

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace veri_yapilari_kule { internal class Program {

public class HanoiKuleleri { // Hamle sayısını takip etmek için global veya sınıf
seviyesinde bir sayaç private static int hamleSayisi = 0;

    /// <summary>
    /// Hanoi Kulesi problemini çözen özyinelemeli (recursive)
fonksiyon.
    /// </summary>
    /// <param name="n">Taşınacak disk sayısı (en büyük
disk).</param>
    /// <param name="kaynak">Disklerin başlangıçtaki çubuğu
(örneğin 'A').</param>
    /// <param name="hedef">Disklerin taşınacağı son çubuk
(örneğin 'C').</param>
    /// <param name="yardimci">Geçici olarak kullanılacak üçüncü
çubuk (örneğin 'B').</param>
    public static void HanoiCoz(int n, char kaynak, char hedef,
char yardimci)
    {
        if (n == 1)
        {
            // Temel Durum: Sadece bir disk varsa, doğrudan
hedefe taşı.
            Console.WriteLine($"Adım {++hamleSayisi}: Diski
{kaynak}'dan {hedef}'a taşı.");
        }
        else
        {
            // 1. En üstteki n-1 disk kaynaktan yardımcı çubuğa
taşı.
            // Bu aşamada hedef ve yardımcı çubukların rolleri
değişir.
            HanoiCoz(n - 1, kaynak, yardimci, hedef);
```

```
// 2. En büyük (n. disk) diski kaynaktan hedef
çubuğa taşı.
    Console.WriteLine($"Adım {++hamleSayisi}: Diski
{kaynak}'dan {hedef}'a taşı.");

    // 3. n-1 diski yardımcı çubuktan hedef çubuğa taşı.
    // Bu aşamada kaynak ve yardımcı çubukların rolleri
değişir.
    HanoiCoz(n - 1, yordimci, hedef, kaynak);
}
}

// Ana programın çalıştığı Main metodu
public static void Main(string[] args)
{
    int diskSayisi = 3; // Örneğin 3 disk
    char baslangicCubugu = 'A';
    char hedefCubugu = 'C';
    char yordimciCubuk = 'B';

    // Hamle sayacını sıfırla (eğer Main birden fazla kez
çalıştırılıyorsa faydalı)
    hamleSayisi = 0;

    Console.WriteLine($"Hanoi Kuleleri Çözümü ({diskSayisi}
disk ile):");
    Console.WriteLine(new string('-', 40));

    // Çözüm fonksiyonunu çağır
    HanoiCoz(diskSayisi, baslangicCubugu, hedefCubugu,
yordimciCubuk);

    // Sonuçları yazdır
    Console.WriteLine(new string('-', 40));
    int minimumHamle = (int)Math.Pow(2, diskSayisi) - 1;
    Console.WriteLine($"Toplam minimum hamle sayısı:
{hamleSayisi}");
    Console.WriteLine($"Doğrulama (Formül  $2^n - 1$ ):
{minimumHamle}");

    // Konsolun kapanmaması için bekletme (Opsiyonel, sadece
bazı konsol ortamları için)
```

```
        // Console.ReadKey();  
    }  
}  
  
}
```