    static void SelectionSort(float[] array)
    {
        for (int i = 0; i < array.Length - 1; i++)
        {
            int minIndex = i;
            for (int j = i + 1; j < array.Length; j++)
            {
                if (array[j] < array[minIndex])
                {
                    minIndex = j;
                }
            }
            if (minIndex != i)
            {
                Swap(ref array[i], ref array[minIndex]);
            }
        }
    }

    static void Swap(ref float a, ref float b)
    {
        float temp = a;
        a = b;
        b = temp;
    }
}
```

Άσκηση 5: Αντιστροφή Ταξινόμησης

Περιγραφή: Επεκτείνετε τον αλγόριθμο για να ταξινομείτε τους αριθμούς σε φθίνουσα σειρά.

Λύση:

```
using System;
```

```
class Program
```

```
{  
    static void Main()  
    {  
        int[] array = { 64, 25, 12, 22, 11 };  
        SelectionSortDescending(array);  
        Console.WriteLine("Ταξινομημένος πίνακας (φθίνουσα): " + string.Join(", ", array));  
    }  
}
```

```
static void SelectionSortDescending(int[] array)
```

```
{  
    for (int i = 0; i < array.Length - 1; i++)  
    {  
        int maxIndex = i;  
        for (int j = i + 1; j < array.Length; j++)  
        {  
            if (array[j] > array[maxIndex])  
            {  
                maxIndex = j;  
            }  
        }  
        if (maxIndex != i)  
        {  
            Swap(ref array[i], ref array[maxIndex]);  
        }  
    }  
}
```

```
static void Swap(ref int a, ref int b)
```

```
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```

Άσκηση 6: Ταξινόμηση Σειράς Αριθμών

Περιγραφή: Δημιουργήστε ένα πρόγραμμα που θα διαβάσει μια σειρά αριθμών από τον χρήστη και θα τους ταξινομήσει.

Λύση:

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Δώστε τον αριθμό των στοιχείων:");
        int n = int.Parse(Console.ReadLine());
        int[] array = new int[n];

        Console.WriteLine("Δώστε τα στοιχεία:");
        for (int i = 0; i < n; i++)
        {
            array[i] = int.Parse(Console.ReadLine());
        }

        SelectionSort(array);
        Console.WriteLine("Ταξινομημένος πίνακας: " + string.Join(" ", array));
    }

    static void SelectionSort(int[] array)
    {
        for (int i = 0; i < array.Length - 1; i++)
        {
            int minIndex = i;
            for (int j = i + 1; j < array.Length; j++)
            {
                if (array[j] < array[minIndex])
                {
                    minIndex = j;
                }
            }
            if (minIndex != i)
            {
                Swap(ref array[i], ref array[minIndex]);
            }
        }
    }
}
```

```

    }
}

static void Swap(ref int a, ref int b)
{
    int temp = a;
    a = b;
    b = temp;
}
}

```

Άσκηση 7: Ταξινόμηση Μικτού Πίνακα

Περιγραφή: Δημιουργήστε έναν πίνακα που περιέχει ακέραιους και δεκαδικούς αριθμούς, και ταξινομήστε μόνο τους ακέραιους.

Λύση:

```

using System;

class Program
{
    static void Main()
    {
        object[] array = { 64, 25.5f, 12, 22.3f, 11 };
        SelectionSortIntegers(array);
        Console.WriteLine("Ταξινομημένος πίνακας (ακέραιοι): " + string.Join(", ", array));
    }

    static void SelectionSortIntegers(object[] array)
    {
        for (int i = 0; i < array.Length - 1; i++)
        {
            int minIndex = i;
            for (int j = i + 1; j < array.Length; j++)
            {
                if (array[j] is int && (array[minIndex] is int) && (int)array[j] < (int)array[minIndex])
                {
                    minIndex = j;
                }
            }
            if (minIndex != i)
            {
                Swap(ref array[i], ref array[minIndex]);
            }
        }
    }
}

```

```
    }  
}
```

```
static void Swap(ref object a, ref object b)  
{  
    object temp = a;  
    a = b;  
    b = temp;  
}  
}
```