1. You are given an array with 1000 elements. What are the minimum and maximum number of comparisons required for a successful search using:
   (a) linear search
      i. min:
      ii. max:
   (b) binary search
      i. min:
      ii. max:

(Why?)

2. Perform and trace binary search for the number 76 in this array. Show the comparison element in each pass. Perform any pre-processing if required.

<table>
<thead>
<tr>
<th>94</th>
<th>82</th>
<th>76</th>
<th>15</th>
<th>35</th>
<th>28</th>
<th>41</th>
<th>47</th>
<th>51</th>
<th>64</th>
<th>10</th>
<th>77</th>
<th>8</th>
<th>85</th>
<th>1</th>
</tr>
</thead>
</table>
3. Help me complete this method for Binary Search!!

```java
public static int recursiveBinarySearch(int[] list, int key) {
    int low = 0;
    int high = list.length - 1;
    return __________________________ ;   //line #4
}
```

```java
public static int recursiveBinarySearch(int[] list, int key,
    int low, int high) {
    if (low > high)  // The list has been exhausted without a match
        return -low -1;   // Return -insertion point -1
    int mid = (low + high) / 2;
    if (key < list[mid])
        return recursiveBinarySearch(list, key, low, mid - 1);
    else if (key == list[mid])
        return mid;
    else
        return recursiveBinarySearch(list, key, mid + 1, high);
}
```

A. recursiveBinarySearch(list, key)
B. recursiveBinarySearch(list, key, low + 1, high - 1)
C. recursiveBinarySearch(list, key, low - 1, high + 1)
D. recursiveBinarySearch(list, key, low, high)
E. None of the above. Here is the correct code for line #4: ____________________________ ;
5. Help me complete this method for computing a fibonacci number (0,1,2,3,5,8,13,21,...)!!

```java
public static long fib(long index) {
    if (index == 0)    // Base case
        return 0;
    else if (index == 1)  // Base case
        return 1;
    else                // Reduction and recursive calls
        return _____________________ ;
}
A. fib(index - 1)
B. fib(index - 2)
C. fib(index - 2) + fib(index - 1)
D. fib(index + 1) + fib(index - 2)
E. None of the above.
```
4. What is selection sort? Why does the worst case occur when the array elements are already sorted in descending order?

Complete this code for selection sort and trace the output (each pass).

// SelectionSort.java - sort an array of integers
class SelectionSort {
    public static void main(String[] args) {
        int[] a = {7, 3, 66, 3, -4, 22, -70, 2};
        sort(a);
        for (int i = 0; i < a.length; i++){
            System.out.println(a[i]);
        }
    }
}

// implements the selection sort algorithm
static void selectionSorter(int[] data) {
    ...
}