

Άσκηση 1: Βασική Αναζήτηση Παρεμβολής

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί Αναζήτηση Παρεμβολής για να βρει ένα αριθμητικό στοιχείο σε μια ταξινομημένη σειρά. Εάν το στοιχείο δεν βρεθεί, η μέθοδος πρέπει να επιστρέφει -1.

Κώδικας:

```
using System;
```

```
class InterpolationSearchBasic  
{
```

```
    static void Main()
```

```
    {
```

```
        int[] array = { 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 };
```

```
        int target = 15;
```

```
        int result = InterpolationSearch(array, target);
```

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

```
        {
```

```
            Console.WriteLine($"Ο αριθμός {target} βρέθηκε στην  
θέση {result}.");
```

```
        }
```

```
        else
```

```
        {
```

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

```
        }
```

```
    }
```

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

```
    {
```

```
        int low = 0;
```

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

```
        while (low <= high && target >= array[low] && target <= array[high])
```

```
        {
```

```
            int pos = low + ((target - array[low]) * (high - low) / (array[high]
```

```

- array[low]));

        if (array[pos] == target)
        {
            return pos;
        }

        if (array[pos] < target)
        {
            low = pos + 1;
        }
        else
        {
            high = pos - 1;
        }
    }

    return -1;
}
}

```

*Άσκηση 2: Αναζήτηση Παρεμβολής με Δοκιμή
Αποτυχίας*

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί Αναζήτηση Παρεμβολής για να βρει ένα αριθμητικό στοιχείο σε μια ταξινομημένη σειρά. Δοκίμασε την μέθοδο με διάφορες περιπτώσεις όπως στοιχείο που δεν υπάρχει στην σειρά και στοιχείο που είναι στην αρχή ή στο τέλος.

Κώδικας:

```
using System;
```

```

class InterpolationSearchFailureTest
{
    static void Main()
    {
        int[] array = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 };

        // Test with target in the array
        TestInterpolationSearch(array, 40);
    }
}

```

```

// Test with target not in the array
TestInterpolationSearch(array, 25);

// Test with target at the beginning of the array
TestInterpolationSearch(array, 10);

// Test with target at the end of the array
TestInterpolationSearch(array, 100);
}

static void TestInterpolationSearch(int[] array, int target)
{
    int result = InterpolationSearch(array, target);
    if (result != -1)
    {
        Console.WriteLine($"Ο αριθμός {target} βρέθηκε στη θέση {result}.");
    }
    else
    {
        Console.WriteLine($"Ο αριθμός {target} δεν βρέθηκε.");
    }
}

static int InterpolationSearch(int[] array, int target)
{
    int low = 0;
    int high = array.Length - 1;

    while (low <= high && target >= array[low] && target <= array[high])
    {
        int pos = low + ((target - array[low]) * (high - low) / (array[high] - array[low]));

        if (array[pos] == target)
        {
            return pos;
        }

        if (array[pos] < target)
        {
            low = pos + 1;
        }
        else
        {
            high = pos - 1;
        }
    }
}

```

```

    }

    return -1;
}
}

```

Άσκηση 3: Συνοδευτικό Παράδειγμα με Μεγαλύτερο Πίνακα

Περιγραφή: Δημιούργησε μια μέθοδο που χρησιμοποιεί Αναζήτηση Παρεμβολής σε έναν μεγαλύτερο ταξινομημένο πίνακα αριθμών και δοκίμασέ την με διάφορους αριθμούς.

Κώδικας:

```

using System;

class InterpolationSearchLargeArray
{
    static void Main()
    {
        int[] array = GenerateLargeArray();
        int[] targets = { 15, 150, 1000, 5000 };

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

```

```

static int[] GenerateLargeArray()
{
    int size = 10000;
    int[] array = new int[size];
    for (int i = 0; i < size; i++)
    {
        array[i] = i * 2; // Δημιουργεί έναν πίνακα με
        ξυρούς αριθμούς
    }
    return array;
}

static int InterpolationSearch(int[] array, int target)
{
    int low = 0;
    int high = array.Length - 1;

    while (low <= high && target >= array[low] && target <= array[high])
    {
        int pos = low + ((target - array[low]) * (high - low) / (array[high]
        - array[low]));

        if (array[pos] == target)
        {
            return pos;
        }

        if (array[pos] < target)
        {
            low = pos + 1;
        }
        else
        {
            high = pos - 1;
        }
    }

    return -1;
}

```

