2 Mystery

```
1   /** This is a function (a.k.a. method). It takes an array
2     * of integers as an argument, and returns an integer. */
3   public static int mystery(int[] inputArray, int k) {
4       int x = inputArray[k];
5       int answer = k;
6       int index = k + 1;
7       while (index < inputArray.length) {
8           if (inputArray[index] < x) {
9               x = inputArray[index];
10              answer = index;
11          }
12          index = index + 1;
13      }
14      return answer;
15  }
16
17  /** Extra for experts. This is another function. It takes an
18    * array of integers and returns nothing at all. */
19  public static void mystery2(int[] inputArray) {
20      int index = 0;
21      while (index < inputArray.length) {
22          int targetIndex = mystery(inputArray, index);
23          int temp = inputArray[targetIndex];
24          inputArray[targetIndex] = inputArray[index];
25          inputArray[index] = temp;
26          index = index + 1;
27      }
28  }
```

- What does `mystery` return if `inputArray` is the array 3, 0, 1, 6, 3, and `k` is 2?
- Describe, in English, what `mystery` returns.
- Extra for experts: What does `mystery2` do if `inputArray` is the array 3, 0, 1, 6, 3? Describe, in English, what `mystery2` does to the array.

3 Writing Your First Program

```
/** fib(N) returns the N^th Fibonacci number, for N ≥ 0.
  * The Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, 13, 21, ... */
public static int fib(int N) {    // (use other side for more space)
```

Extra for experts: Complete `fib2` in 5 lines or less. Your answer must be efficient.

```
public static int fib2(int n, int k, int f0, int f1)
```