Searching Worksheet

1. Given an array of size **n**, how many comparisons is needed using a linear search to determine if a specific value is in the list using a **best case** scenario?

2. Given an array of size **n**, how many comparisons is needed using a linear search to determine if a specific value is in the list using a **worst case** scenario?

3. Given an array of size **n**, how many comparisons is needed using a linear search to determine if a specific value is in the list using an **average case** scenario?

```
public int binarySearch(int[] a, int key)
{
   int left = 0;                           // Establish the initial
   int right = a.length - 1;               // endpoints of the array

   while(left <= right)                    // Loop until endpoints cross
   {
       int midpoint = (left + right) / 2;  // Compute the current midpoint
       if(a[midpoint] == key)              // Target found; return its index
          return midpoint;
       else if(a[midpoint] < key)          // Target to right of midpoint
          left = midpoint + 1;
       else                                // Target to left of midpoint
          right = midpoint - 1;
   }
   return -1;                              // Target not found
}
```

4. Given the following array: `int[] a = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}`
   Using a binary search how many loop iterations are required to find the value 30 in the list?

5. Given the following array: `int[] a = {17, 21, 33, 38, 55, 60, 72, 88, 94}`
   Using a binary search how many loop iterations are required to find the value 72 in the list?

6. Given an array of size **n**, how many comparisons is needed using a binary search to determine if a specific value is in the list using a **best case** scenario?

7. Given an array of size **n**, how many comparisons is needed using a binary search to determine if a specific value is in the list using a **worst case** scenario?