```
public static int BinarisKeres(int[] tomb, int keresettertek)
{
    int eleje = 0;
    int vege = tomb.Length - 1;
    while (eleje <= vege)
    {
        int i = (eleje + vege) / 2;
        if (tomb[i] == keresettertek) return i;
        else if (tomb[i] < keresettertek)
        {
            eleje = i + 1;
        }
        else if (tomb[i] > keresettertek)
        {
            vege = i - 1;
        }
    }
    return -1;
}
```

```
//rekurzív implementáció
3 references
public static int BinarisKeresRekurziv(int[] tomb, int keresettertek, int eleje, int vege)
{
    int mid = (eleje + vege) / 2;
    if (vege < 1)
    {
        return -1;
    }
    if (tomb[mid] == keresettertek)
    {
        return mid;
    }
    if (tomb[mid] > keresettertek)
        return BinarisKeresRekurziv(tomb, keresettertek, eleje, mid - 1);

    return BinarisKeresRekurziv(tomb, keresettertek, mid + 1, vege);
}
```

```
static void Main(string[] args)
{
    var tomb = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };

    int index = BinarisKeres(tomb, 8);
    int index2 = BinarisKeresRekurziv(tomb, 1, 0, tomb.Length);
    int index3 = BinarisKeres(tomb, 18); //-1, mert nincs ilyen

    //int index = Array.BinarySearch(tomb, 8);
    Console.WriteLine("A nyolcas indexe: {0}", index);
    Console.WriteLine("Az egyes indexe: {0}", index2);
    Console.WriteLine("Az 18 indexe: {0}", index3);

    Console.ReadKey();
}
```