Here are couple of useful functions:

```java
static boolean isFound(int val, int[] data)
// Lets caller know if number val appears
// as an element of array data.
{
    for(int x : data) {
        if( x==val ) return true;
    }
    return false;
}
```

```java
static int whereFound(int val, int[] data)
// Lets caller know at what position
// number val appears
// as an element of array data -
// or else returns -1
{
    for(int k=0; k<data.length; ++k) {
        int x = data[k];
        if( x==val ) return k;
    }
    return -1;
}
```
The previous isFound follows a "boolean Search Function" model. This is PSEUDO code - not really Java programming.

```java
boolean isFound(   searchInfo   ,    searchRange   )
// tells caller if searchRange contains an element
// satisfying the searchInfo
{
    Loop running variable x through searchRange {
        if(x satisfies searchInfo) return true;
    }
    return false;
}
```

**searchInfo**: some parameters determining what is being searched for

**searchRange**: some parameters determining a range of elements amongst which to search.

**Loop**: one of the various kinds of possible loop

**x**: a loop variable (or variables) representing ONE element of the searchRange.

**satisfies**: some kind of condition that explains how the element x satisfies the searchInfo.
The previous whereFound follows a "position Search Function" model. This is PSEUDO code - not really Java programming.

```
boolean whereFound(   searchInfo   ,    searchRange   )
// tells caller where the searchRange contains
// an element satisfying the searchInfo
{
    Loop running variable k through positions in the searchRange
    {
        x = element of searchRange at position k.
        if(x satisfies searchInfo) return k;
    }
    return bogus position;
}
```

**searchInfo**: some parameters determining what is being searched for

**searchRange**: some parameters determining a range of elements amongst which to search.

**Loop**: one of the various kinds of possible loop

**k**: a loop variable (or variables) representing the position of ONE element of the searchRange.

**x**: a loop variable (or variables) representing ONE element of the searchRange.

**satisfies**: some kind of condition that explains how the element x satisfies the searchInfo.