

Άσκηση 1: Υπολογισμός Αριθμού Fibonacci με DP

Να γραφεί μια μέθοδος σε C# που να υπολογίζει τον n-οστό αριθμό Fibonacci χρησιμοποιώντας δυναμικό προγραμματισμό.

```
using System;
```

```
public class FibonacciDP
{
    public static int Fibonacci(int n)
    {
        if (n <= 1)
            return n;

        int[] memo = new int[n + 1];
        memo[0] = 0;
        memo[1] = 1;

        for (int i = 2; i <= n; i++)
        {
            memo[i] = memo[i - 1] + memo[i - 2];
        }

        return memo[n];
    }

    public static void Main()
    {
        int n = 6;
        int result = Fibonacci(n);
        Console.WriteLine($"Ο {n}-οστός αριθμός Fibonacci είναι: {result}");
    }
}
```

Άσκηση 2: Υπολογισμός Μέγιστης Ακολουθίας Αύξουσας Υπολογισμένων Αριθμών (LIS)

Να γραφεί μια μέθοδος σε C# που να υπολογίζει το μήκος της μέγιστης αυξανόμενης υποσειράς αριθμών σε έναν πίνακα αριθμών.

```
using System;
```

```
public class LISDP
```

```

{
    public static int LengthOfLIS(int[] nums)
    {
        int n = nums.Length;
        if (n == 0)
            return 0;

        int[] dp = new int[n];
        dp[0] = 1;
        int maxLength = 1;

        for (int i = 1; i < n; i++)
        {
            dp[i] = 1;
            for (int j = 0; j < i; j++)
            {
                if (nums[i] > nums[j])
                {
                    dp[i] = Math.Max(dp[i], dp[j] + 1);
                }
            }
            maxLength = Math.Max(maxLength, dp[i]);
        }

        return maxLength;
    }

    public static void Main()
    {
        int[] nums = { 10, 22, 9, 33, 21, 50, 41, 60 };
        int lisLength = LengthOfLIS(nums);
        Console.WriteLine($"Το μήκος της μέγιστης αυξανόμενης υποσειράς είναι: {lisLength}");
    }
}

```

Άσκηση 3: Υπολογισμός Ελάχιστης Ακολουθίας Αθροίσματος Υποσειράς (Minimum Subset Sum Difference)

Να γραφεί μια μέθοδος σε C# που να υπολογίζει τη διαφορά του ελάχιστου αθροίσματος υποσειράς αριθμών από το συνολικό άθροισμα του πίνακα.

```
using System;
```

```

public class MinSubsetSumDifferenceDP
{
    public static int MinSubsetSumDifference(int[] nums)

```

```

{
    int n = nums.Length;
    int totalSum = 0;
    foreach (int num in nums)
    {
        totalSum += num;
    }

    bool[] dp = new bool[totalSum + 1];
    dp[0] = true;

    for (int i = 0; i < n; i++)
    {
        for (int j = totalSum; j >= nums[i]; j--)
        {
            if (dp[j - nums[i]])
            {
                dp[j] = true;
            }
        }
    }

    int minDifference = int.MaxValue;
    for (int j = totalSum / 2; j >= 0; j--)
    {
        if (dp[j])
        {
            minDifference = totalSum - 2 * j;
            break;
        }
    }

    return minDifference;
}

public static void Main()
{
    int[] nums = { 3, 1, 4, 2, 2, 1 };
    int minDiff = MinSubsetSumDifference(nums);
    Console.WriteLine($"Η ελάχιστη διαφορά του αθροίσματος της υποσειράς είναι: {minDiff}");
}
}

```

