CSCI 145 Week 6
Lab 4

Goals for this Lab

1. Practice with recursion.
2. Returning multiple values from a function in Java.

Task Description

Exercise 1: Towers of Hanoi

I've uploaded some notes on recursion. In these notes there is a discussion of the recursive solution to the Towers of Hanoi puzzle. In this exercise you will write a Java program that implements this solution. Here are the requirements for your program:

1. Your program should be named Hanoi.java.
2. Your program will accept a single command line argument which is the number of disks in the puzzle. This argument must be a positive (> 0) integer that is <= 64. (Note: 64 is an unreasonably large number of disks.) Your program will produce an appropriate error message if any of this fails. (That is, wrong number of arguments, not an integer, too small, too big.)
3. The three needles in the puzzle will be named S, D, and A. You should use variables with the `char` type to designate needles.
4. Your program will output the instructions for moving the specified number of disks from needle S to needle D. The output will consist of one line for each instruction. The line will look like:

   ```
   Move disk n from needle X to needle Y.
   ```

   Where `n` is the number of the disk being moved and `X` and `Y` are the names of the needles (S, D, or A).

Here is a sample of what your output should look like:

```
> java -cp . Hanoi 3
Move disk 1 from S to D
Move disk 2 from S to A
Move disk 1 from D to A
Move disk 3 from S to D
Move disk 1 from A to S
Move disk 2 from A to D
Move disk 1 from S to D
```

Exercise 2: Compute an Efficient Fibonacci

In the notes on recursion, there is discussed an efficient recursive Fibonacci program. The fib.py program, on the Canvas site, is an example of such a program. In this exercise, you will need to write a version of fib.py in Java that computes Fibonacci numbers efficiently. Here are the requirements for

your program:

1. Your program should be named Fibonacci.java.
2. Your program should prompt on the console for a single integer input `n`. This argument should be non-negative (>= 0) and <= 46. Your program should produce an appropriate error message if the number is too big or too small. (You do not have to handle the error if a non-integer is input.)
3. Your program should output the value of fib(n) where fib(n) is the nth Fibonacci number. (fib(0) = 0, fib(1) = 1, fib(2) = 1, etc.).

Here is a sample output:

```
Enter n (0 <= n <= 46): 46
fib(46) is 1836311903
```

Exercise 2 Notes

In order to make this program work, you need to return multiple integers from a function. Java does not support this directly. In order to make this work, your auxiliary function (`fibaux` in fib.py) should return an array of two integers. Thus, in this case, the return type for `fibaux` should be `int[]`.

You know that you can create a new array variable and initialize it in a single statement as follows:

```
type[] arrayVar = {initializers};
```

In addition to this, you can create a new, initialized array using the expression:

```
new type[] {initializers}
```

where *`type`* and *`initializers`* are the same as the declaration and initialization of an array variable. This allows you to create an array without having to create a separate array variable. So, for example, the statement

```
return 1, 0
```

in fib.py could be written in Java as

```
return new int[] {1, 0};
```

You can use this to reduce the amount of code you need to enter for this exercise.

Additional note: Usually, using an array is not the best way to handle returning multiple values in Java. The better approach is to make a class whose fields are the return values and return an instance of that class.

Turn in your program on Canvas

Create a zip file with your work. It should contain the following two files: `Hanoi.java` and `Fibonacci.java`. Look for the Assignment link on the Canvas page for Lab 4 and click that to turn in the zip file you just created.

Grading

Your program is due by 11:59pm, Sunday, February 14th. The grading will be 50% for each of the two exercises.