Sequential search

SEARCHING FOR SPECIFIC VALUES

In the sequential search, each element of the array is compared to the **key value**.

The search is sequential or in the order it appears in the array. If you are looking for an element that is near the front of the array, the sequential search will find it quickly. The more data that must be searched, the longer it will take to find the data that matches the key.

Consider this method which will search for a **key** integer value. If found, the index (subscript) of the first location of the **key** will be returned. If not found, a value of -1 will be returned.

```
public static int search(int [ ] numbers, int key)
{
     for (int index = 0; index < numbers.length; index++)
    {
        if ( numbers[index] = = key )
            return index;  //FOUND
    }
   // If not found return -1
   return -1;
}
```

If the **key** value is found, the index (subscript) of the location is returned.

This tells us that the return value x, will be the first integer found such that numbers [x] = key.

There may be additional **key** locations in this array beyond this location.

Search a String array

```
//Find the number of times the name "Leah" appears in an array of name
public static void main(String[] args)
{
   String key = "Leah";
   String[ ] list = {"Betty","Mary","Liam","Andrew","Caroline","Maria","Crystal","Lee","Leah","Li",Leah"};   // create array

   int count = search (list, key);    // invoke the method
   System.out.println("Count = " + count);
}
public static int search(String [ ] list, String key) //method to find "Leah"
{
    int i, count = 0;
    for( i = 0; i< list.length; i++)
    {
        if (list [ i ].equals( key ))
          count = count+1;
    }
    return (count);
}
```