// Author of question: Snorri Agnarsson

// Permalink of question: https://rise4fun.com/Dafny/CGB1z

// Authors of solution:   Alexander Guðmundsson

// Permalink of solution: https://rise4fun.com/Dafny/VnB5

// Use the command

//   dafny H2-skeleton.dfy

// or

//   compile H2-skeleton.dfy

// to compile the file.

// Or use the web page rise4fun.com/dafny.

// When you have solved the problem put

// the solution on the Dafny web page,

// generate a permalink and put it in

// this file.

method SearchRecursive( a: seq<real>, i: int, j: int, x: real ) returns ( k: int )

    decreases j-i;

    requires 0 <= i <= j <= |a|;

    requires forall p, q :: i <= p < q < j ==> a[p] >= a[q];

    ensures i <= k <= j

    ensures forall r | i <= r < k :: a[r] >= x;

    ensures forall r | k <= r < j :: a[r] < x;

{

    if i == j

    {

        return i;

    }

    var m := i + (j-i)/2;

    if a[m] < x

    {

        k := SearchRecursive(a,i,m,x);

    }

    else

    {

        k := SearchRecursive(a,m+1,j,x);

    }

}

method SearchLoop( a: seq<real>, i: int, j: int, x: real ) returns ( k: int )

    requires 0 <= i <= j <= |a|;

    requires forall p, q :: i <= p < q < j ==> a[p] >= a[q];

    ensures i <= k <= j;

    ensures forall r | i <= r < k :: a[r] >= x;

    ensures forall r | k <= r < j :: a[r] < x;

{

    if i == j

    {

        return i;

    }

    var p := i;

    var q := j;

    while p != q

        decreases q-p;

        invariant i <= p <= q <= j;

        invariant forall r | i <= r < p :: a[r] >= x;

        invariant forall r | q <= r < j :: a[r] < x;

    {

        var m := p + (q-p)/2;

        if a[m] < x

        {

            q := m;

        }

        else

        {

            p := m+1;

        }

    }

    return p;

}

// Ef eftirfarandi fall er ekki samþykkt þá eru

// föllin ekki að haga sér rétt að mati Dafny.

method Test( a: seq<real>, x: real )

    requires forall p,q | 0 <= p < q < |a| :: a[p] >= a[q];

{

    var k1 := SearchLoop(a,0,|a|,x);

    assert forall r | 0 <= r < k1 :: a[r] >= x;

    assert forall r | k1 <= r < |a| :: a[r] < x;

    var k2 := SearchRecursive(a,0,|a|,x);

    assert forall r | 0 <= r < k2 :: a[r] >= x;

    assert forall r | k2 <= r < |a| :: a[r] < x;

}