```
// Author of question: Snorri Agnarsson
// Permalink of question: https://rise4fun.com/Dafny/0HRr

// Author of solution:    Alexander Guðmundsson
// Permalink of solution: https://rise4fun.com/Dafny/8pxWd

// Use the command
//   dafny LinearSearch-skeleton.dfy
// or
//   compile LinearSearch-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<int>, i: int, j: int, x: int ) returns
(k: int)
    decreases j-i;
    requires 0 <= i <= j <= |a|;
    ensures i <= k < j || k == -1;
    ensures k != -1 ==> a[k] == x;
    ensures k != -1 ==> forall r | k < r < j :: a[r] != x;
    ensures k == -1 ==> forall r | i <= r < j :: a[r] != x;
{

    // Put program text here so that Dafny
    // accepts this function.
    // In this function loops are not allowed
    // but recursion should be used, and it
    // is not allowed to call the function
    // SearchLoop below.

    if j == i
    {
        k := -1;
        return;
    }
    if a[j-1] == x
    {
        k := j-1;
```

```
        return;

    }
    else
    {
        k := SearchRecursive(a, i, j-1, x);
    }
}




method SearchLoop( a: seq<int>, i: int, j: int, x: int ) returns (k: i
nt)
    requires 0 <= i <= j <= |a|;
    ensures i <= k < j || k == -1;
    ensures k != -1 ==> a[k] == x;
    ensures k != -1 ==> forall r | k < r < j :: a[r] != x;
    ensures k == -1 ==> forall r | i <= r < j :: a[r] != x;
{
    // Put program text here so that Dafny
    // accepts this function.
    // In this function recursion is not allowed
    // and it is not allowed to call the function
    // SearchRecursive above.

    if i == j
    {
        return -1;
    }

    var t := j;
    while t > i && a[t-1] != x
        decreases t;



    {
        if a[t-1] == x
        {
            k := t-1;
            return;
        }
        else
```

```
        {
            t := t - 1;

        }


    }

    k := -1;




}
```