1 Our First Java Program

Below is our first Java program of the semester. Next to each line, write out what you think the code will do when run. This exercise is adapted from Head First Java.

```java
int size = 27;
String name = "Fido";
Dog myDog = new Dog(name, size);
int x = size - 5;
if (x < 15) {
    myDog.bark(8);
}
while (x > 3) {
    x -= 1;
    myDog.play();
}
int[] numList = {2, 4, 6, 8};
System.out.print("Hello ");
System.out.println("Dog: " + name);
System.out.println(numList[1]);
if (numList[3] == 8) {
    System.out.println("potato");
}
```

For your convenience, here is the same code in Python:

```python
size = 27
name = "Fido"
myDog = Dog(name, size)
x = size - 5
if x < 15:
    myDog.bark(8)
while x > 3:
    x -= 1
    myDog.play()
umList = [2, 4, 6, 8]
print("Hello")
print("Dog: " + name)
print(numList[1])
if numList[3] == 8:
    print("potato")
```
2 Mystery

This is a function (a.k.a. method). It takes an array of integers and an integer as arguments, and returns an integer.

```java
public static int mystery(int[] inputArray, int k) {
    int x = inputArray[k];
    int answer = k;
    int index = k + 1;
    while (index < inputArray.length) {
        if (inputArray[index] < x) {
            x = inputArray[index];
            answer = index;
        }
        index = index + 1;
    }
    return answer;
}
```

(a) What `mystery` returns if `inputArray = [3, 0, 4, 6, 3]` and `k = 2`.

(b) Can you explain in English what does `mystery` do?

Extra: This is another function. It takes an array of integers and returns nothing.

```java
public static void mystery2(int[] inputArray) {
    int index = 0;
    while (index < inputArray.length) {
        int targetIndex = mystery(inputArray, index);
        int temp = inputArray[targetIndex];
        inputArray[targetIndex] = inputArray[index];
        inputArray[index] = temp;
        index = index + 1;
    }
}
```

Describe what `mystery2` does if `inputArray = [3, 0, 4, 6, 3]`. 
3 Writing Your First Program

Implement `fib` which takes in an integer `n` and returns the `n`th Fibonacci number. You may not need to use all the lines.

The Fibonacci sequence is 0, 1, 2, 3, 5, 8, 13, 21, . . .

```java
public static int fib(int n) {
    // Implement the Fibonacci sequence calculation
}
```

*Extra:* Implement `fib` in 5 lines or fewer. Your answer must be efficient. You don't have to make use of the parameter `k` in your solution.

```java
public static int fib2(int n, int k, int f0, int f1) {
    // Implement the Fibonacci sequence calculation
}
```