

Άσκηση 1: Βασική Αναζήτηση

Περιγραφή: Έχεις έναν πίνακα από αριθμούς που είναι τακτοποιημένος σε αύξουσα σειρά. Γράψε μια μέθοδο που χρησιμοποιεί τον Δυαδικό Αλγόριθμο Αναζήτησης για να βρει αν ένας συγκεκριμένος αριθμός υπάρχει στον πίνακα και επιστρέφει τη θέση του.

Εντολή: Δημιούργησε μια κλάση `BinarySearchExample` με τη μέθοδο `Main`, η οποία θα περιλαμβάνει:

- Έναν πίνακα από αριθμούς: {1, 2, 4, 6, 8, 10, 12, 14, 16, 18}
- Τον αριθμό προς αναζήτηση.
- Μια μέθοδο `BinarySearch` που θα επιστρέφει τη θέση του αριθμού αν βρεθεί ή `-1` αν δεν βρεθεί.

Λύση:

```
using System;
```

```
class BinarySearchExample
{
    static void Main()
    {
        int[] numbers = { 1, 2, 4, 6, 8, 10, 12, 14, 16, 18 };
        int target = 10;

        int result = BinarySearch(numbers, target);

        if (result != -1)
        {
            Console.WriteLine($"Ο αριθμός {target} βρέθηκε στην  
θέση {result}.");
        }
        else
        {
            Console.WriteLine($"Ο αριθμός {target} δεν  
βρέθηκε.");
        }
    }

    static int BinarySearch(int[] array, int target)
    {
        int left = 0;
        int right = array.Length - 1;
```

```

while (left <= right)
{
    int mid = (left + right) / 2;

    if (array[mid] == target)
    {
        return mid;
    }
    else if (array[mid] < target)
    {
        left = mid + 1;
    }
    else
    {
        right = mid - 1;
    }
}

return -1;
}
}

```

Άσκηση 2: Αναζήτηση για Λανθασμένο Αριθμό

Περιγραφή: Χρησιμοποίησε τον ίδιο πίνακα αριθμών όπως στην Άσκηση 1. Τώρα ψάξε για έναν αριθμό που δεν υπάρχει στον πίνακα, για παράδειγμα, τον αριθμό 5.

Εντολή: Τροποποίησε το πρόγραμμα της Άσκησης 1 ώστε να ψάχνει για τον αριθμό 5 και να εμφανίζει κατάλληλο μήνυμα αν δεν βρεθεί.

Λύση:

```

using System;

class BinarySearchExample
{

```

```

static void Main()
{
    int[] numbers = { 1, 2, 4, 6, 8, 10, 12, 14, 16, 18 };
    int target = 5;

    int result = BinarySearch(numbers, target);

    if (result != -1)
    {
        Console.WriteLine($"Ο αριθμός {target} βρίσκεται στην θέση {result}.");
    }
    else
    {
        Console.WriteLine($"Ο αριθμός {target} δεν βρίσκεται.");
    }
}

static int BinarySearch(int[] array, int target)
{
    int left = 0;
    int right = array.Length - 1;

    while (left <= right)
    {
        int mid = (left + right) / 2;

        if (array[mid] == target)
        {
            return mid;
        }
        else if (array[mid] < target)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }
}

```

```

    }

    return -1;
}
}

```

Άσκηση 3: Αναζήτηση με Μεγαλύτερη Είσοδο

Περιγραφή: Δημιούργησε έναν πίνακα με 20 αριθμούς σε αύξουσα σειρά. Γράψε έναν αλγόριθμο αναζήτησης που θα ψάχνει για διάφορους αριθμούς στον πίνακα, όπως 3, 15, και 25.

Εντολή: Αναβάθμισε το πρόγραμμα για να περιλαμβάνει έναν πίνακα με 20 αριθμούς και έλεγξε την αναζήτηση για διαφορετικούς αριθμούς.

Λύση:

```

using System;

class BinarySearchExample
{
    static void Main()
    {
        int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20 };

        int[] targets = { 3, 15, 25 };

        foreach (int target in targets)
        {
            int result = BinarySearch(numbers, target);
            if (result != -1)
            {
                Console.WriteLine($"Ο αριθμός {target} βρέθηκε
στη θέση {result}.");
            }
            else

```

```

        {
            Console.WriteLine($"Ο αριθμός {target} δεν
βρέθηκε.");
        }
    }
}

static int BinarySearch(int[] array, int target)
{
    int left = 0;
    int right = array.Length - 1;

    while (left <= right)
    {
        int mid = (left + right) / 2;

        if (array[mid] == target)
        {
            return mid;
        }
        else if (array[mid] < target)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }

    return -1;
}
}

```

Άσκηση 4: Εύρεση Στοιχείου με Δεδομένο Κριτήριο

Περιγραφή: Δημιούργησε έναν πίνακα με 10 αριθμούς και χρησιμοποίησε τον Δυαδικό Αλγόριθμο Αναζήτησης για να βρεις αν υπάρχει ένα στοιχείο που είναι μεγαλύτερο από 7.

Εντολή: Αναβάθμισε το πρόγραμμα για να ψάχνεις για το πρώτο στοιχείο στον πίνακα που είναι μεγαλύτερο από 7.

Λύση:

```
using System;
```

```
class BinarySearchExample  
{
```

```
    static void Main()
```

```
    {
```

```
        int[] numbers = { 1, 3, 5, 7, 8, 9, 10, 12, 14, 16 };
```

```
        int target = 7;
```

```
        int result = BinarySearchGreaterThan(numbers, target);
```

```
        if (result != -1)
```

```
        {
```

```
            Console.WriteLine($"Το πρώτο στοιχείο  
μεγαλύτερο από {target} είναι {numbers[result]} στη θέση  
{result}.");
```

```
        }
```

```
        else
```

```
        {
```

```
            Console.WriteLine($"Δεν βρέθηκε στοιχείο  
μεγαλύτερο από {target}.");
```

```
        }
```

```
    }
```

```
static int BinarySearchGreaterThan(int[] array, int target)
```

```
{
```

```
    int left = 0;
```

```
    int right = array.Length - 1;
```

```
    while (left <= right)
```

```
    {
```

```
        int mid = (left + right) / 2;
```

```
        if (array[mid] > target)
```

```
        {
```

```
            // Ελέγχουμε αν είναι το πρώτο  
μεγαλύτερο στοιχείο
```

```
        if (mid == 0 || array[mid - 1] <= target)
        {
            return mid;
        }
        right = mid - 1;
    }
    else
    {
        left = mid + 1;
    }
}

return -1;
}
```