

Hands-On: Linear Scan

As we've seen, general *solution strategies* are important to understand because they can be used to solve many different problems. *Linear scan* is the term I used for a general solution strategy that we've applied to two different problems so far: finding the location of an element in an array and finding the minimum element in an array.

Linear scan is very simple, and we can express it at a high level as follows.

```
for each item that we have {
    process the current item
}
```

Of course, that isn't the solution to any problem at all; that's why we call this a *strategy*. It's a general approach that can be tailored to form the solution to a specific problem. For example, we can apply the linear scan strategy to find the minimum element in an array as follows.

```
set min to the first element of the array
for each remaining element in the array {
    if the current element is less than min {
        set min to the current element
    }
}
min now stores the smallest element in the array
```

As another example, we can apply the linear scan strategy to find the location of an element in an array.

```
for each element in the array {
    if the current element matches the value we're searching for {
        return the index of the current element
    }
}
return a value indicating that the value we're searching for isn't in the array
```

These solutions are just pseudocode, but they're still solutions. And they are important first steps in constructing any solution in software. We just have to express those solutions in the programming language that we're using (Java, in our case).

Here is one translation of that last solution into Java.

```
public static int search(int[] a, int target) {
    for (int i = 0; i < a.length; i++) {
```

```
        if (a[i] == target) {  
            return i;  
        }  
    }  
    return -1;  
}
```

Applying linear scan

Let's think about using linear scan to find how many negative values are in an array of integers. Start by thinking about the general linear scan strategy:

```
for each item that we have {  
    process the current item  
}
```

Tailoring that to our problem becomes:

```
initialize a counter called negativeCount to 0  
for each element in the array {  
    if the current element is less than zero {  
        add one to negativeCount  
    }  
}  
return negativeCount
```

Submission

The submission page for this activity asks you to apply linear scan to express the solution to a problem and then submit it for a grade.