

Άσκηση 1: Βασική Δυαδική Αναζήτηση

Περιγραφή: Γράψε μια μέθοδο που υλοποιεί τη δυαδική αναζήτηση σε έναν ταξινομημένο πίνακα ακέραιων αριθμών. Δοκίμασε να βρεις ένα στοιχείο.

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        int key = 5;
        int result = BinarySearch(array, 0, array.Length - 1, key);

        if (result != -1)
        {
            Console.WriteLine($"Element {key} found at index: {result}");
        }
        else
        {
            Console.WriteLine($"Element {key} not found in the array.");
        }
    }

    static int BinarySearch(int[] array, int left, int right, int key)
    {
        if (left > right)
            return -1;

        int middle = (left + right) / 2;

        if (array[middle] == key)
            return middle;
        else if (array[middle] > key)
            return BinarySearch(array, left, middle - 1, key);
        else
            return BinarySearch(array, middle + 1, right, key);
    }
}
```

Άσκηση 2: Αναζήτηση με Εναλλαγή Κλειδιού

Περιγραφή: Δημιούργησε μια μέθοδο που αναζητά ένα στοιχείο στον πίνακα και επιστρέφει το μήνυμα αν το στοιχείο δεν βρέθηκε.

```
using System;

class Program
{
    static void Main()
    {
        int[] array = { 10, 20, 30, 40, 50 };
        int key = 25;
        int result = BinarySearch(array, 0, array.Length - 1, key);

        Console.WriteLine(result != -1 ? $"Element {key} found at index: {result}" : $"Element {key}
not found.");
    }

    static int BinarySearch(int[] array, int left, int right, int key)
    {
        if (left > right)
            return -1;

        int middle = (left + right) / 2;

        if (array[middle] == key)
            return middle;
        else if (array[middle] > key)
            return BinarySearch(array, left, middle - 1, key);
        else
            return BinarySearch(array, middle + 1, right, key);
    }
}
```

Άσκηση 3: Δυναδική Αναζήτηση με Συνθήκη Εισόδου

Περιγραφή: Πρόσθεσε μια μέθοδο που ζητά από τον χρήστη να εισάγει το κλειδί και εκτελεί τη δυναδική αναζήτηση.

```
using System;

class Program
{
    static void Main()
```

```

{
    int[] array = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    Console.Write("Enter a number to search: ");
    int key = int.Parse(Console.ReadLine());
    int result = BinarySearch(array, 0, array.Length - 1, key);

    Console.WriteLine(result != -1 ? $"Element {key} found at index: {result}" : $"Element {key}
not found.");
}

static int BinarySearch(int[] array, int left, int right, int key)
{
    if (left > right)
        return -1;

    int middle = (left + right) / 2;

    if (array[middle] == key)
        return middle;
    else if (array[middle] > key)
        return BinarySearch(array, left, middle - 1, key);
    else
        return BinarySearch(array, middle + 1, right, key);
}
}

```

Άσκηση 4: Δυαδική Αναζήτηση σε Πραγματικούς Αριθμούς

Περιγραφή: Υλοποίησε τη δυαδική αναζήτηση σε πίνακα πραγματικών αριθμών και εκτύπωσε την αντίστοιχη θέση του στοιχείου.

```

using System;

class Program
{
    static void Main()
    {
        double[] array = { 1.1, 2.2, 3.3, 4.4, 5.5 };
        double key = 3.3;
        int result = BinarySearch(array, 0, array.Length - 1, key);

        Console.WriteLine(result != -1 ? $"Element {key} found at index: {result}" : $"Element {key}
not found.");
    }

    static int BinarySearch(double[] array, int left, int right, double key)

```

```

{
    if (left > right)
        return -1;

    int middle = (left + right) / 2;

    if (array[middle] == key)
        return middle;
    else if (array[middle] > key)
        return BinarySearch(array, left, middle - 1, key);
    else
        return BinarySearch(array, middle + 1, right, key);
}
}

```

Άσκηση 5: Ευρύτερος Πίνακας και Δοκιμή

Περιγραφή: Δημιούργησε έναν μεγαλύτερο πίνακα και δοκίμασε τη δυαδική αναζήτηση με διαφορετικά κλειδιά.

```
using System;
```

```
class Program
```

```

{
    static void Main()
    {
        // Δημιουργία μεγαλύτερου ταξινομημένου πίνακα
        int[] array = { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41 };

        // Διαφορετικά κλειδιά για αναζήτηση
        int[] keys = { 11, 22, 1, 39, 40 };

        // Έλεγχος για κάθε κλειδί
        foreach (int key in keys)
        {
            int result = BinarySearch(array, 0, array.Length - 1, key);
            Console.WriteLine(result != -1 ? $"Element {key} found at index: {result}" : $"Element {key} not found.");
        }
    }
}

// Μέθοδος δυαδικής αναζήτησης
static int BinarySearch(int[] array, int left, int right, int key)
{
    if (left > right)
        return -1; // Το στοιχείο δεν βρέθηκε

```

```

int middle = (left + right) / 2;

if (array[middle] == key)
    return middle; // Βρέθηκε το στοιχείο
else if (array[middle] > key)
    return BinarySearch(array, left, middle - 1, key); // Αναζήτηση στο αριστερό υποτιμήμα
else
    return BinarySearch(array, middle + 1, right, key); // Αναζήτηση στο δεξί υποτιμήμα
}
}

```

Εκτέλεση

Όταν εκτελέσεις το παραπάνω πρόγραμμα, θα ελέγξει τη δυαδική αναζήτηση για τα κλειδιά 11, 22, 1, 39, και 40. Η έξοδος θα είναι:

```

Element 11 found at index: 5
Element 22 not found.
Element 1 found at index: 0
Element 39 found at index: 19
Element 40 not found.

```

Επεξήγηση

- **Πίνακας:** Δημιουργήσαμε έναν ταξινομημένο πίνακα από ακέραιους αριθμούς.
- **Κλειδιά:** Δοκιμάσαμε πέντε διαφορετικά κλειδιά για αναζήτηση.
- **Δυαδική Αναζήτηση:** Η μέθοδος `BinarySearch` εκτελεί την αναζήτηση και επιστρέφει την θέση του στοιχείου αν το βρει ή `-1` αν δεν υπάρχει στον πίνακα.